SYNGRESS®

4 FREE BOOKLETS
YOUR SOLUTIONS MEMBERSHIP

CONFIGURING Juniper® Networks NetScreen® & SSG Firewalls

"...The authors of this book have done a wonderful job collecting and collating what we need to know to make intelligent networking decisions."

—Scott Kriens, CEO, Juniper® Networks

Rob Cameron Technical Editor
Brad Woodberg
Mohan Krishnamurthy Madwachar
Mike Swarm
Neil R. Wyler
Matthew Albers
Ralph Bonnell

FOREWORD BY SCOTT KRIENS
CEO, JUNIPER NETWORKS

SYNGRESS®

CONFIGURING Juniper® Networks NetScreen® & SSG Firewalls

Rob Cameron **Technical Editor**
Brad Woodberg
Mohan Krishnamurthy Madwachar
Mike Swarm
Neil R. Wyler
Matthew Albers
Ralph Bonnell

FOREWORD BY SCOTT KRIENS
CEO, JUNIPER NETWORKS

Syngress Publishing, Inc., the author(s), and any person or firm involved in the writing, editing, or production (collectively "Makers") of this book ("the Work") do not guarantee or warrant the results to be obtained from the Work.

There is no guarantee of any kind, expressed or implied, regarding the Work or its contents. The Work is sold AS IS and WITHOUT WARRANTY. You may have other legal rights, which vary from state to state.

In no event will Makers be liable to you for damages, including any loss of profits, lost savings, or other incidental or consequential damages arising out from the Work or its contents. Because some states do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

You should always use reasonable care, including backup and other appropriate precautions, when working with computers, networks, data, and files.

Syngress Media®, Syngress®, "Career Advancement Through Skill Enhancement®," "Ask the Author UPDATE®," and "Hack Proofing®," are registered trademarks of Syngress Publishing, Inc. "Syngress: The Definition of a Serious Security Library"™, "Mission Critical™," and "The Only Way to Stop a Hacker is to Think Like One™" are trademarks of Syngress Publishing, Inc. Brands and product names mentioned in this book are trademarks or service marks of their respective companies.

KEY	SERIAL NUMBER
001	HJIRTCV764
002	PO9873D5FG
003	829KM8NJH2
004	5489IJJLPP
005	CVPLQ6WQ23
006	VBP965T5T5
007	HJJJ863WD3E
008	2987GVTWMK
009	629MP5SDJT
010	IMWQ295T6T

PUBLISHED BY
Syngress Publishing, Inc.
800 Hingham Street
Rockland, MA 02370

Configuring Networks NetScreen & SSG Firewalls

Copyright © 2007 by Syngress Publishing, Inc. All rights reserved. Except as permitted under the Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher, with the exception that the program listings may be entered, stored, and executed in a computer system, but they may not be reproduced for publication.

1 2 3 4 5 6 7 8 9 0
ISBN-10: 1-59749-118-7
ISBN-13: 978-1-59749-118-1

Publisher: Andrew Williams
Acquisitions Editor: Gary Byrne
Technical Editor: Rob Cameron
Cover Designer: Michael Kavish

Page Layout and Art: Patricia Lupien
Copy Editors: Mike McGee, Sandy Jolley
Indexer: Nara Wood

Distributed by O'Reilly Media, Inc. in the United States and Canada.

For information on rights, translations, and bulk sales, contact Matt Pedersen, Director of Sales and Rights, at Syngress Publishing; email matt@syngress.com or fax to 781-681-3585.

Lead Author and Technical Editor

Rob Cameron (JNCIS-FWV, JNCIA-M, CCSP, CCSE+) is a Security Solutions Engineer for Juniper Networks. He currently works to design security solutions for Juniper Networks that are considered best practice designs. Rob specializes in network security architecture, firewall deployment, risk management, and high-availability designs. His background includes five years of security consulting for more than 300 customers. This is Rob's second book; the previous one being *Configuring NetScreen Firewalls* (ISBN: 1-932266-39-9) published by Syngress Publishing in 2004.

Contributing Authors

Matthew Albers (CCNP, CCDA, JNCIA-M, JNCIS-FWV, JNCIA-IDP) is a senior systems engineer for Juniper Networks. He currently serves his enterprise customers in the Northern Ohio marketplace. His specialties include routing platforms, WAN acceleration, firewall/VPNs, intrusion prevention, strategic network planning, network architecture and design, and network troubleshooting and optimization. Matthew's background includes positions as a senior engineer at First Virtual Communications, Lucent Technologies, and Bay Networks.

Matthew wrote Chapter 1 and cowrote Chapter 11.

Ralph Bonnell (CISSP, LPIC-2, CCSI, CCNA, MCSE: Security) is a senior information security consultant at Accuvant in Denver, CO. His primary responsibilities include the deployment of various network security products and product training. His specialties include NetScreen deployments, Linux client and server deployments, Check Point training, firewall clustering, and PHP Web programming. Ralph also runs a Linux consulting firm called Linux Friendly. Before moving to Colorado, Ralph was a senior security engineer and instructor at Mission Critical Systems, a Gold Check Point partner and training center in South Florida.

Ralph cowrote Chapter 11.

Mohan Krishnamurthy Madwachar (JNCIA-FWV, CWNA, and CCSA) is AVP-Infrastructure Services for ADG Infotek, Inc., Almoayed Group, Bahrain. Almoayed Group is a leading systems integration group that has branches in seven countries and executes projects in nearly 15 countries. Mohan is a key contributor to the company's infrastructure services division and plays a key role in the organization's network security and training initiatives. Mohan has a strong networking, security, and training background. His tenure with companies such as Schlumberger Omnes and Secure Network Solutions India adds to his experience and expertise in implementing large and complex network and security projects.

Mohan holds leading IT industry certifications and is a member of the IEEE and PMI.

Mohan would like to dedicate his contributions to this book to his sister, Geetha Prakash, and her husband, C.V. Prakash, and their son, Pragith Prakash.

Mohan has coauthored the book *Designing and Building Enterprise DMZs* (ISBN: 1-597491004), published by Syngress Publishing. He also writes in newspaper columns on various subjects and has contributed to leading content companies as a technical writer and a subject matter expert.

Mohan wrote Chapter 12.

Mike Swarm is a Security Solutions Engineer at Juniper Networks. Mike consults with Juniper's technical field and customer communities worldwide on security design practices. Mike has over a decade of experience focused on network security. Prior to Juniper Networks and its NetScreen Technologies acquisition, Mike has been a Systems Engineer at FTP Software and Firefox Communications.

Mike wrote Chapter 10.

Brad Woodberg (JNCIS-FWV, JNCIS-M, JNCIA-IDP, JNCIA-SSL, CCNP) is a Security Consultant at Networks Group Inc. in Brighton, MI. At Networks Group his primary focus is designing and implementing security solutions for clients ranging from small business to Fortune 500 companies. His main areas of expertise include network perimeter security, intrusion prevention, security analysis, and network infrastructure. Outside of work he has a great interest in proof-of-concept vulnerability analysis, open source integration/development, and computer architecture.

Brad currently holds a bachelor's degree in Computer Engineering from Michigan State University, and he participates with local security organizations. He also mentors and gives lectures to students interested in the computer network field.

Brad wrote Chapters 5–8 and contributed to Chapter 13. He also assisted in the technical editing of several chapters.

Neil R. Wyler (JNCIS-FWV, JNCIA-SSL) is an Information Security Engineer and Researcher located on the Wasatch Front in Utah. He is the co-owner of two Utah-based businesses, which include a consulting firm with clients worldwide and a small software start-up. He is currently doing contract work for Juniper Networks, working with the company's Security Products Group. Neil is a staff member of the Black Hat Security Briefings and Def Con hacker conference. He has spoken at numerous security conferences and been the subject of various online, print, film, and tele-

vision interviews regarding different areas of information security. He was the Lead Author and Technical Editor of *Aggressive Network Self-Defense* (Syngress, 1-931836-20-5) and serves on the advisory board for a local technical college.

Neil cowrote Chapter 13.

Contents

Foreword . . . . . **xiii**
Chapter 1 Networking, Security, and the Firewall **1**
Introduction 2
Understanding Networking 3
The OSI Model 3
Moving Data along with TCP/IP 6
Understanding Security Basics 17
Understanding Firewall Basics 26
Types of Firewalls 26
Firewall Ideologies 31
DMZ Concepts 31
Traffic Flow Concepts 35
Networks with and without DMZs 38
DMZ Design Fundamentals 41
Designing End-to-End Security for Data Transmission between Hosts on the Network 42
Traffic Flow and Protocol Fundamentals 43
Summary 44
Solutions Fast Track 45
Frequently Asked Questions 46
Chapter 2 Dissecting the Juniper Firewall **49**
Introduction 50
The Juniper Security Product Offerings 51
Juniper Firewalls 52
SSL VPN 53
Intrusion Detection and Prevention 54
Unified Access Control (UAC) 56
The Juniper Firewall Core Technologies 57
Zones 57
Virtual Routers 57
Interface Modes 58
Policies 58
VPN 59
Intrusion Prevention 59
Device Architecture 61
The NetScreen and SSG Firewall Product Line 63
Product Line 63
Summary 85
Solutions Fast Track 86
Frequently Asked Questions 87
Chapter 3 Deploying Juniper Firewalls **89**
Introduction 90
Managing Your Juniper Firewall 90
Juniper Management Options 91
Administrative Users 93
The Local File System and the Configuration File 95
Using the Command Line Interface 99
Using the Web User Interface 103
Securing the Management Interface 104
Updating ScreenOS 118
System Recovery 119
Configuring Your Firewall for the First Time 121
Types of Zones 122

Virtual Routers 123
Types of Interfaces 123
Configuring Security Zones 126
Configuring Your Firewall for the Network 131
Binding an Interface to a Zone 132
Setting Up IP Addressing 133
Configuring the DHCP Client 133
Using PPPoE 133
Interface Speed Modes 135
Port Mode Configuration 136
Bridge Groups 137
Configuring Basic Network Routing 140
Configuring System Services 142
Setting the Time 143
DHCP Server 145
DNS 147
SNMP 149
Syslog 151
Web Trends 152
Resources 153
Summary 154
Solutions Fast Track 154
Frequently Asked Questions 156

Chapter 4 Policy Configuration 157

Introduction 158
Firewall Policies 158
Theory of Access Control 160
Types of Juniper Policies 162
Policy Checking 164
Getting Ready to Make a Policy 166
Policy Components 167
Zones 167
Address Book Entries 168
Services 172
Creating Policies 176
Creating a Policy 177
Summary 187
Solutions Fast Track 187
Frequently Asked Questions 188

Chapter 5 Advanced Policy Configuration 191

Introduction 192
Traffic-Shaping Fundamentals 192
The Need for Traffic Shaping 192
How Traffic Shaping Works 195
Choosing the Traffic-Shaping Type 196
Deploying Traffic Shaping on Juniper Firewalls 197
Methods to Enforce Traffic Shaping 197
Traffic-Shaping Mechanics 202
Traffic-Shaping Examples 205
Advanced Policy Options 215
Counting 216
Scheduling 222
Summary 228
Solutions Fast Track 228
Frequently Asked Questions 230

Chapter 6 User Authentication 233

Introduction 234
User Account Types 234

Authentication Users 239
Internal Authentication Server 252
Configuring the Local Authentication Server 253
External Authentication Servers 254
Policy-Based User Authentication 269
Explanation of Policy-Based Authentication 269
Configuring Policies with User Auth 270
802.1x Authentication 277
Components of 802.1x 278
Enhancing Authentication 284
Firewall Banner Messages 284
Group Expressions 287
Summary 289
Solutions Fast Track 289
Frequently Asked Questions 291

Chapter 7 Routing 293

Introduction 294
Virtual Routers 294
Virtual Routers on Juniper Firewalls 295
Routing Selection Process 298
Equal Cost Multiple Path 299
Virtual Router Properties 300
Route Maps and Access Lists 306
Route Redistribution 311
Importing and Exporting Routes 311
Static Routing 313
Using Static Routes on Juniper Firewalls 314
Routing Information Protocol 321
RIP Overview 322
RIP Informational Commands 332
Open Shortest Path First 335
Concepts and Terminology 336
Configuring OSPF 341
OSPF Informational Commands 350
Border Gateway Protocol 354
Overview of BGP 354
Configuring BGP 358
BGP Informational Commands 372
Route Redistribution 375
Redistributing Routes in the Juniper Firewall 375
Redistributing Routes between Routing Protocols 376
Redistributing Routes into BGP 380
Policy-Based Routing 383
Components of PBR 383
Summary 393
Solutions Fast Track 393
Frequently Asked Questions 396

Chapter 8 Address Translation 399

Introduction 400
Overview of Address Translation 400
Port Address Translation 401
Advantages of Address Translation 402
Disadvantages of Address Translation 403
Juniper NAT Overview 404
Juniper Packet Flow 405
Source NAT 406
Interface-Based Source Translation 407
MIP 409

Policy-Based Source NAT 417
Destination NAT 428
Policy-Based Destination NAT 433
Summary 446
Links to Sites 446
Solutions Fast Track 446
Frequently Asked Questions 449
Chapter 9 Transparent Mode 457
Introduction 458
Interface Modes 458
Understanding How Transport Mode Works 459
Configuring a Device to Use Transport Mode 462
Transparent Mode Deployment Options 466
Summary 476
Solutions Fast Track 477
Frequently Asked Questions 478
Chapter 10 Attack Detection and Defense. 479
Introduction 480
Understanding Attacks 480
Old Root Causes, New Attacks 482
Unified Threat Management 482
Vulnerability Databases 482
Bug Databases 483
Common Name Dictionary 483
The Juniper Security Research Team 483
Understanding the Anatomy of an Attack 484
The Three Phases of a Hack 484
Script Kiddies 484
Black Hat Hackers 485
Worms, Viruses, and Other Automated Malware 487
Configuring Screen Settings 490
UDP Data Rate Limiting 497
TCP/IP Protocol Anomaly Detection 498
Applying Deep Inspection 501
Deep Inspection Concepts 503
Deep Inspection Planning 505
Getting the Database 507
Using Attack Objects 510
Setting Up Content Filtering 524
Web Filtering 524
Antivirus 532
Antivirus Rules 538
Understanding Application Layer Gateways 540
Applying Best Practices 542
Defense-in-Depth 542
Zone Isolation 542
Egress Filtering 543
Explicit Permits, Implicit Denies 543
Retain Monitoring Data 543
Keeping Systems Updated 543
Summary 544
Solutions Fast Track 545
Frequently Asked Questions 548
Chapter 11 VPN Theory and Usage 551
Introduction 552
Understanding IPSec 552

IPSec Modes 553
Protocols 553
Key Management 555
Security Associations 556
IPSec Tunnel Negotiations 556
Phase 1 557
Phase 2 558
Public Key Cryptography 559
PKI 560
Certificates 560
CRLs 561
How to Use VPNs in NetScreen Appliances 561
Site-to-Site VPNs 561
Policy-Based VPNs 563
Route-Based VPNs 569
Dial-Up VPNs 569
L2TP VPNs 575
Advanced VPN Configurations 576
VPN Monitoring 577
Gateway Redundancy 578
Back-to-Back VPNs 579
Hub and Spoke VPNs 579
Multitunnel Interfaces 580
Summary 580
Solutions Fast Track 581
Links to Sites 584
Mailing Lists 584
Frequently Asked Questions 584

Chapter 12 High Availability 587

Introduction 588
The Need for High Availability 588
High-Availability Options 589
Improving AvailabilityUsing NetScreen SOHO Appliances 591
Failing Over between Interfaces 592
Using Dual Untrust Interfaces to Provide Redundancy 592
Falling Back to Dial-Up 597
Restricting Policies to a Subset When Using the Serial Interface 601
Using IP Tracking to Determine Failover 601
Monitoring VPNs to Determine Failover 604
Introducing the NetScreen Redundancy Protocol 608
Virtualizing the Firewall 608
Understanding NSRP States 610
The Value of Dual HA Links 612
Building an NSRP Cluster 613
Connecting the Firewalls Directly to the Routers 613
Connecting the Firewalls to Routers via Switches 615
Cabling for a Full-Mesh Configuration 616
Using Directly Connected HA Links 617
Connecting HA Links via Switches 618
Adding a NetScreen to an NSRP Cluster 619
Synchronizing the Configuration 621
Determining When to Fail Over: The NSRP Ways 624
Using NSRP Heartbeats 624
Using Optional NSRP Monitoring 626
Using NSRP Interface Monitoring 627
Using NSRP Zone Monitoring 629
Using NSRP IP Tracking 630
Reading the Output from get nsrp 638

Looking into an NSRP Cluster 638
Using NSRP-Lite on Midrange Appliances 641
Basic NSRP-Lite Usage 642
Working with Local Interfaces in an NSRP-Lite Setup 646
Creating Redundant Interfaces 652
Taking Advantage of the Full NSRP 654
Synchronizing State Using RTO Mirroring 655
Setting Up an Active/Active Cluster 657
Implementing a Full-Mesh Active/Active Setup 664
Failing Over 670
Failing Over Virtual Systems 671
Avoiding the Split-Brain Problem 673
Avoiding the No-Brain Problem 674
Configuring HA through NSM 676
Creating a Cluster 676
Adding Members to the Cluster 677
Configuring NSRP Parameters 680
Configuring VSD 682
Summary 682
Solutions Fast Track 683
Frequently Asked Questions 687

Chapter 13 Troubleshooting the Juniper Firewall 689

Introduction 690
Troubleshooting Methodology 690
Troubleshooting Tools 692
Network Troubleshooting 706
Debugging the Juniper Firewall 706
Debugging NAT 712
Debugging VPNs 713
Policy-Based VPNs 714
Route-Based VPNs 714
Debugging NSRP 715
Debugging Traffic Shaping 715
NetScreen Logging 717
Traffic 717
Self 718
Event 718
Summary 720
Solutions Fast Track 720
Frequently Asked Questions 723

Chapter 14 Virtual Systems 725

Introduction 726
What Is a Virtual System? 726
Virtual System Components 726
How Virtual Systems Work 728
Classifying Traffic 728
Virtual System Administration 729
Configuring Virtual Systems 729
Creating a Virtual System 729
Network Interfaces 731
Virtual System Profiles 739
Summary 741
Solutions Fast Track 742
Frequently Asked Questions 743

Index 745

Foreword

As we expand networks to include new services, we must continually strive to secure them. It is not an inherently easy thing to do.

First, we need to balance growth and total security without duplicating operations. Second, our networks need to support the mobility of our workforces as the number of remote sites that are connected continues to multiply. And finally, while one cannot predict what will be needed for tomorrow, we must build in the flexibility to adapt to whatever unknown priorities may arise in the near future.

These challenges are why Juniper Networks is so focused on providing mission-critical products for today with the capacity to adapt for tomorrow's shifting priorities. And the authors of this book have done a wonderful job collecting and collating what we need to know to make intelligent networking decisions.

Delivering performance and extensibility is one of the key traits of Juniper Networks. We allow networks to grow without duplicating operations, all the while securing them from multiple levels of potential attack. As you read through this book, please remember that performance and flexibility are fundamental to how Juniper Networks' VPN, firewall, and intrusion prevention products are built and how they will work for you.

—Scott Kriens, CEO, Juniper Networks
November 2006

Chapter 1

Networking, Security, and the Firewall

Solutions in this chapter:

- **Understanding Networking**
- **Understanding Security Basics**
- **Understanding Firewall Basics**

- ☑ **Summary**
- ☑ **Solutions Fast Track**
- ☑ **Frequently Asked Questions**

Introduction

Every organization that connects to the Internet has business partners and other external entities, requiring them to use firewall technology. Firewalls are a required component of your data network, and provide a protective layer of security. Security risks have greatly increased in recent years, and so the call for a stronger breed of firewall has been made. In the past, simple packet filtering firewalls allowing access to your internal resources have helped to mitigate your network's risk. The next development was stateful inspection, allowing you to monitor network sessions instead of single packets. Today's risks are far greater, and require a new generation of devices to help secure our networks' borders from the more sophisticated attacks. The industry calls these firewalls L4/L7 firewalls. L4/L7 stands for Layer 4 through Layer 7, which refers to layer 4 through layer 7 of the OSI security model. These firewalls are often equipped with IPS, and are generally known as firewalls with application layer support. Later in this chapter, we delve deeper into L4/L7 firewalls.

Firewalls police your network traffic. A firewall is a specialized device that allows or denies traffic based upon administratively defined policies. They contain technologies to inspect your network's traffic. This technology is not something that is exclusive to firewalls, but firewalls are designed specifically for inspecting traffic, and therefore do it better than any other type of device. Many networks can have millions of packets transverse it in a short period of time. Some firewall models are built upon software, like firewalls from Cisco Systems, Checkpoint, and Secure Computing. Conversely, such as with the Juniper Networks NetScreen firewall, they can be constructed around a purpose-built operating system and hardware platform.

Juniper Networks (Juniper) NetScreen firewall appliances were originally designed to support 100-Mbps and 1-Gbps connection speeds of early secure Internet service providers such as Korea Telecom, as well as customers like NASA. Performance of the stateful packet inspection method of firewalling was crucial for these early deployments. Therefore, Juniper firewalls are engineered much like layer 3 switches rather than software only–based firewalls.

The Juniper NetScreen firewall product line has complete offerings from the home office to the carrier-class networks. In this chapter, we will review networking basics. Security requires a strong basic knowledge of networking protocols. In our first section, "Understanding Networking," we will look at networking from a top-down approach. This section starts with the basic ideas of networking models and then works into full networking communications. We will also discuss the components and prerequisites of IP addresses and how they are divided up to make networks.

We will next look at networking in general by breaking it down to a layered approach. This will help you understand the flow of networking. Each specific layer in the networking model has a purpose. Working together, these layers allow for data to seamlessly pass over the network between systems. An example of browsing a Web site will be used. You will see all of the effort it takes just to fetch a Web page. We will then focus on the TCP/IP protocol suite. This is the most commonly used networking protocol, and is the protocol used for Internet communications. Finally, we will take a look at network security fundamentals.

There are many important concepts to be aware of for information security. This will help you understand some network design considerations and the background behind them.

Layered security is now the tried-and-true method of protecting your organization. Many organizations choose to implement a variety of technology from a variety of manufacturers in a variety of locations. As an example, it is typical to see Internet-facing firewalls to be of brand A, while the internal, corporate-facing firewalls are brand B. At the same time, intrusion prevention technology from brand C is deployed in the DMZs (demilitarized zones), and antivirus and anti-spam technology is then deployed by brand D. By choosing the best-of-breed for each layer, you are insuring a higher degree of protection than you could if you chose to pick a single vendor for all layers. Juniper NetScreen firewalls are designed to fit specific layers, and they are created to provide protection and performance at these specific layers. It *is* possible, however, to deploy a Juniper NetScreen firewall in a layer that it was not designed for, making your protection and performance suffer.

Understanding Networking

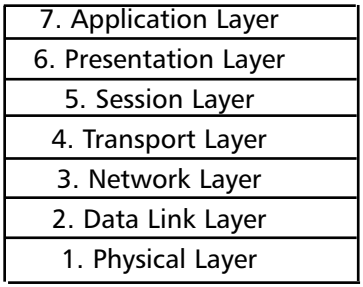
To understand networking is to understand the language of firewalls. A firewall is used to segment resources and limit access between networks. Before we can really focus on what a firewall does for us, we need to understand how networking works. Today in most environments and on the Internet, the protocol suite TCP/IP (Transmission Control Protocol/Internet Protocol) is used to transport data from here to there. We will begin this chapter by looking at networking as a whole with a focus on the Open System Interconnection (OSI) model.

The OSI Model

The OSI model was originally developed as a framework to build networking protocols on. During the time when the Internet was being developed, a protocol suite named TCP/IP was also developed. TCP/IP was found to meet the requirements of the Internet's precursor, ARPANET. At this point, TCP/IP was already integrated into UNIX, and was quickly adopted by the academic community as well. With the advent of the Internet and its widespread usage, TCP/IP has become the de facto standard protocol suite of internetworking today.

The OSI model consists of seven distinct layers. These layers each contain the fundamental ideas of networking. In Figure 1.1, we can see the way that the seven layers stack on top of each other. The idea is that each upper layer is encapsulated inside of each lower layer. So ultimately, any data communications are transformed into the electrical impulses that pass over the cables or through the air that surrounds us. Understanding the OSI model gives you knowledge of the core of networking. In many places throughout this book, the OSI model is used to create a visual representation of networking.

Figure 1.1 The Seven-Layer OSI Model



The reality, however, is that the OSI model is just a reference model that protocols are based upon. The next section, called "Moving Data Along with TCP/IP," demonstrates how some of the layers blur together. All in all, the OSI model is a great tool to help anyone understand networking and perform troubleshooting. Over the years, the OSI model has served as a reference for all protocols that have been developed. Almost every book, manual, white paper, or Web site that talks about networking protocols references the OSI model. It is important to have a baseline when discussing every topic.

For example, let's compare cars and trucks. They are effectively the same device. Both are used to get from here to there, but they are designed very differently. A truck has a sturdier frame to allow it to tow heavy loads. A car is smaller and is designed to transport people. While these devices are very different, they still have common components: wheels, doors, brakes, and engines. This is much like the different components of a network protocol, which is essentially a vehicle for data. Networking protocols have components to help get the data from here to there, like wheels. They have components to control the flow of data, like brakes. These are all requirements of any protocol. Using and understanding the OSI model makes protocol usage and design easier. Whether TCP/IP or IPX/SPX, most protocols are built around the same framework (model).

Layer 7: The Application Layer

The application layer contains application data. This is the layer at which applications communicate to one another. The reason for all of the other layers is essentially to transport the messages contained at the application layer. When communicating with each other, the applications use their own language, as specified by that application's standard. A perfect example of an application protocol is Hypertext Transfer Protocol (HTTP). HTTP is used to send and receive Web content. When HTTP is used to pass data from server to client, it employs something called HTTP *headers*. HTTP headers are effectively the language of HTTP. When the client wants to request data from a server, it issues a request to get the content from the server. The server then responds with is headers and the data that was requested. This communication cycle is performed at the application layer. Other examples of application layer protocols are File Transfer Protocol (FTP), Domain Name Service (DNS), Telnet, and Secure Shell (SSH).

Layer 6: The Presentation Layer

The presentation layer controls the presentation or formatting of the data content. At this point in the OSI model, there is no data communication per se. The focus of this layer is having a common ground to present data between applications. For example, let's take image files. Billions of image files are transferred every day. Each of these files contains an image that ultimately will be displayed or stored on a computer. However, each image file must be the proper specified file format. This way, the application that reads the image file understands the type of data and the format contained in it. A JPEG file and a PNG file may contain the same image, but each uses a separate format. A JPEG file cannot be interpreted as a PNG, and vice versa. Additionally, file-level encryption occurs at the presentation layer.

Layer 5: The Session Layer

The session layer controls sessions between two systems. It is important to have sessions since they are the core of any communications for networking. If you did not have sessions, all communications would run together without any true idea of what is happening throughout the communication. As you will see in the following, TCP/IP really has no session layer. Instead, the session layer blends together with the transport layer. Other protocols such as NetBIOS, used on Microsoft networks, use the session layer for reliable communications.

Layer 4: The Transport Layer

The transport layer provides a total end-to-end solution for reliable communications. TCP/IP relies on the transport layer to effectively control communications between two hosts. When an IP communication session must begin or end, the transport layer is used to build this connection. The elements of the transport layer and how it functions within TCP/IP are discussed in more detail later in the chapter. The transport layer is the layer at which TCP/IP ports listen. For instance, the standard port which HTTP listens on is TCP Port 80, although HTTP could really run on any TCP port; this is the standard. Again, there is no difference between TCP port 80, 1000, or 50000; any protocol can run on it. Standardized port numbers are used to help ease the need to negotiate the port number for well-known applications.

Layer 3: The Network Layer

When packets are sent between two stations on a network, the network layer is responsible for the transportation of these packets. The network layer determines the path and the direction on the network in order to allow communications between two stations. The IP portion of TCP/IP rests in this part of the OSI model. IP is discussed in detail in the following section.

Layer 2: The Data Link Layer

Layer two, or the data link layer, is the mechanism that determines how to transmit data between two stations. All hosts that communicate at this level must be on the same physical

network. The way in which the transmission of data at this level is handled is based upon the protocol used. Examples of protocols at the data link layer are Ethernet, Point-to-Point Protocol (PPP), Frame Relay, Synchronous Data Link Control (SDLC), and X.25. Protocols such as Address Resolution Protocol (ARP) function at the Data Link Layer.

Layer 1: The Physical Layer

The last but most important layer of the OSI model is the physical layer. The physical layer consists of the objects that connect stations together physically. This layer is responsible for taking the bits and bytes of the higher layers and passing them along the specified medium. You have probably already heard of many examples of the physical layer, such as Cat5 cable, T1, and wireless.

Moving Data along with TCP/IP

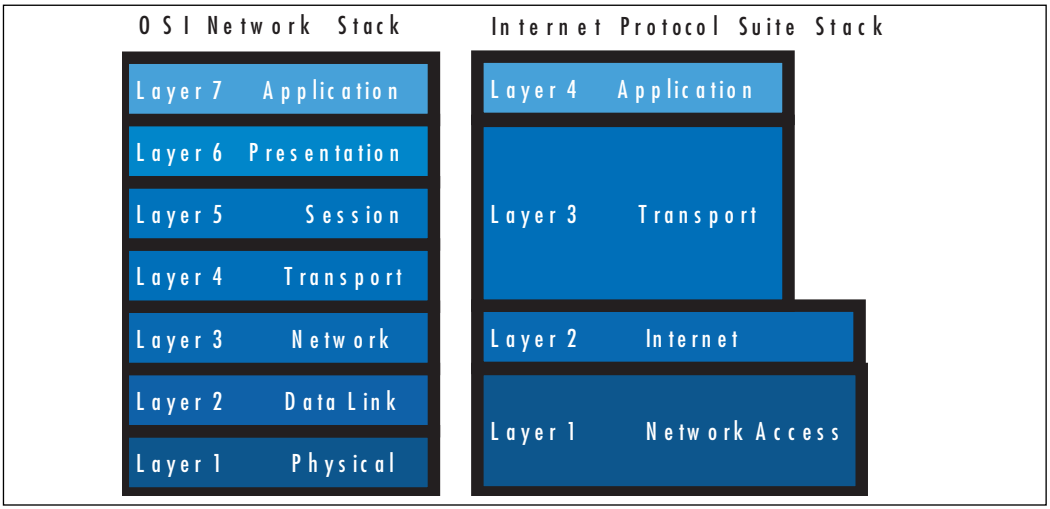
On the Internet and most networks, TCP/IP is the most commonly used protocol for passing along network data. At the time of its development, TCP/IP used a very advanced design. Decades later, TCP/IP continues to meet the needs of the Internet. The most commonly used version of IP used today is version 4, the version covered in this book. The next generation IP, version 6, is starting to be used much more throughout the world. Many vendors (including Juniper Networks, Cisco, Microsoft, and Apple) are developing software productsthat support the new IP version 6 standard.

Over the course of this section, we will cover how systems use TCP/IP to interact, and we will review the IP protocol and how its protocol suite compares to the OSI model. We will also discuss how IP packets are used to transmit data across networks, and we will examine the transport layer protocols TCP and User Datagram Protocol (UDP) and how they are used to control data communications in conjunction with IP. Finally, we will wrap up the discussion of TCP/IP with information about the data link layer.

Understanding IP

The Internet Protocol is used to get data from one system to another. The IP protocol sits on the third layer of the OSI model: the network layer. When you need to send data across a network, that data is encapsulated in a packet. A packet is simply a segment of data that is sent across the network. In TCP/IP, however, there are not seven true layers, as there are in the OSI model (see Figure 1.2 for a comparison of TCP/IP and OSI model layers).

Figure 1.2 OSI Model Layers vs. TCP/IP Layers



When an application needs to pass its communication to another system on the network, it passes its information down the protocol stack. This is the process that creates an IP packet.

Let's look at an example of IP connectivity. We will be referencing the TCP/IP model since it will be easier to understand for this example. Remember that the TCP/IP model is a condensed version of the OSI model. Use Figure 1.2 to reference the steps of the OSI model on the left to the TCP/IP model on the right. You can use your Web browser to connect to www.syngress.com and view the series of events that occur during a network (in this case, the Internet) connection. We will look at the course of action that happens for the first packet that is created for this connection.

First, enter the address in the Web browser and then press **Enter**. The browser will make a request to get the data from the server. This request is then given to the transport layer where it initiates a session to the remote machine. To get to the remote machine, the transport layer sends its data to the network layer and creates a packet. The data link layer's job is to get the packet across the local network. At this point, the packet is called a *frame*. At each junction point between systems and routing devices, the data link layer makes sure that the frame is properly transmitted. The physical layer is used during the entire connection to convert the raw data into electrical or optical impulses.

When the end station receives the packet, that station will convert the packet back to the application layer. The electrical impulses are changed at the physical layer into the frame. The frame is then decapsulated and converted to individual packets. Because the packet is at its end destination, the network layer and transport portions of the packet are removed and then the application data is passed to the application layer. That sounds like a lot of work for just one packet to transverse the Internet, but all of this happens on a broadband connection in 30 milliseconds or less. This, of course, is the simplified version of how all of it occurs. In the following sections, we will expand on this example and show you what happens behind the scenes when two stations have a network conversation.

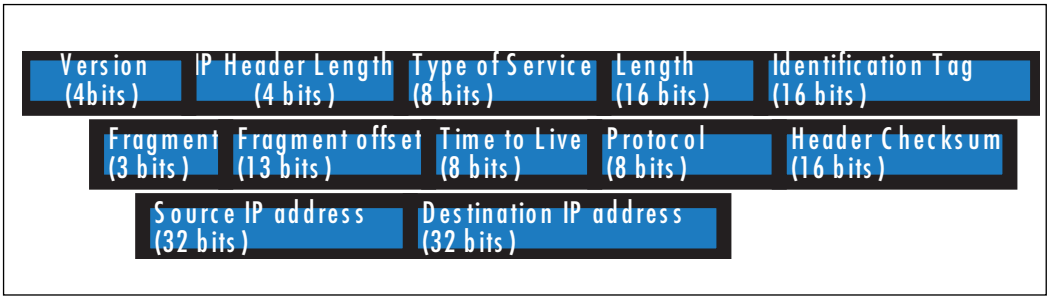
The following list provides a rundown of the phases of connectivity:

1. The URL www.syngress.com is entered into the browser.
2. The user presses **Enter** and forces the browser to connect to the Web site.
3. The browser makes a request to the server.
4. The browser request is handed to the transport layer.
5. The transport layer initiates a session to the remote server.
6. The transport layer passes its request to the network layer.
7. The network layer creates a packet to send to the remote server.
8. The data link layer takes the packet and turns it into a frame.
9. The frame is passed over the local network by the physical layer.
10. The physical layer takes the frame and converts it into electrical or optical impulses.
11. These impulses pass between devices.
12. At each junction point or router, the packet is transformed to the data link layer.
13. The packet is taken from the data link layer to the network layer.
14. The router looks at the packet and determines the destination host.
15. The router forwards the packet to the next and all subsequent routers until it reaches the remote system.
16. The end station receives the packet and converts it back through the layers to the application layer.
17. The remote system responds to the client system.

IP Packets

As discussed in the previous sections, IP is essentially used to transfer data from one system to another. The anatomy of IP is very straightforward. In Figure 1.3, you can see what exactly makes up an IP packet header. An IP packet contains the very important application data that needs to be transported. This data is contained in the last portion of the packet. The IP portion of a packet is called the IP header. It contains all of the information that is useful for getting the data from system to system. The IP header includes the source and destination IP addresses.

Figure 1.3 IP Packet Header Contents



So the question remains, "how do IP packets actually get from system to system?" Let's reference our previous example of browsing to www.syngress.com. When the IP packet is formed, it includes the source IP address (the IP address of the client system making the request). This is like the return address on an envelope that tells the recipient where to send return mail to. The packet also receives the destination address of the Web server being contacted. There are other parts that are set in the IP header, but are not germane to this discussion. After the packet is created, it is sent to the originating system's routing table. The routing table is referenced and then the operating system determines which path to send this packet to. In routing, each system that receives the packet determines the next location or *hop* to send the packet to. So when sending information or requests across the Internet, there may be 15 hops or routers to go through before you get to the final system you are trying to connect to. Simply stated, a router is a system whose primary function is to route traffic from one location to another. As each router receives a packet, it determines the next best location to send it to.

This, of course, is very simplified since there are millions of routers on the Internet. Once the destination system receives the IP packet, it formulates a response. This is then sent back to the client system. The IP header contains the source address of the server that received the first packet and then the destination address of the initiating client machine. This is the fundamental basis of IP communications.

One of the confusing things about IP is that IP packets are not just used to transport data; the IP protocol suite does more than that. If you refer back to Table 1.1, you can see a field called *protocol*. This determines which IP protocol the packet is using. All of the available IP protocols are specified in RFC 1700. Table 1.1 is a short reference of the IP protocols we will be discussing in this book. For example, if the packet was UDP, it would be using IP protocol 17, and if the packet was IP Security (IPSec) ESP, it would be using IP protocol 50.

Table 1.1 IP Protocol Suite

Protocol Number	Name	Protocol
1	ICMP	Internet Control Message Protocol
4	IP	IP to IP Encapsulation
6	TCP	Transmission Control Protocol
17	UDP	User Datagram Protocol
50	ESP	Encapsulating Security Payload
51	AH	Authentication Header

One of the most important protocols in the IP protocol suite is the Internet Control Messaging Protocol (ICMP). ICMP is used as a messaging protocol to give information to the source or destination machine that is engaging in IP communications. Table 1.2 lists all of the commonly used ICMP types and codes. To give an example of ICMP, let's look at the common application *ping*. Ping is an application that is on pretty much any operating system, including Screen OS, the underlying security operating system of Juniper NetScreen firewalls. It is used to test if a host is responsive from a network perspective. When you ping a host, an IP packet is generated that has the source IP address of the requesting system, and the destination IP address of the system you are trying to contact. This packet then has an ICMP type of eight and a code of zero. The destination system then would receive the packet and recognize that the IP packet is *echo* or *echo request packet*. It then creates an ICMP packet that is a type zero code zero. This is an *echo reply packet*, acknowledging the original request.

Table 1.2 ICMP Types and Codes

Type	Name
0	Echo Reply
Codes	**Name**
0	No Code
Type	**Name**
3	Destination Unreachable
Codes	**Name**
0	Network Unreachable
1	Host Unreachable
2	Protocol Unreachable
3	Port Unreachable

Devices use ICMP for other reasons as well. If a system had a route in its routing table that specified a host could be found at a location that did not exist, the router it points to would send an ICMP message to the initiating host. That router would send a type three code zero or code one message specifying that the network or host is not available. Now apply that to the Internet and all of those millions of routers out there. This makes the ICMP protocol very helpful for notifying users when there is a problem with getting IP packets from one location to another.

What Does an IP Address Look Like?

IP addresses are 32 bits in length. They consist of four eight-bit numbers. An example of an IP address is 1.2.3.4. This looks like a very simple format, but it has a great deal of meaning. Each of the four numbers can contain a value from 0 to 255. IP addresses are allocated in blocks or subnets. A subnet is a grouping of IP addresses based upon a subnet mask. There are three major types of IP address blocks: class A, B, and C. Each class is determined based upon the three leading bits for each number. The class A grouping of IP addresses all start with the binary digit 0. The class B grouping of IP addresses all start with binary digits 10 (not read as ten). Finally, the class C grouping of IP addresses all starts with binary digits 110 (not read as one-hundred ten). In Table 1.3 you can see all of the ranges of IP addresses based upon class. There are two other classes of IP addresses, classes D and E, which have special functions not covered in this book.

Table 1.3 IP Address Ranges by Class

Class	Address Range
A	0.0.0.0 to 127.255.255.255
B	128.0.0.0 to 191.255.255.255
C	192.0.0.0 to 223.255.255.255
D	224.0.0.0 to 239.255.255.255
E	240.0.0.0 to 255.255.255.255

You can also use your own local computer to look at your IP address. We will use both a Windows system and a UNIX-based system as an example. Open up a DOS window on your Microsoft Windows system, then enter the command *ipconfig*. An example of this is shown in Figure 1.4. You can also do the same thing on a UNIX-based system by using the command *ifconfig* (shown in Figure 1.5).

Figure 1.4 Microsoft Windows *ipconfig* Output

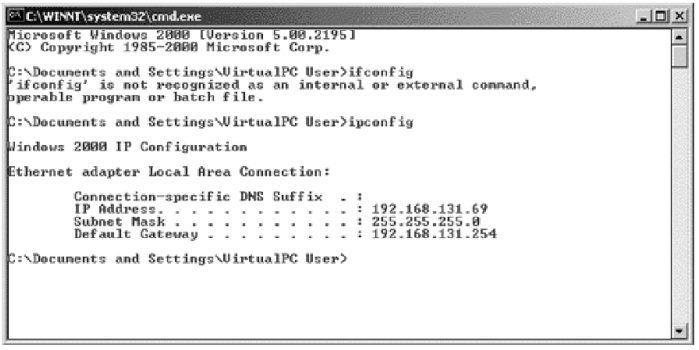


Figure 1.5 UNIX *ifconfig* Output

```
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 10.6.0.123 netmask 0xffffff00 broadcast 10.6.0.255
        ether 00:0d:93:8c:62:2e
        media: autoselect status: active
        supported media: autoselect
```

IP Address Allocation

When creating a network, deciding on IP address allocation is very important. But with billions of options, how does one decide? The Internet Assigned Numbers Authority, or IANA, is responsible for allocating IP addresses. They determine which organizations get which IP address ranges. They are also responsible for conserving IP addresses and planning for future uses for IP addresses. Does this mean you need to contact them to get IP addresses? Unless you are starting your own Internet service provider (ISP) the size of Qwest or SBC, you do not need to contact them. Your ISP will always assign any Internet or public IP addresses, and for private IP address networks you would use the IP addresses specified in RFC 1918. See Table 1.4 for a list of the private IP address ranges. A non-Internet routable IP is an IP address that is not routed on the Internet. If a packet was to leave your network with a source or destination IP address in one of these ranges, it might be dropped by an ISP firewall or router. Even if it did make it to the remote network, the machine on that side would not be able to route the traffic back to the private IP address which the packet came from because it is not unique and not publicly routable.

Table 1.4 RFC 1918 IP Address Ranges

Class	Address Range
A	10.0.0.0 to 10.255.255.255
B	172.16.0.0 to 172.31.255.255
C	192.168.0.0 to 192.168.255.255

NAT and Private IP Addresses

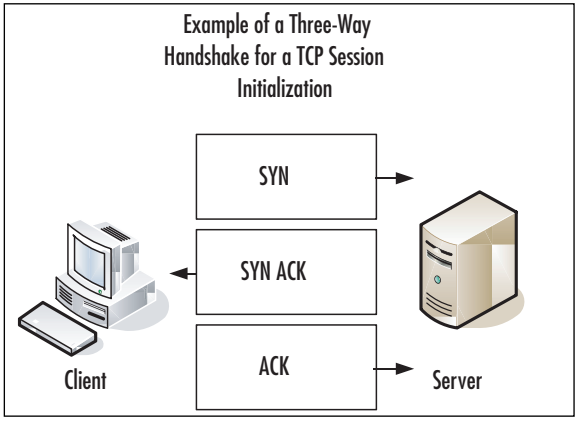
Most companies need to access Internet resources while preserving Internet IP addresses. The solution is Network Address Translation, or NAT. NAT is used to hide your private IP address behind a public IP address. This allows private IP-addressed systems to access publicly addressed systems. NAT also provides a layer of security by hiding the real IP addresses of your internal network. A gateway device such as a Juniper NetScreen firewall performs NAT for IP packets that pass through the device. Once the firewall receives an IP packet with the source IP address, it changes the private IP address into a public IP address. When the Juniper NetScreen firewall receives the return packet, it translates the new destination address to the private IP address. Two types of NAT exist: NAT source and NAT destination.

TCP Communications

The Transmission Control Protocol is used to control the creation and form of data transfer connections. TCP is one of two transport layer protocols used as part of the TCP/IP protocol suite. TCP is designed to provide many functions, mostly based on reliability. TCP is used for applications that require reliability over speed. When talking about speed at this level, we are talking about calculations of milliseconds or less. TCP functions as a stateful protocol. This means that during the communications, the connection has specific states in which it functions. There is a clear beginning, middle, and end to a TCP connection.

When a TCP session begins, it goes through a three-way handshaking process. Inside of a TCP header, options (called flags) are set. These flags identify the type of TCP message that has been sent. The three-way handshake process is shown in Figure 1.6. Let's continue to use our earlier example of employing your Web browser to access www.syngress.com. When your Web browser attempts to make its connection to the Web server, it attempts to open a connection to TCP port 80. A port is a particular communications channel specific to a particular application. TCP port 80 is the default port for HTTP.

Figure 1.6 TCP Session Initialization



The first packet that is sent to the Web server is a SYN packet, which is used to synchronize a connection between two hosts. This packet is also sent with a sequence number that is used to identify the packet inside of this connection. This sequence number is to be used for the initiating systems packets. Next, the Web server that receives the packet acknowledges it. To do this, the server creates and sends a packet with the TCP flags SYN and ACK. A packet that has the ACK (or acknowledgement) flag set is sending a message to the other system that says, "I have received your packet." A sequence number is also given to this packet that is independent of the sequence number associated with the initiating system's sequence number. The system that initiated the connection now sends an ACK packet to acknowledge the connection. The ACK packet has a sequence number that is incremented since it is the second packet that has been sent from this system. The TCP session has now been created and the requested data from the Web server can begin to pass to the client.

The data that was requested is divided into packets by TCP. The client sends a TCP packet with the ACK flag for each part of the data. Again, each packet sent from the client has a sequence number that is incremented by one. The sequence number is used to identify all of the packets of a TCP exchange. If, for example, a client receives packets with sequence numbers 6, 7, 8, and 10, but never receives packet 9, the client will request that packet 9 be re-sent from the server. On the client, all of the packets would be reordered before passing the data back to the application. When the connection is completed, the server system would send a packet with the FIN flag. This indicates that the connection is finished. The client would then send an ACK packet back to the server acknowledging that the conversation has completed.

UDP Communications

The User Datagram Protocol is a connectionless protocol that is designed to stream data. When a UDP connection occurs, there is no beginning, middle, or end to the conversation. Data simply begins to flow between the two systems. UDP is a very simple protocol and is used when speed is an issue. UDP packet receipt is not verified. An example of a use of the UDP protocol is DNS queries. When you attempt to use your Web browser to access www.syngress.com, it must first resolve the name to an IP address. This would require a DNS query. The query is sent over a single UDP packet. The DNS server would then respond by telling the originating system the IP address of the Web server. Because the UDP response is faster than setting up a TCP session, UDP makes sense in these situations. Another example of using UDP is Voice over IP (VoIP). The downfall, of course, is the lack of reliability, so you may have to employ other methods to guarantee delivery.

What Is a Port?

Both TCP and UDP support the use of ports. But what exactly is a port? Let's look at an example that can help further explain this. When you turn on your television, you get a picture and sound. Every time you change the channel, each new channel contains different content. This is much like a TCP or UDP port. Each port contains a specific type of content

or application. When you tune to that port, you can access those specific resources. Theoretically, you can put any application on any port, but by specifying specific ports for specific applications, you can always be assured of the type of content you will find on a specific port.

This is why a specification of well-known ports has been established. Table 1.5 lists well-known TCP and UDP ports. Using our earlier television example, this is much like a channel lineup. If television programming could appear on any television channel, there would be a lot of confusion about which programming you were watching. When you use your television, the service provider gives you a channel lineup. This lineup is specified so that you know which channel is which. Most Web servers serve data over port 80. Again, they can serve the data over any port, but it would be very hard to get the content if you did not know which port to use.

Table 1.5 Well-Known TCP and UDP Ports

Well-Known TCP Ports		Well-Known UDP Ports	
FTP	21	DNS	53
SSH	22	DHCP-Relay	67
Telnet	23	TFTP	69
SMTP	25	NTP	123
HTTP	80	IKE	500
IMAP	143	Syslog	514
HTTPS	443	H.323	1719

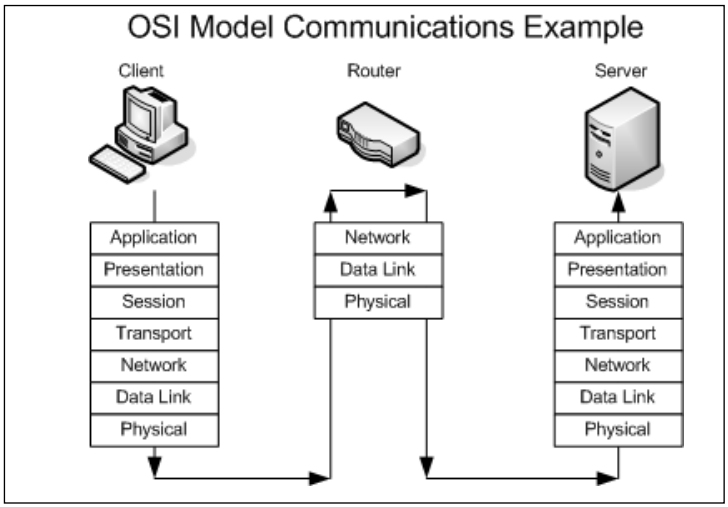
Data Link Layer Communication

The last part of networking we are going to discuss is the data link layer, or layer two. This layer is essentially the protocol that operates on the specific physical medium. Each of the following function differently on the data link layer: Ethernet, ATM, Frame Relay, HDLC, SDLC, PPP, and Serial Line Internet Protocol (SLIP) to name a few. In this section how Ethernet functions will be focused on. As of the time of this writing the main layer two protocol that is used by NetScreen firewalls is Ethernet.

Ethernet is the most commonly used medium today in corporate networks. It is inexpensive to purchase, easy to set up, and can operate at incredible speeds. The data link layer is used to communicate across the local medium. Figure 1.7 shows the breakdown of the use of layers and where they take place during system-to-system communication. When systems need to talk over Ethernet, they cannot use IP addresses, because Ethernet is at a lower level and it is used to move IP between layer three devices. So each device on an Ethernet segment uses a Media Access Control (MAC) address. When a station needs to have a conversation, the source and destination systems use their MAC addresses to identify each other. Each

manufacturer is assigned a range to use when creating Ethernet adapters. Then each individual adapter is given a unique number to create the MAC address.

Figure 1.7 A Layered Look at Network Communications



Because systems communicate via IP, but need to talk over Ethernet (which requires the use of MAC addresses), there has to be a way to resolve an IP to a MAC address. The method used is called the Address Resolution Protocol. For example, if system A, which has an IP address of 192.168.1.10, wanted to view the Web pages on system B, which has an IP address of 192.168.1.25, before the communications can begin, system A must learn the MAC address of system B. System A broadcasts a request over the local broadcast domain asking who has the IP address 192.168.1.25. A broadcast is a communication that is sent out to every system that is within a broadcast area. All of the systems in the broadcast area get this request and the system with the requested IP address responds with a unicast message that specifies it has the IP address of 192.168.1.25 and also provides its MAC address.

Because almost everyone uses a computer today, a typical company can contain at least 20 computers or more. There are many ways to connect computers together. If you have just two systems, you can connect them with just a crossover Ethernet cable. A crossover cable is an Ethernet cable that allows two systems to directly connect to each other back to back. If you have two to four computers, you could use a hub or bridge. If you have four or more computers, you will likely want to use a switch. A hub or bridge is a device that connects several systems together. When two systems want to access the Ethernet media to transfer data, their communications take up the use of the media while they are talking. If a third system wants to talk over the network, it simply starts talking and the data frames will collide with those of the already ongoing communication. An Ethernet segment where the media is shared between them is called a *collision domain*. Switches, however, do not have this problem. When two systems begin a network conversation on a network with a switch, the packets

are isolated and the switch prevents packets from colliding. If a system was to broadcast, however, the broadcast would be sent to every system connected to the switch. When the switch sends the data between two hosts, it sends it in such a way that other network conversations are not interrupted.

Understanding Security Basics

The first key to understanding network security is to understand networking. We hope, the previous section has started you on the path to understand networking. Just be patient while reading this book. There may be many new concepts you have never heard of before. Working with these technologies over time will help solidify your knowledge. You can also reach for other Syngress Publishing books on the topic of information security (infosec) that can help build on your body of knowledge. In this section, we discuss basic security concepts that will prepare you for the final section about firewalls, and focus on some of the different aspects of what it takes to have a secure organization. As you will see, there are no hard and fast rules about what it really takes to make your network secure. I have been to many organizations that would fall well below the line I would call good security practices. However, some of those same organizations have gone years without a security breach. On the other hand, I have seen other companies spend much more on their security and have more problems with break-ins and data loss.

The Need for Security

Enterprise security is the hottest technology trend today. Every aspect of a company's data infrastructure has a need for security. With ever-growing, ever-evolving networks in all organizations, managing security has become harder. For many companies, the operating budget for security is less than one percent of their total budget. When it comes down to purchasing security products, firewalls are the core product used to secure the enterprise network. However, firewalls should by no means be the only method used to secure your network, but if used effectively, they can mitigate the risks of network security breaches and data loss. With integrated technologies such as antivirus software, deep packet inspection, Uniform Resource Locator (URL) filtering, and virtual private networks (VPNs), the firewall can provide a host of security applications all in one system. Nevertheless, as the old saying goes, never put all your eggs in one basket.

Introducing Common Security Standards

Security and network professionals use a number of currently accepted procedures and standards to conduct business and ensure we are following the accepted practices for security and access. Although we have a responsibility as network and systems administrators to try to attain perfection in the availability and integrity of our data, we also have constraints placed on us in accomplishing those tasks. These constraints include budgets, physical plant capa-

bility, and the training of users and technicians to maintain the security and integrity of the data. These constraints do not relieve us of our responsibility of maintaining the data safely and securely. To that end, we currently employ some accepted standards for security that help us perform our tasks to the best possible level. In this section, we remind you of the common security standards and briefly discuss them:

- **Authentication, authorization, and auditing (AAA)** AAA use is required in security operations for creating and maintaining the method of authenticating users and processes, and validating their credentials prior to allowing access to resources. It is also the method we use to grant access or deny access to the resource. Auditing of activity is a crucial part of this function.
- **Confidentiality, integrity, and availability (CIA)** CIA is the originally defined process that establishes the goals we have used to try to protect our data from unauthorized view, corruption, or unauthorized modification, and to provide constant availability. Over the past few years, the CIA processes have expanded to include a more comprehensive guideline that also includes the process of defining risk and use of risk management tools to provide a more complete method of protection.
- **Least privilege** This concept is used by the security planners and teams to define the levels of access to resources and the network that should be allowed. From a security standpoint, it is always preferable to be too restrictive with the capability to relax the access levels than to be too loose and have a breach occur.

Remember, too, that the security process involves a three-tiered model for security protection:

- **Computer security**, including the use of risk assessment, the expanded CIA goals, and enterprise planning that extends throughout the entire enterprise, rather than to just a portion of it.
- **Physical security**, in which we must build and include physical access systems and coordinate them with our network access systems.
- **Trusted users**, who become an important cog in maintaining the integrity of our security efforts.

Common Information Security Concepts

A generic dictionary definition of *security* (taken from the *American Heritage Dictionary*) is, "freedom from risk or danger; safety." This definition is perhaps a little misleading when it comes to computer and networking security, because it implies a degree of protection that is inherently impossible to achieve in the modern connectivity-oriented computing environment.

For this reason, the same dictionary provides another definition specific to computer science: "The *level to which* a program or device is safe from unauthorized use" (emphasis added). Implicit in this definition is the caveat that the objectives of security and accessibility—the two top priorities on the minds of many network administrators—are, by their very nature, diametrically opposed. The more accessible your data, the less secure it is. Likewise, the more tightly you secure your data, the more you impede accessibility. Any security plan is an attempt to strike the proper balance between the two.

Defining Information Security

Over the last couple of decades, many companies began to realize that their most valuable assets were not only their buildings or factories but also intellectual property (Known as IP in the industry) and other key business information. Company managers, who are used to dealing with risk in their business activities, started to worry about what might happen if this information fell into the wrong hands, perhaps a competitor's. In addition, the Sarbanes-Oxley Act of 2002 (a.k.a. SOX or SARBOX) generally legislated IT governance and controls, thrusting information security to the front stage in publicly traded companies

For a while, this risk was not too large, due to how and where that information was stored. *Closed systems* was the operative phrase. Key business information, for the most part, was stored on servers accessed via terminals or terminal emulators and had few interconnections with other systems. Any interconnections tended to be over private leased lines to a select few locations, either internal to the company or to a trusted business partner.

However, over the last five to seven years, the Internet has changed how businesses operate, and there has been a huge acceleration in the interconnectedness of organizations, systems, and networks. Entire corporate networks have access to the Internet, often at multiple points. This proliferation has created risks to sensitive information and business-critical systems where they had barely existed before. The importance of information security in the business environment has now been underscored, as has the need for skilled, dedicated practitioners of this specialty.

We have traditionally thought of security as consisting of people, sometimes with guns, watching over and guarding tangible assets such as a stack of money or a research lab. Maybe they sat at a desk and watched via closed-circuit cameras installed around the property. These people usually had minimal training and sometimes did not understand much about what they were guarding or why it was important. However, they did their jobs (and continue to do so) according to established processes, such as walking around the facility on a regular basis and looking for suspicious activity or people who do not appear to belong there.

Information security moves that model into the intangible realm. Fundamentally, information security involves making sure that only authorized people (and systems) have access to information. Information security professionals sometimes have different views on the role and definition of information security.

The three primary areas of concern in information security have traditionally been defined as follows:

- **Confidentiality** Ensuring that only authorized parties have access to information. Encryption is a commonly used tool to achieve confidentiality. Authentication and authorization, treated separately in the following discussion, also help with confidentiality.
- **Integrity** Ensuring that information is not modified by unauthorized parties (or even improperly modified by authorized ones!) and that it can be relied on. Checksums and hashes are used to validate data integrity, as are transaction-logging systems.
- **Availability** Ensuring that information is accessible when it is needed. In addition to simple backups of data, availability includes ensuring that systems remain accessible in the event of a Denial-of-Service (DoS) attack. Availability also means that critical data should be protected from erasure—for example, preventing the wipeout of data on your company's external Web site.

Often referred to simply by the acronym *CIA*, these three areas serve well as a security foundation. To fully scope the role of information security, however, we also need to add a few more areas of concern to the list. Some security practitioners include the following within the three areas previously described, but by getting more granular, we can get a better sense of the challenges that must be addressed:

- **Authentication** Ensuring that users are, in fact, who they say they are. Passwords, of course, are the longstanding way to authenticate users, but other methods such as cryptographic tokens and biometrics are also used.
- **Authorization/access control** Ensuring that a user, once authenticated, is only able to access information to which he or she has been granted permission by the owner of the information. This can be accomplished at the operating-system level using file system access controls, or at the network level using access controls on routers or firewalls.
- **Audit capability** Ensuring that activity and transactions on a system or network can be monitored and logged in order to maintain system availability and detect unauthorized use. This process can take various forms: logging by the operating system, logging by a network device such as a router or firewall, or logging by an intrusion detection system (IDS) or packet-capture device.
- **Nonrepudiation** Ensuring that a person initiating a transaction is authenticated sufficiently such that he or she cannot reasonably deny that they were the initiating party. Public key cryptography is often used to support this effort.

You can say that your information is secure when all seven of these areas have been adequately addressed. The definition of *adequately* depends, however, on how much risk exists in each area. Some areas may present greater risk in a particular environment than in others.

Insecurity and the Internet

The federation of networks that became the Internet consisted of a relatively small community of users by the 1980s, primarily in the research and academic communities. Because it was rather difficult to get access to these systems and the user communities were rather closely knit, security was not much of a concern in this environment. The main objective of connecting these various networks together was to share information, not keep it locked away. Technologies such as the UNIX operating system and the TCP/IP networking protocols that were designed for this environment reflected this lack of security concern. Security was simply viewed as unnecessary.

By the early 1990s, however, commercial interest in the Internet grew. These commercial interests had very different perspectives on security, ones often in opposition to those of academia. Commercial information had value, and access to it had to be limited to specifically authorized people. UNIX, TCP/IP, and connections to the Internet became avenues of attack and did not have much capability to implement and enforce confidentiality, integrity, and availability. As the Internet grew in commercial importance, with numerous companies connecting to it and even building entire business models around it, the need for increased security became quite acute. Connected organizations now faced threats that they had never had to consider before.

When the corporate computing environment was a closed and limited-access system, threats mostly came from inside the organizations. These *internal threats* came from disgruntled employees with privileged access who could cause a lot of damage. Attacks from the outside were not much of an issue since there were typically only a few, if any, private connections to trusted entities. Potential attackers were few in number, since the combination of necessary skills and malicious intent were not at all widespread.

With the growth of the Internet, *external threats* grew as well. There are now millions of hosts on the Internet as potential attack targets, which entice the now large numbers of attackers. This group has grown in size and skill over the years as its members share information on how to break into systems for both fun and profit. Geography no longer serves as an obstacle, either. You can be attacked from another continent thousands of miles away just as easily as from your own town.

Threats can be classified as structured or unstructured. *Unstructured threats* are from people with low skill and perseverance. These usually come from people called *script kiddies*—attackers who have little to no programming skill and very little system knowledge. Script kiddies tend to conduct attacks just for bragging rights among their groups, which are often linked only by an Internet Relay Chat (IRC) channel. They obtain attack tools that have been built by others with more skill and use them, often indiscriminately, to attempt to exploit vulnerabilities in their target. If their attack fails, they will likely go elsewhere and keep trying. Additional risk comes from the fact that they often use these tools with little to no knowledge of the target environment, so attacks can wind up causing unintended results. Unstructured threats can cause significant damage or disruption, despite the attacker's lack of sophistication. These attacks are usually detectable with current security tools.

Structured attacks are more worrisome because they are conducted by hackers with significant skill. If the existing tools do not work for them, they are likely to modify them or write their own. They are able to discover new vulnerabilities in systems by executing complex actions that the system designers did not protect against. Structured attackers often use so-called *zero-day exploits*, which are exploits that target vulnerabilities that the system vendor has not yet issued a patch for or does not even know about. Structured attacks often have stronger motivations behind them than simple mischief. These motivations or goals can include theft of source code, theft of credit card numbers for resale or fraud, retribution, or destruction or disruption of a competitor. A structured attack might not be blocked by traditional methods such as firewall rules, or be detected by an IDS. It could even use non-computer methods such as social engineering.

NOTE

Social engineering, also known as *people hacking*, is a means of obtaining security information from people by tricking them. The classic example is calling up a user and pretending to be a system administrator. The hacker asks the user for his or her password to ostensibly perform some important maintenance task. To avoid being hacked via social engineering, educate your user community that they should always confirm the identity of any person calling them and that passwords should never be given to *anyone* over e-mail, instant messaging, or the phone.

Another key task in securing your systems is closing vulnerabilities by turning off unneeded services and bringing them up-to-date on patches. Services that have no defined business need present an additional possible avenue of attack and are just another component that needs patch attention. Keeping patches current is actually one of the most important activities you can perform to protect yourself, yet it is one that many organizations neglect.

The Code Red and Nimda worms of 2001 were successful primarily because so many systems had not been patched for the vulnerabilities they exploited, including multiple Microsoft Internet Information Server (IIS) and Microsoft Outlook vulnerabilities. Patching, especially when you have hundreds or even thousands of systems, can be a monumental task. However, by defining and documenting processes, using tools to assist in configuration management, subscribing to multiple vulnerability alert mailing lists, and prioritizing patches according to criticality, you can get a better handle on the job.

One useful document to assist in this process has been published by the U.S. National Institute of Standards and Technology (NIST), which can be found at http://csrc.nist.gov/publications/nistpubs/800-40/sp800-40.pdf (800-40 is the document number).

Also important is having a complete understanding of your network topology and some of the key information flows within it, as well as in and out of it. This understanding helps

you define different zones of trust and highlights where re-architecting the network in places might improve security—for example, by deploying additional firewalls internally or on your network perimeter.

Identifying Potential Threats

As you prepare your overall security plan and demilitarized zone (DMZ), it is important that you identify and evaluate the potential risks and threats to your network, systems, and data. You must evaluate your risks thoroughly during the identification process to assign some sort of value to the risks in order to determine priorities for protection and likelihood of loss resulting from those risks and threats if they materialize. In this vein, you should be looking at and establishing a risk evaluation for anything that could potentially disrupt, slow, or damage your systems, data, or credibility. In this area, it is important to assign these values to potential threats such as:

- Outside hacker attacks
- Trojans, worms, and virus attacks
- DoS or Distributed Denial-of-Service (DDoS) attacks
- Compromise or loss of internal confidential information
- Network monitoring and data interception
- Internal attacks by employees
- Hardware failures
- Loss of critical systems

This identification process creates the basis for your security plan, policies, and implementation of your security environment. You should realize that this is an ongoing evaluation that is subject to change as conditions within your company and partners (as well as the employee need for access) change and morph over time. We have learned that security is a process and is never truly "finished." However, a good basic evaluation goes a long way toward creating the most secure system we can achieve.

Using VPNs in Today's Enterprise

Ensuring that your data arrives safe and sound when it passes through a network is something everyone wants to have. In an ideal world, your data's integrity and confidentiality would be guaranteed. If you believe this all sounds like nothing more than a fantasy, you are wrong. These types of guarantees can be made when you use IP Security (IPSec as defined in RFC 2401 and later in RFC4301) VPN technologies. When you use an IPSec connection either between two networks or a client and a network, you can ensure that no one looked at the data and no one modified it. Almost every company today uses VPN technologies to secure its data as it passes through various networks. In fact, there are many regulations that specify that a VPN connection must be used to pass specific types of data.

IPSec provides integrity checking to ensure your data was not modified. It also provides encryption, ensuring no one has looked at the data. When two sides create a VPN connection, each side is authenticated to verify that each party is who they say they are. Combined with integrity checking and encryption, you have an almost unbeatable combination.

The Battle for the Secure Enterprise

This book covers the Juniper NetScreen firewall product line and focuses on that specific product and technology. A firewall is the core of securing your network, but there are other products out there that should also be implemented in your network. These additional devices help ensure a network that has security covered from all angles. The following technologies are usually the minimum that companies should implement to provide security in the organization.

A *firewall* can contain many different types of technology to increase its importance in your network. Many firewall products today can integrate several different technologies. Almost all firewalls today provide VPN services. This allows secure streams of data to terminate to your firewall. This is usually over the Internet, but may also be over other unprotected networks. When the traffic gets to your secured network it no longer requires encryption. You can also force users to authenticate before accessing resources through the firewall. This commonly used practice denies access to systems until the user authenticates. When doing this, clients cannot see the resource *until authentication has occurred.*

URL filtering is another requirement in many organizations. URL filtering provides a way to accept or reject access to specific Web sites. This allows companies to reduce liability by users accessing inappropriate Web content. Many firewalls can integrate with this type of scanning when used with another product.

Antivirus software is a requirement for any organization today. With more viruses being written, the last thing you want to have happen in your network is a virus outbreak. The Windows operating system is built to provide so many different functions that there are many ways it can be exploited. In recent months, Microsoft has done a great job of coming out with security patches when or before an exploit is discovered. Typically though, when a vulnerability is discovered, an antivirus software company has a way to stop it much faster than Microsoft. An outbreak on your network can mean disaster, data loss, or loss of your job. Data is a company's most valuable asset today, and loss of that data or access to it can cost companies millions of dollars or more per day. Firewalls can be used to perform virus scanning. These devices are usually deployed in a central area on the network. A tiered antivirus solution is a requirement for any organization.

You should have antivirus scanning on all of your desktops and servers to stop infections at the source. This will help prevent most virus outbreaks. Also, you should have antivirus scanning on your Simple Mail Transfer Protocol (SMTP) mail forwarder, and it should be resident directly on your mail server. Your chances for a virus outbreak should be small as long as you keep all of those devices up-to-date with the appropriate virus definitions. New technologies such as inline virus scanning in firewalls and other network appliances can also provide extra protection from viruses.

Patch management has become a truly Herculean effort with all of the software an organization needs to run today. Patching operating systems and applications as soon as a vulnerability occurs is a must. With limited staff and increased software deployed, this task is almost impossible to accomplish. However, by employing an antivirus system, you can provide a first level of defense against the spreading of malicious software or malware.

No matter what device or security you provide, everything usually comes down to some type of access token, usually a username and password. Unfortunately, using static usernames and passwords is not enough anymore. Even 15 to 30 days may be too long to keep the same password. Two-factor authentication, digital certificates, and personal entropy are leading the march to provide a stronger nonstatic type of authentication that is hard to break.

Your network has millions if not billions of packets traversing it every day. Do you know what they are all doing? This is where an intrusion detection or intrusion detection and prevention device comes into play. These devices detect application- and network-based attacks. Intrusion detection devices sit on your network and watch traffic. They provide alerts for unusual traffic as well as TCP resets to close TCP sessions. The newer technology of intrusion detection and prevention provide the capability to stop malicious traffic altogether, as well as alert users about it. However, heavy tuning of the products is required to make it effective.

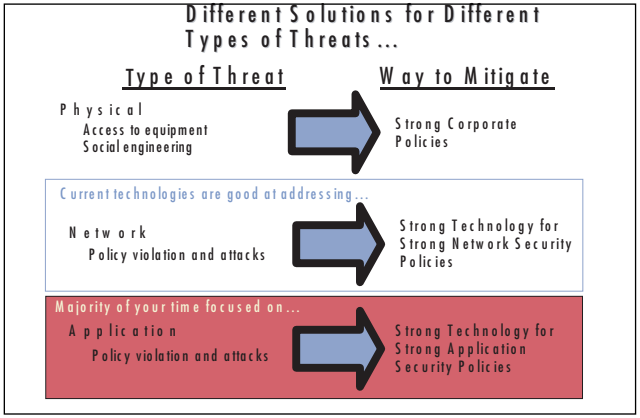
Access into your network should be encrypted whenever possible. This ensures that parties that are not authorized to see your data do not get access to it by any means. IPSec VPN clients are one of the most popular ways to do this. This type of client provides strong encryption of your data as well as access to your internal resources without having them be publicly accessible. A new trend in VPN solutions is the Secure Sockets Layer (SSL) VPN. These products allow you to put more behind them and do not require pre-deployment of a VPN client.

Making Your Security Come Together

In today's security battlefield, it almost seems impossible to win. You must identify the best products and procedures for your organization. If you have all of the suggested security solutions, but not enough staff to manage it, then the solutions may not be effective enough. Simply having the appropriate products is not going to resolve all of your problems; you must effectively understand how to use and configure the products. There is no easy solution regarding the best way to go about securing your organization. This is why companies all over the world spend hundreds of millions of dollars on consulting companies to come in and make security decisions for them.

Three common types of threats exist: physical, network, and application. Today, physical threats, those that come from someone gaining physical access to equipment and data, can be mitigated by several implementations of corporate policy and access control. Network threats, those that come from communications across one's data network infrastructure, are best mitigated by today's available technology that checks data network transmission policies and blocks violations. Finally, application threats—those that come from someone wanting to gain access to the application and data—are the focus of the leading-edge security technology (see Figure 1.8).

Figure 1.8 Mitigating Three Common Types of Threats



Understanding Firewall Basics

A firewall is a device that is part hardware and part software that is used to secure network access. Throughout this book, we will cover every aspect of the Juniper NetScreen firewall product line, its usage, and configuration. Before we begin to look at the various aspects of the Juniper NetScreen firewall, we need to look at some general firewall information. This will give you a better perspective on the pros and cons of the Juniper NetScreen firewall. Firewalls have come a long way since the original inception of the idea.

In the first part of this section, we discuss the firewall in today's network. We look at the types of firewalls and how their importance has increased, as well as their increased deployments in each network. Next, the many types of firewalls are discussed and contrasted and compared. Finally, we will review some common firewall concepts that will be used throughout the book.

Types of Firewalls

In the past, an organization may have had one firewall that protected the edge of the network. Some companies did not have their network attached to the Internet, or may have had perhaps one or two stations that would dial up to the Internet or to another computer they needed to exchange data with. After the late 1990s, however, the need for the Internet, its information, and e-mail was undeniable.

With the requirement for instantaneous e-mail access comes the requirement for an always-on Internet connection. At first, companies would place their systems directly on the Internet with a public IP address. This, of course, is not a scalable solution for the long term. With limited IP addresses and unlimited threats, a better solution is required. At first, the border router that connected the Internet medium to the local network was used to provide a simple layer of access control between the two networks. With the need for better security, new types of firewalls were developed to meet the new needs for an Internet-enabled office.

Better security, the capability of the firewall to provide more secured segments, and the need to thwart newer styles of attacks brought firewalls to where they are today.

Packet Filters

The most basic firewall technology is the packet filter. A packet filter is designed to filter packets based on source IP, destination IP, source port, and destination port, and do so on a packet-per-packet basis to determine if that packet should be allowed through.

The basic security principles of a packet filter, such as allowing or denying packets based upon IP address, provide the minimum amount of required security. So then, where does the packet filter go wrong? A packet filter cannot determine if the packet is associated with any other packets that make up a session. A packet filter does a decent enough job of protecting networks that require basic security, as depicted in Figure 1.9. The packet filter does not look to the characteristics of a packet, such as the type of application it is or the flags set in the TCP portion of the packet. Most of the time this will work for you in a basic security setting. However, there are ways to get around a packet filter. Because the packet filter does not maintain the state of exactly what is happening, it cannot determine the proper return packets that should be allowed through the connection.

For example, if you wanted to permit outbound access to DNS on UDP port 53, you would need to allow access for the return packet as well. A packet filter checks only for a packet match as it traverses the router, and the router cannot determine what the proper return packet will be in order to let it also through. So now you have to allow access inbound for that DNS entry to return. So its source port would be UDP 53 and the inbound destination port would be the source port, which could be 1024-65535. Now add that up with all of the other applications you need to allow through the firewall and you can see the problem. Because the packet filter has no way to dynamically create an access rule to allow inbound traffic, the packet filter is not effective as a security gateway.

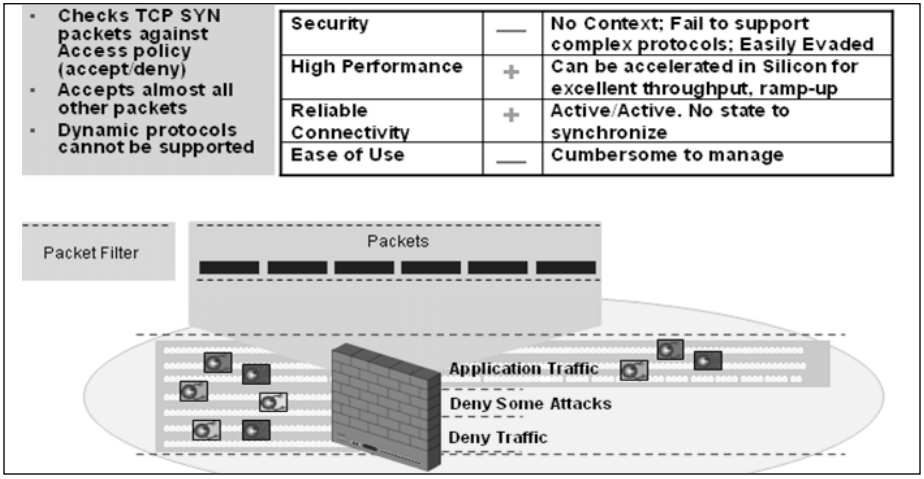
Packet filters in modern routers can be extremely important. Cisco and Juniper routers, for example, use access lists (in the case of Cisco) and firewall filters (in the case of Juniper, JUNOS) to do more than just provide packet filtering. These packet filters are also used in CoS/QoS, policy-based routing, rate-limiting and policing, as well as controlling what can reach the router.

Packet filters in modern routers are also very high performance. Many ASICs-based routers, like the Juniper M and T series routers, are able to filter traffic without any discernible performance degradation. This is extremely important when the routers are used at the Internet edge, either at the carrier side or the customer side of the Internet link. Routers here may have to perform some "course screening" of inbound Internet traffic. This is commonly done to offload the impact of these inbound sessions on the perimeter firewalls next in line.

An example of such a case would be a SYN flood from the Internet inbound to a Web server protected behind a perimeter firewall in the DMZ. With today's Internet speeds available to companies/organizations being DS-3 (45 Mbps) or greater, with many opting for 100Mbps Fast Ethernet Internet "drains," it is possible to have a SYN flood, or other denial-

of-service (DOS) attacks from the Internet hit your perimeter firewall and overload its ability to "fend off" the attack, compromising the other sessions/flows traversing the firewall. In this case, the upstream, Internet-facing router may be utilized with a packet filter to rate-limit the inbound SYN traffic, or the precise characteristics of the DOS attack, to a level in which the perimeter firewall is able to withstand.

Figure 1.9 Packet Filter (ACL) Advantages/Disadvantages



Access Control Lists, or ACLs, in Cisco routers are packet filters. Juniper routers, running JUNOS, call packet filters firewall filters. These are high-performance packet filters that look at L3 and L4 information for filtering. It is important to note that packet filters in either Cisco's or Juniper's case, are able to look at information only in the packet header, and not the payload.

Application Proxy

Application proxies provide one of the most secure types of access you can have in a security gateway. An application proxy sits between the protected network and the network you want to be protected from. Every time an application makes a request, the application intercepts the request to the destination system. The application proxy initiates its own request, as opposed to actually passing the client's initial request. When the destination server responds back to the application proxy, the proxy responds back to the client as if it was the destination server. This way the client and the destination server never actually interact directly. This is the most secure type of firewall because the entire packet, including its application portion, can be completely inspected.

However, this is not dominant technology today for several reasons. The first downfall of the application proxy, as depicted in Figure 1.10, is performance. Because the application proxy essentially has to initiate its own second connection to the destination system, it takes

twice the amount of connections to complete its interaction. On a small scale, the slowdown will not be a persistent problem, but when you get into a high-end requirement with many concurrent connections this is not a scalable technology. Even today, with extremely high performance, general-purpose CPUs and efficient operating systems, application proxies still tend to have significant variation in performance in real-world environments. Furthermore, when the application proxy needs to interact with all of today's different applications, it needs to have some sort of engine to interact with the applications it is connecting to. For most highly used vanilla applications such as Web browsing or HTTP, this is not a problem. However, if you are using a proprietary protocol, an application proxy might not be the best solution for you.

Figure 1.10 Application Proxy Advantages/Disadvantages

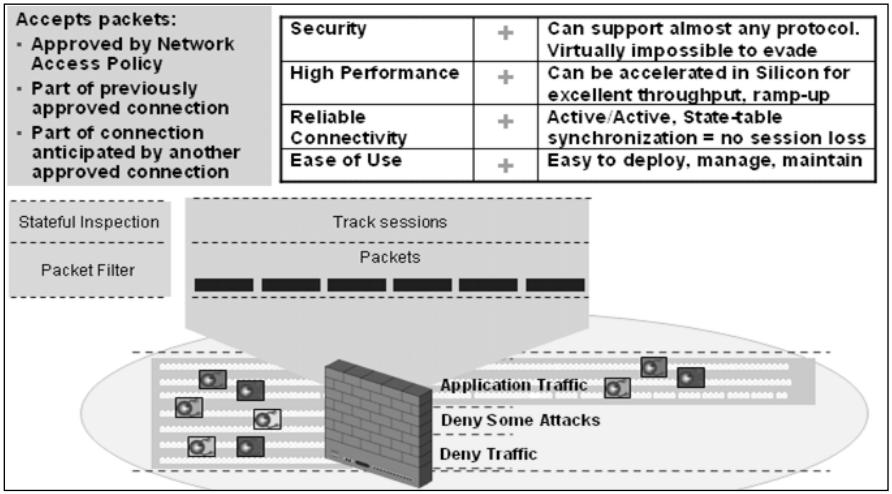
Security	+	Caveat: Limited Protocol Support
High Performance	—	Breaking client/server connection, processing intensive
Reliable Connectivity	—	No High Availability
Ease of Use	+	Ease of use

Stateful Inspection

Stateful inspection is today's choice for the core inspection technology in firewalls. Stateful inspection functions like a packet filter by allowing or denying connections based upon the same types of filtering. However, a stateful firewall also monitors the "state" of a communication. So, for example, when you connect to a Web server and that Web server must respond to you, the stateful firewall has the proper access open and ready for the responding connection. When the connection ends, that opening is closed. Among the big three names in firewalls today, all of them use this reflexive technology. There are, as mentioned earlier, protocols such as UDP and ICMP that do not have any sort of state to them. The major vendors recognize this and have to make their own decisions about what exactly constitutes a UDP or ICMP connection. Overall though, most uses of stateful technology across vendors have been in use for some time and have worked the bugs out of those applications.

Many companies that implement stateful inspection use a more hybrid method between application proxy and stateful inspection when inspecting specific protocols (see Figure 1.11). For example, if you were to do URL filtering on most firewalls, you might need to actually employ application proxy-type techniques to provide the proper inspection. This, much like application proxy firewalls, does not scale and is not a good idea for a large amount of users. However, depending on the vendor and function, your mileage may vary.

Figure 1.11 Stateful Packet Inspection Advantages



Firewall Incarnate

A firewall can function in many different ways, but it always has the same basic requirements. A firewall is part hardware and part software, and the combination of each makes a huge difference. In this section, we will look at the differences between an appliance-based firewall and a standard operating system (OS) running a firewall as an application.

First, we will look at the firewall application that sits on an OS. In this case, there is an underlying operating system that runs on a standard computer system. The computer system consists of a processor, memory, and hard disk drive. The operating system will most likely be used for other functions without the firewall application. The operating system may be a multifunction operating system such as Microsoft Windows, Red Hat Linux, or Sun Solaris. To provide the utmost security, the operating system will have to be stripped down either by the end user or the manufacturer before it is suitable for use as a secure gateway. The firewall software is then installed on top of the operating system, permitting the OS to provide additional services or resources to other systems or users.

The other scenario is an operating system that has the firewall application integrated with it. In this case, the operating system is not used for any purpose other than to provide the firewall application. The device has a processor, memory, and flash memory for long-term storage. This device is an appliance.

In the first scenario, the device has some clear advantages. It does not require a single-purpose device to be used and the underlying hardware most likely can be employed for another purpose besides the firewall. This type of firewall can use third-party applications on the system and ultimately the firewall application may be able to have more advanced features because it has so much system behind it. The firewall application is also limited based upon the limits of the specified hardware it is running upon, as well as the underlying OS.

For example, if you wanted to add additional interfaces, it is limited to the specific type of hardware you are running. In most cases, you can upgrade your hardware and then simply reinstall your application to upgrade the system.

In the second scenario, we have an appliance whose sole purpose is to provide a firewall that will pass packets in and out as fast as possible while inspecting them based upon the defined security policy. The device's hardware is specialized for providing that single application. However, one disadvantage of using this type of firewall is that you cannot load other third-party applications on that system. Furthermore, the device may have some specific limitations, such as limited memory or physical interfaces, and the only way to upgrade the device is to do a forklift upgrade and replace the entire device.

Firewall Ideologies

No matter which type of firewall you choose, there are some basic design considerations involved. Placement is usually the biggest question. Where is the most effective location to place my firewall to maximize its effectiveness? Is one firewall enough? How do I protect the servers that I need to make publicly accessible? These and many other questions come to mind when discussing firewall effectiveness. Unfortunately, the answers to all of these questions are beyond the scope of this section.

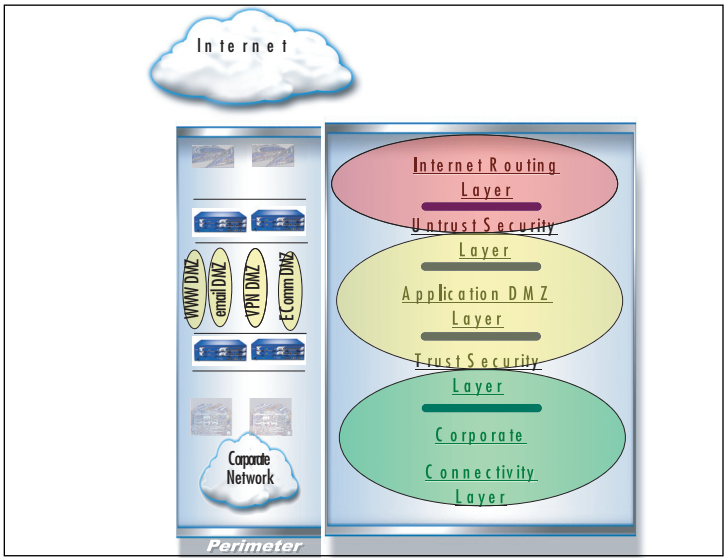
DMZ Concepts

The use of a DMZ and its overall design and implementation can be relatively simple or extremely complex, depending on the needs of the particular business or network system. The DMZ concept came into use as the need for the separation of networks became more acute when we began to provide more access to services for individuals or partners outside the LAN infrastructure. One of the primary reasons that the DMZ has come into favor is the realization that a single type of protection is subject to failure. This failure can arise from configuration errors, planning errors, equipment failure, or deliberate action on the part of an internal employee or external attack force. The DMZ has proven to be more secure and to offer multiple layers of protection for the security of the protected networks and machines. It has also proven to be very flexible, scalable, and relatively robust in its ability to provide the protection we need. Figure 1.12 shows how "sandwiching" of DMZs between Internet-facing firewalls and intranet/corporate-facing firewalls is used to protect segments of a corporate network such as e-commerce, e-mail, Web access, and virtual private networks (VPNs). DMZ design now includes the ability to use multiple products (both hardware- and software-based) on multiple platforms to achieve the level of protection necessary, and DMZs are often created to provide failover capabilities as well.

When we are working with a DMZ, we must have a common ground to work from. To facilitate understanding, we examine a number of conceptual paths for traffic flow in the following section. Before we look at the conceptual paths, let's make sure we understand the basic configurations that can be used for firewall and DMZ location, and how each of them can be visualized. In the following figures, we'll see and discuss these configurations. Please

note that each of these configurations is useful on internal networks needing protection, and in addition helps protect your resources from networks such as the Internet. Our first configuration is shown in Figure 1.13.

Figure 1.12 Common Perimeter Firewall Deployment



Designing & Planning…

Know What You Want to Secure First

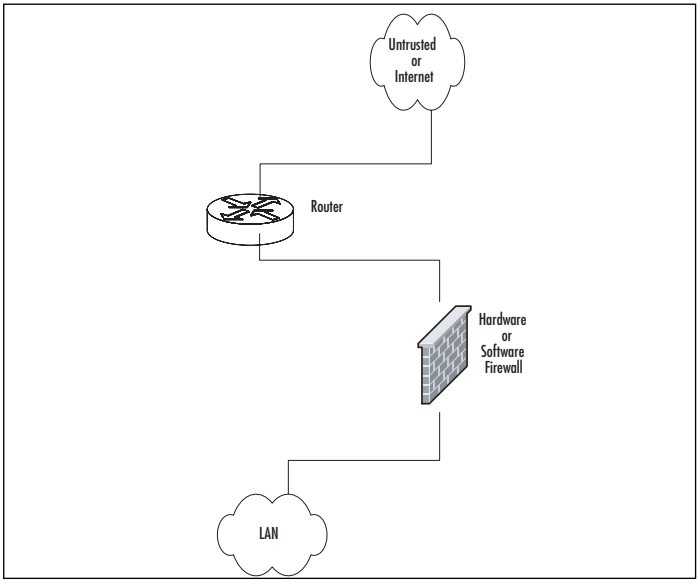
As you begin your DMZ design process, you must first be clear about what your design is intended for. A design that is only intended to superficially limit internal users' access to the Internet, for instance, requires much less planning and design work than a system protecting resources from multiple access points or providing multiple services to the public network or users from remote locations. An appropriate path to follow for your predesign path might look like the following:

- Perform baseline security analysis of existing infrastructure, including OS and application analysis
- Perform baseline network mapping and performance monitoring
- Identify the risk to resources and outline appropriate mitigation processes

Continued

- Identify potential security threats, both external and internal
- Identify needed access points from external sources
- Public networks
- VPN access
- Extranets
- Remote access services
- Identify critical services
- Plan your DMZ

Figure 1.13 A Basic Network with a Single Firewall



In Figure 1.13, we can see the basic configuration that would be used in a simple network situation in which there was no need to provide external services. This configuration would typically be used to begin to protect a small business or home network. It could also be used within an internal network to protect an inner network that had to be divided and isolated from the main network. This situation could include payroll, finance, or development divisions that need to protect their information and keep it away from general network use and view.

Figure 1.14 details a protection design that would allow for the implementation and provision of services outside the protected network. In this design, it would be absolutely imperative that rules be enacted to not allow the untrusted host to access the internal network. Security of the bastion host machine would be accomplished on the machine itself,

and only minimal and absolutely necessary services would be enabled or installed on that machine. In this design, we might be providing a Web presence that did not involve e-commerce or the necessity to dynamically update content. This design would not be used for provision of virtual private network (VPN connections, FTP services, or other services that required other content updates to be performed regularly.

Figure 1.14 Basic Network, Single Firewall, and Bastion Host (Untrusted Host)

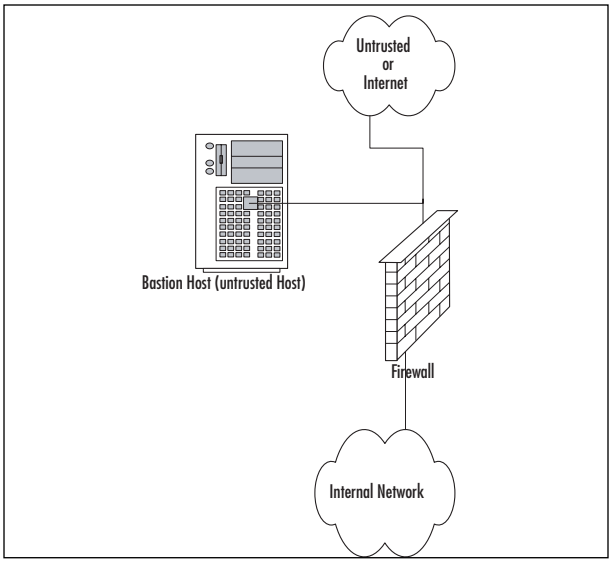


Figure 1.15 shows a basic DMZ structure. In this design, the bastion host is partially protected by the firewall. Rather than the full exposure that would result to the bastion host in Figure 1.14, this setup would permit us to specify that the bastion host in Figure 1.14 could be allowed a full outbound connection, but the firewall could be configured to allow only port 80 traffic inbound to the bastion host (assuming it was a Web server) or others as necessary for connection from outside. This design would allow connection from the internal network to the bastion host if it was necessary. This design would potentially allow the updating of Web server content from the internal network if permitted by firewall rule, which could let traffic flow to and from the bastion host on specific ports as designated.

Figure 1.16 shows a generic dual-firewall DMZ configuration. In this arrangement, the bastion host can be protected from the outside and allowed to connect to or from the internal network. In this arrangement, like the conditions in Figure 1.15, flow can be controlled to and from both of the networks away from the DMZ. This configuration and method is more likely to be used if more than one bastion host is needed for the operations or services being provided.

Figure 1.15 A Basic Firewall with a DMZ

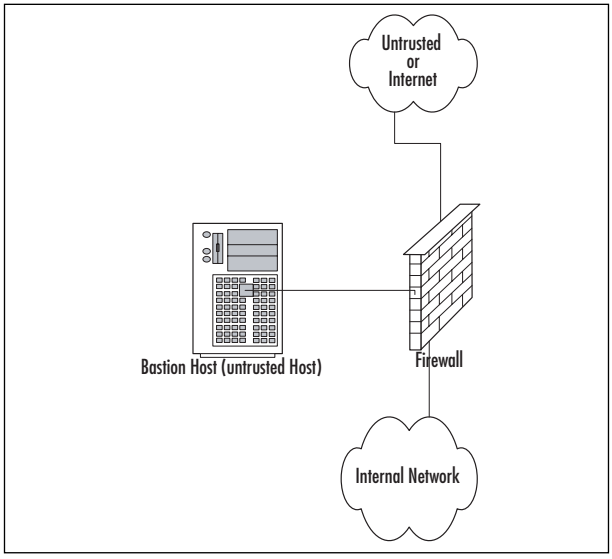
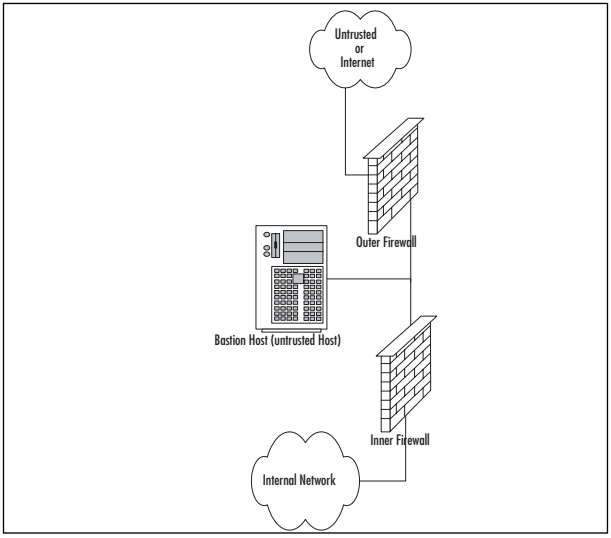


Figure 1.16 A Dual Firewall with a DMZ



Traffic Flow Concepts

Now that we've had a quick tour of some generic designs, let's take a look at the way network communications traffic typically flows through them. Be sure to note the differences between the levels and the flow of traffic and protections offered in each of them.

Figure 1.17 illustrates the flow pattern for information through a basic single-firewall setup. This type of traffic control can be achieved through hardware or software and is the basis for familiar products such as Internet Connection Sharing (ICS) and the NAT functionality provided by digital subscriber lines (DSLs) and cable modems used for connection to the Internet. Note that flow is unrestricted outbound, but the basic configuration will drop all inbound connections that did not originate from the internal network.

Figure 1.17 Basic Single-Firewall Flow

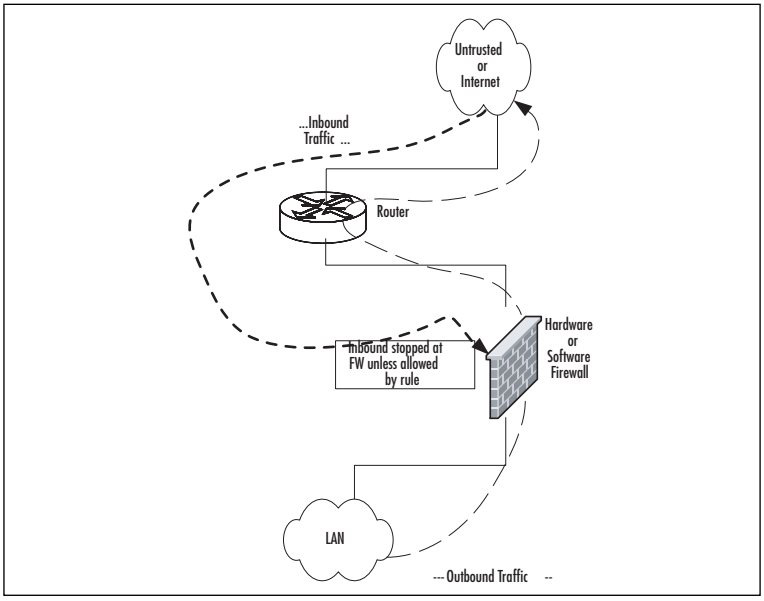


Figure 1.18 reviews the traffic flow in a network containing a bastion host and a single firewall. This network configuration does not produce a DMZ; the protection of the bastion host is configured individually on the host and requires extreme care in setup. Inbound traffic from the untrusted network or the bastion host is dropped at the firewall, thus providing protection to the internal network. Outbound traffic from the internal network is allowed.

Figure 1.19 shows the patterns of traffic as we implement a DMZ design. In this form, inbound traffic flows through to the bastion host if allowed through the firewall and is dropped if destined for the internal network. Two-way traffic is permitted as specified between the internal network and the bastion host, and outbound traffic from the internal network flows through the firewall and out, generally without restriction.

Figure 1.18 A Basic Firewall with Bastion Host Flow

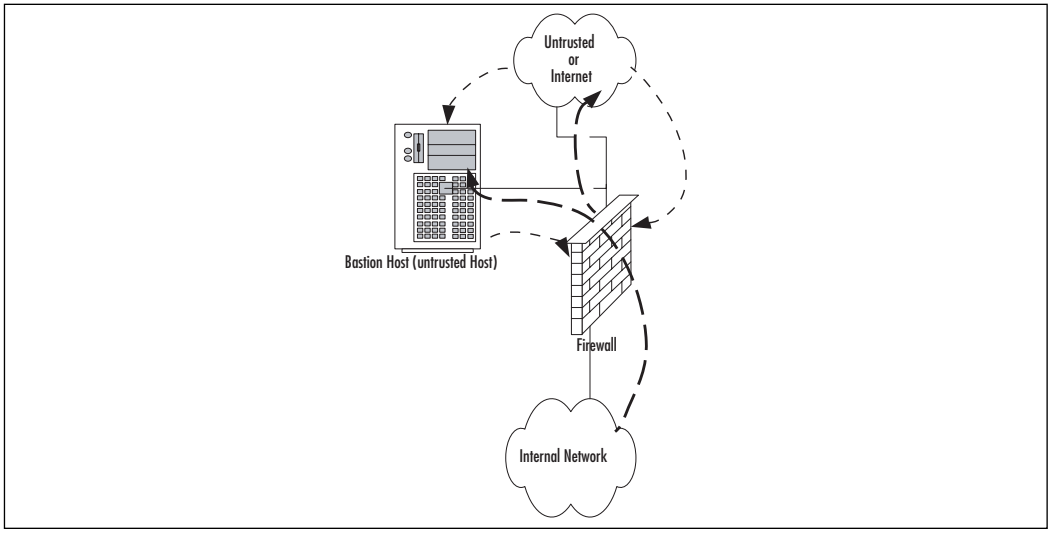


Figure 1.19 A Basic Single Firewall with DMZ Flow

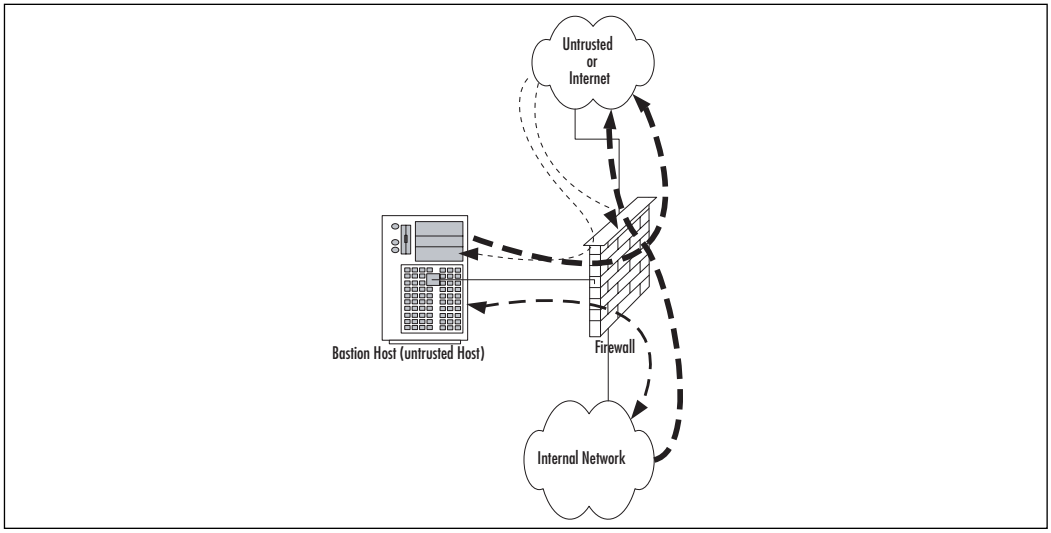
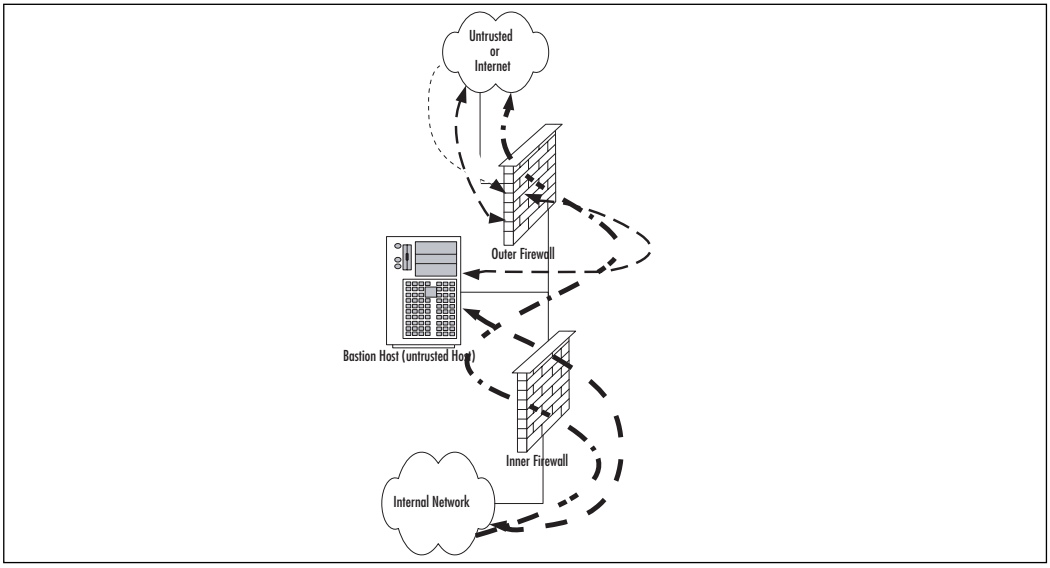


Figure 1.20 contains a more complex path of flow for information but provides the most capability in these basic designs to allow for configuration and provision of services to the outside. In this case, we have truly established a DMZ that is separated and protected from both the internal and external networks. This type of configuration is used quite often when there is a need to provide more than one type of service to the public or outside world, such as e-mail, Web servers, DNS, and so forth. Traffic to the bastion host can be

allowed or denied as necessary from both the external and internal networks, and incoming traffic to the internal network can be dropped at the external firewall. Outbound traffic from the internal network can be allowed or restricted either to the bastion host (DMZ network) or the external network.

Figure 1.20 A Dual Firewall with DMZ Flow



As you can see, there is a great amount of flexibility in the design and function of your protection mechanisms. In the sections that follow, we expand further on conditions for the use of different configurations and on the planning that is done to implement them.

Networks with and without DMZs

As we pursue our discussions about the creation of DMZ structures, it is appropriate to also take a look at the reasoning behind the various structures of the DMZ and when and where we'd want to implement a DMZ or perhaps use some other alternative.

During our preview of the concepts of DMZs, we saw in Figures 1.15 to 1.20 some examples of potential design for network protection and access. Your design may incorporate any or all of these types of configurations, depending on your organization's needs. For instance, Figure 1. 17 shows a configuration that may occur in the case of a home network installation or perhaps with a small business environment that is isolated from the Internet and does not share information or need to provide services or information to outside customers or partners. This design would be suitable under these conditions, provided configuration is correct and monitored for change.

Figure 1.14 illustrates a network design with a bastion host located outside the firewall. In this design, the bastion host must be stripped of all unnecessary functionality and services and protected locally with appropriate file permissions and access control mechanisms. This design would be used when an organization needs to provide minimal services to an external network, such as a Web server. Access to the internal network from the bastion host is generally not allowed, because this host is absolutely subject to compromise.

Figure 1.19 details the first of the actual DMZ designs and incorporates a screened subnet. In this type of design, the firewall controls the flow of information from network to network and provides more protection to the bastion host from external flows. This design might be used when it is necessary to be able to regularly update the content of a Web server, or provide a front end for mail services or other services that need contact from both the internal and external networks. Although better for security purposes than the design shown in Figure 1.14, this design still produces an untrusted relationship in the bastion host in relation to the internal network.

Finally, Figure 1.20 provides a design that allows for the placement of many types of service in the DMZ. Traffic can be very finely controlled through access at the two firewalls, and services can be provided at multiple levels to both internal and external networks. Some vendors call this design a "firewall sandwich"; it is also commonly called "defense in depth."

In modern financial services perimeter environments, several layers of these DMZs may occur. It is not uncommon to see 3, 5, or 7 layers within a DMZ environment.

Intra-DMZ traffic is also a concern at the Internet perimeter. There might be a need to have one application server sitting in the DMZ to communicate with another application server sitting in another DMZ. The intra-DMZ security policy, performance characteristics, and L7 protection differ greatly from the traffic that is inbound from the Internet.

Finally, the Internet-facing firewall(s) have typically different security policies, performance requirements, and SLA requirements than the intranet/corporate-facing firewall(s). Therefore, it is logical to segregate these different characteristics into different platforms.

In the next section, we profile some of the advantages and disadvantages of the common approaches to DMZ architecture and provide a checklist of sorts to help you make a decision about the appropriate use (or not) of the DMZ for protection.

The Pros and Cons of Basic DMZ Designs

Table 1.6 details the advantages and disadvantages of the various types of basic design discussed in the preceding section.

Table 1.6 The Pros and Cons of Basic DMZ Designs

Basic Design	Advantages	Disadvantages	Appropriate Utilization
Single firewall	Inexpensive, fairly easy configuration, low maintenance	Much lower security capabilities, no growth or expansion potential	Home, small office/home office (SOHO), small business without need to provide services to others
Single firewall with bastion host	Lower cost than more robust alternatives	Bastion host extremely vulnerable to compromise, inconvenient to update content, loss of functionality other than for absolutely required services; not scalable	Small business without resources for more robust implementation or static content being provided that doesn't require frequent updates
Single firewall with screened subnet and bastion host	Firewall provides protection to both internal network and bastion host, limiting some of the potential breach possibilities of an unprotected bastion host	Single point of failure; some products limit network addressing to DMZ in this configuration to public addresses, which might not be economic or possible in your network	Networks requiring access to the bastion host for updating of information
Dual firewall with DMZ	Allows for establishment of multiple service-providing hosts in the DMZ; protects bastion hosts in DMZ from both networks, allows	Requires more hardware and software for implementation of this design; more configuration work and monitoring required much more granular control of resources and access; removes single point of failure and attack	Larger operations that require the capability to offer multiple types of Web access and services to both the internal and external networks involved

Tools & Traps…

Bastion Hosts

Bastion hosts must be individually secured and hardened because they are always in a position that could be attacked or probed. This means that before placement, a bastion host must be stripped of unnecessary services; fully updated with the latest service packs, hot fixes, and updates; and isolated from other trusted machines and networks to eliminate the possibility that its compromise would allow for connection to (and potential compromise of) the protected networks and resources. This also means that a machine being used for this purpose should have no user accounts relative to the protected network or directory services structure, which could lead to enumeration of your internal network.

DMZ Design Fundamentals

DMZ design, like security design, is always a work in progress. As in security planning and analysis, we find DMZ design carries great flexibility and change potential to keep the protection levels we put in place in an effective state. The ongoing work is required so that the system's security is always as high as we can make it within the constraints of time and budget while still allowing appropriate users and visitors to access the information and services we provide for their use. You will find that the time and funds spent in the design process and preparation for the implementation are very good investments if the process is focused and effective; this will lead to a high level of success and a good level of protection for the network you are protecting. In this section of the chapter, we explore the fundamentals of the design process. We also incorporate the information we discussed in relation to security and traffic flow to make decisions about how our initial design should look. Additionally, we'll build on that information and review some other areas of concern that could affect the way we design our DMZ structure.

NOTE

In this section, we look at the design of a DMZ from a logical point of view. Physical design and configuration are covered in following chapters based on the vendor-based solution you are interested in deploying.

Why Design Is so Important

Design of the DMZ is critically important to the overall security of your internal network—and the success of your firewall and DMZ deployment. The DMZ design can incorporate sections that isolate incoming VPN traffic, Web traffic, partner connections, employee connections, and public access to information provided by your organization. Design of the DMZ structure throughout the organization can protect internal resources from internal attack. As we discussed in the security section, it has been well documented that much of the risk of data loss, corruption, and breach actually exists *inside* the network perimeter. Our tendency is to protect assets from external harm but to disregard the dangers that come from our own internal equipment, policies, and employees.

These attacks or disruptions do not arise solely from disgruntled employees either. Many of the most damaging conditions that occur are because of inadvertent mistakes made by well-intentioned employees. Each and all of these entry points is a potential source of loss for your organization, and ultimately can provide an attack point to defeat your other defenses. Additionally, the design of your DMZ will allow you to implement a multilayered approach to securing your resources that does not leave a single point of failure in your plan. This minimizes the problems and loss of protection that can occur because of misconfiguration of rule sets or Access Control Lists (ACLs), as well as reduces the problems that can arise due to hardware configuration errors. In the last chapters of this book, we look at how to mitigate risk through testing of your network infrastructure to make sure your firewalls, routers, switches, and hosts are thoroughly hardened so that when you do deploy your DMZ segment, you can see for yourself that it is in fact secure from both internal as well as external threats.

Designing End-to-End Security for Data Transmission between Hosts on the Network

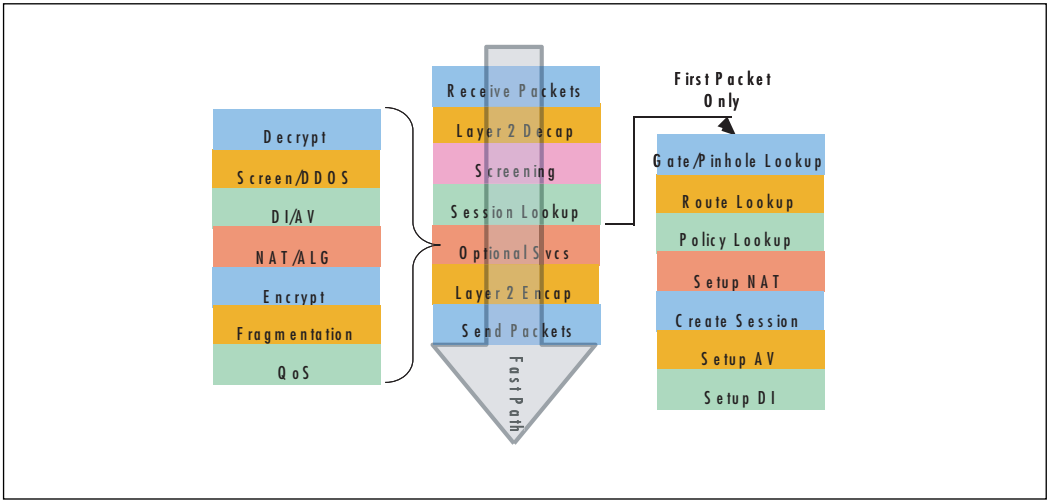
Proper DMZ design, in conjunction with the security policy and plan developed previously, allows for end-to-end protection of the information being transmitted on the network. The importance of this capability is explored more fully later in the chapter, when we review some of the security problems inherent in the current implementation of TCP/IPv4 and the transmission of data. The use of one or more of the many firewall products or appliances currently available will most often afford the opportunity not only to block or filter specific protocols but also to protect the data as it is being transmitted. This protection may take the form of encryption and can utilize the available transports to protect data as well. Additionally, proper utilization of the technologies available within this design can provide for the necessary functions previously detailed in the concepts of AAA and CIA, utilizing the multilayer approach to protection that we have discussed in earlier sections. This need to provide end-to-end security requires that we are conversant with and remember basic network traffic patterns and protocols. The next few sections help remind us about these, and further illustrate the need to design the DMZ with this capability in mind.

Traffic Flow and Protocol Fundamentals

Another of the benefits of using a DMZ design that includes one or more firewalls is the opportunity to control traffic flow into and out of the DMZ much more cohesively and with much more granularity and flexibility. When the firewall product in use (either hardware or software) is a product designed above the home-use level, the capability usually exists to not only control traffic that is flowing in and out of the network or DMZ through packet filtering based on port numbers but often to allow or deny the use of entire protocols. For instance, the rule set might include a statement that blocks communication via ICMP, which would block protocol 1. A statement that allowed IPSec traffic where it was desired to allow traffic utilizing ESP or AH would be written permitting protocol 50 for ESP or 51 for Authentication Header (AH). (For a listing of the protocol IDs, visit www.iana.org/assignments/protocol-numbers.) Remember that like the rule of security that follows the principle of least privilege, we must include in our design the capability to allow only the absolutely necessary traffic into and out of the various portions of the DMZ structure.

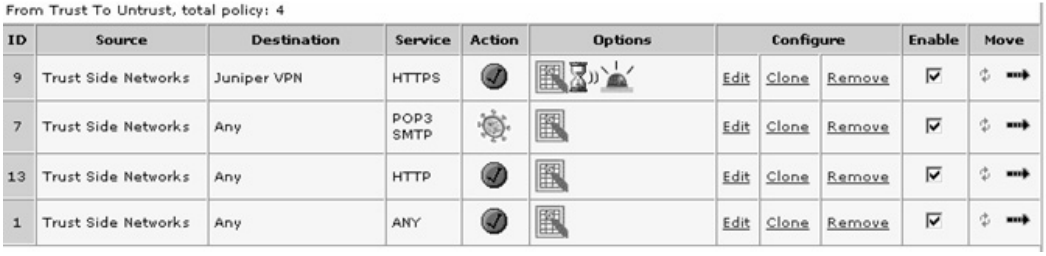
To understand Juniper NetScreens and how they work as high-performance firewalls, one must be able to understand how an IP packet is processed through the firewall. This is important in terms of understanding the performance of the appliance and where to look when troubleshooting problems. A Juniper NetScreen firewall works similar to early layer 3 switches and hardware-accelerated routers in that the device has slow-path and fast-path avenues thru the appliance. The first packet of a new session to enter the firewall, or a non-IPSec packet destined for the firewall, is handled by the CPU of the firewall. All subsequent packets of the existing session, or all IPSec packets, are handled by ASICs in all of the NS-5x00 and ISG series firewalls. For all other Juniper NetScreen firewalls that do not include an ASIC, as in the case of the NS-5GT and SSG series firewalls, the flows are processed efficiently via the CPU (see Figure 1.21).

Figure 1.21 The Path of a Packet through a Juniper NetScreen Firewall



Juniper NetScreen firewalls set up the flows with the routing lookup, firewall rules, and access control being performed once on the first packet of the flow only. After that, fast lookup occurs for the remainder of the packets in the flow. This allows the Juniper NetScreen to have no performance penalty for having a large rule set (10,000 entries takes just as little time as eight entries). This also allows fast handling of NAT and route table lookups. All session matching occurs on a five tuple–based policy: source and destination zone, source and destination address, and finally service. An example of the five policy entries is shown in Figure 1.22.

Figure 1.22 Policy Entries


From Trust To Untrust, total policy: 4

ID	Source	Destination	Service	Action	Options	Configure			Enable	Move
9	Trust Side Networks	Juniper VPN	HTTPS			Edit	Clone	Remove	☑	
7	Trust Side Networks	Any	POP3 SMTP			Edit	Clone	Remove	☑	
13	Trust Side Networks	Any	HTTP			Edit	Clone	Remove	☑	
1	Trust Side Networks	Any	ANY			Edit	Clone	Remove	☑	

Zone trust, source network(s), zone untrust, destination network(s), and finally the service are used to match the five-tuple policy.

Summary

In this chapter, we reviewed the many different fundamental concepts that are important to networking. First, we looked at the OSI model. As you can see, it's very important to understand the OSI model. It's used throughout this book and other documentation from Juniper Networks. In fact, almost any documentation referencing networking uses the OSI model as a base to explain how networking functions. An important fact to remember is that the OSI model is just that; a model. As you saw in the explanation of the TCP/IP model, the OSI model does not exactly fit together with TCP/IP. However, once you understand TCP/IP and how it works, you will understand it for all platforms and applications.

If you were to truly have a book titled *Understanding Security Basics*, it could easily span over a thousand pages. In this chapter, we have brought together a concise version of that material to help you begin to understand the expansive world that is security. In the battle for the secure enterprise, the most important thing to remember is that there is no single solution to secure everything. Many products claim to have the silver bullet for securing your network, but this is nothing more than marketing. Each company has different restrictions on resources and has different security requirements.

In the last section, we reviewed the basics of firewalls. The evolution of firewalls has been a long and harrowed path. As new threats come to light, new technologies will always be created

to stop these threats. The concept of a DMZ is an important one to understand. Segmenting your important hosts is one of the critical things you can do to secure your network.

Solutions Fast Track

Understanding Networking

- ☑ The OSI model is used as a reference for all networking protocols.
- ☑ TCP/IP is used as the core networking protocol today on both the Internet and in the enterprise.
- ☑ The TCP protocol has clearly defined points where a session begins and ends.

Understanding Security Basics

- ☑ Security is a process that is never finished; security needs are constantly changing as well as the needs for new technologies.
- ☑ There is no single product or solution that can be used to ensure your network's security.
- ☑ Each organization has its own specific needs to best help it minimize security risks.

Understanding Firewall Basics

- ☑ Juniper NetScreen firewalls use stateful inspection to ensure the security of connections passing though the firewall.
- ☑ Firewall technology is constantly changing to meet the security needs of today's organizations.
- ☑ DMZ design depends on the designer's ability to accurately assess the actual risks in order to design an adequate structure.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: If the OSI model does not match the way in which TCP/IP functions, then why is it still used?

A: The OSI model is just that; a model. It was to be originally used as the model for the development of networking protocols. However, developers found the specifications too rigid for practical use. Most networking protocol suites do not follow the OSI model fully, but do follow the layered concept that was identified first during the development of the OSI model.

Q: Why would I want to use a Juniper NetScreen firewall appliance when I could just use Red Hat Linux with ipchains?

A: A Juniper NetScreen firewall appliance is built with one thing in mind: security. It doesn't have to provide any other services. Many more services and applications run on conventional operating systems that can contain security vulnerabilities. Furthermore, Juniper NetScreens do not have hard drives. This is the most likely part to fail on a computer when running for an extended period of time. Finally, the Juniper NetScreen firewall architecture runs on Application-Specific Integrated Circuit (ASIC) chips. These are specifically designed to perform special tasks, providing more performance with less horsepower, while general-purpose processors are not optimized for networking performance. This requires you to have more horsepower to provide the same function as a lower-end Juniper NetScreen firewall.

Q: You mention that each organization has different security needs. Why don't you provide a definite answer that can resolve my security issues?

A: Every organization has different types of requirements that they need to provide for their users. Some companies may have hundreds of Web servers, while others have only a few file-sharing servers. Some good baselines have been outlined that do a fine job of securing your resources, such as every organization's need for antivirus software; however, application-level protection may not be required for some organizations.

Q: Do I really need a DMZ? It only confuses my users.

A: Segregated networks should be a requirement for any company that has resources that must be accessible to the Internet or resources that everyone in the company does not need access to. The slight complexities that the DMZ creates simply do not outweigh its benefits.

Q: If I follow your guidelines for security, is this all I will ever need to secure my network?

A: Your organization's security requirements are something that should never be written in stone. You should always be on the lookout for new technologies and methodologies that can provide additional security to your environment.

Q: Why do I need to know so much about networking?

A: Knowing networking allows you to truly understand the risks that can occur in a network. When using networking, the more you know about it, the more options you give yourself. For example, if you were trying to build a house to provide protection against a hurricane, and all the knowledge you had concerned using sticks and straw to build with, your chances of building a successful house would be close to zero. However, if you were familiar with several construction styles, you would have more flexibility in choosing methods and materials that would provide you a better chance of creating a better house.

Chapter 2

Dissecting the Juniper Firewall

Solutions in this chapter:

- **The Juniper Security Product Offerings**
- **The Juniper Firewall Core Technologies**
- **The NetScreen and SSG Firewall Product Line**

- ☑ **Summary**
- ☑ **Solutions Fast Track**
- ☑ **Frequently Asked Questions**

Introduction

This chapter will introduce you to the Juniper firewall product. We will begin by looking at all of Juniper Networks' security products, exploring the wide range of products available, and allowing you to determine which is best suited for your security needs. A well-designed and properly implemented security infrastructure must be multitiered. Juniper Networks now offers a host of security solutions for your organization. Over the past several years Juniper has increased its product portfolio dramatically. Through both acquisition and internal development, Juniper has become a premier security vendor.

Juniper Networks delivers an integrated firewall and virtual private network (VPN) solution, the NetScreen firewall. The firewall product line has several tiers of appliances and systems. These tiers allow you to choose the right hardware for your network, giving the precise fit for your needs. Juniper has recently released a new firewall product line, the Security Services Gateway (SSG). This firewall line is designed to allow you to use new enhanced software features to better help protect your network from attack. Many of the SSG firewall products also enable you to use wide area network (WAN) interfaces as well.

Juniper also offers a Secure Sockets Layer (SSL) VPN product. The Secure Access series offers a clientless remote access solution as well as a collaboration tool. With a clientless VPN approach, you remove the need for software deployment and management of the remote clients. You can easily deploy the SSL portal to thousands of users in mere hours. This is a great boon to any organization. Also available in the SSL VPN product line is the secure meeting application, which allows for online collaborative meetings where users can share their desktops and engage in chat. These are secured by SSL. You can use this feature to conduct presentations or to perform remote support. It's a great tool for any organization.

In recent years, access control for desktop PCs has become increasingly important. In the past organizations have focused primarily on protecting servers from external threats. Today, new technologies allow companies to restrict access to the network itself, thereby allowing administrators to deny untrusted users from accessing the network and its available resources. Juniper today uses its Unified Access Control (UAC) product to address this industry need.

The last part to the security product line is *intrusion detection and prevention* (IDP). Whereas some products allow you only to detect incoming malicious traffic, the IDP allows you to fully prevent it from continuing on your network. The IDP is a necessary device for any network.

We will explore the core technologies of the Juniper firewalls. These are the frameworks that are used throughout this book. This discussion will give you an idea of the features of the Juniper firewall and will prepare you to actually implement these solutions on it. We will look at fundamental concepts such as *zones*. Zones are used to logically separate areas of the network. They allow you to take a more granular approach when you begin writing access rules to allow or deny network traffic.

In the last section of the chapter, we will look thoroughly at the NetScreen and SSG firewall products. The products range from small office devices that would allow for VPN

connectivity into a central location to the carrier class products that can serve as much as 12 gigabits per second (Gbps) of firewall traffic—a gigantic level of throughput for a firewall. The options provided in the Juniper firewall product line enable you to take your network to new heights.

The Juniper Security Product Offerings

NetScreen is the fastest-growing firewall product line on the market today. It has clinched the number two spot among the worldwide security appliance market. The NetScreen product line is robust and competitive, and it is now part of Juniper networks. On April 16, 2004, Juniper Networks completed its purchase of NetScreen for $4 billion. Juniper chose to purchase NetScreen to enter the enterprise market space. Previously, Juniper focused on the carrier class market for high-end routers. Juniper is aiming high; it is vying directly with Cisco for the position as the number one firewall appliance vendor, as well as the number one router vendor in the world.

- The Juniper firewall appliance is Juniper's firewall/VPN solution. Throughout the book, the firewall is referred to as a NetScreen firewall because Juniper chose to keep the NetScreen firewall product name for brand recognition. The other products in the NetScreen security line all kept their original names as well.
- The NetScreen IDP product is used to provide protection against network attacks. The IDP can alert you, log events, and capture attacks as they occur. This product offers several modes of operation that allow it to be used in one of several different network designs. It can also prevent against worms, viruses, and Trojans.
- The third part to the NetScreen security product line is the SSL VPN. The NetScreen Secure Access SSL VPN allows for clientless access into your network. The SSL VPN is currently the fastest-growing product line for Juniper. The Secure Access SSL VPN appliance is the market leader in its segment with 45 percent of the market share as of the first quarter of 2004. An offshoot from the SSL VPN product line is the secure meeting product. Secure Meeting can be integrated with the SSL VPN appliance, or it can be run on its own dedicated appliance. It provides Web conferencing collaboration to share your desktop and documents over the Web.
- The UAC product solution is the next generation of security. The UAC architecture provides network access control to client systems. The deployment architecture can be twofold. You can use the firewalls to provide enforcement or you can also use switches that are 802.1x compatible to provide access management to clients as well.

Juniper Firewalls

Juniper Networks' premier security platform is the NetScreen firewall product line. This product line provides integrated firewall and Internet Protocol Security (IPSec) VPN solutions in a single appliance. The NetScreen firewall core is based on stateful inspection technology. This technology provides a connection-oriented security model by verifying the validity of every connection while still providing a high-performance architecture. The NetScreen firewalls themselves are based on a custom-built architecture consisting of application-specific integrated circuit (ASIC) technology. ASIC is designed to perform a specific task at a higher performance level than a general-purpose processor. ASIC connects over a high-speed bus interface to the core processor of the firewall unit, a reduced instruction set computing (RISC) CPU.

The firewall platform also contains additional technologies to increase your network's security. First, the products support *deep inspection*. This technology allows you inspect traffic at the application level to look for application-level attacks. It can help prevent the next worm from attacking your Web servers or someone from trying to send illegal commands to your SMTP server. The deep inspection technology includes a regularly updated database as well as the capability for you to create your own custom expression-based signatures. All the appliances include the capability to create IPSec VPNs to secure your traffic. The integrated VPN technology has received both the Common Criteria and the ICSA (www.icsalabs.com) firewall certifications. Thus, the IPSec VPN technologies have good cross-compatibility as well as standards compliance. Juniper also offers two client VPN solutions to pair with the NetScreen firewall. First, NetScreen-Remote provides the user with the capability to create an IPSec connection to any NetScreen firewall or any IPSec-compliant device. The second client product is NetScreen-Security Client. This product not only creates IPSec tunnels but also includes a personal firewall to secure the end user's system. The NetScreen firewall product line leverages the technologies of Trend Micro's and Kaspersky Lab's antivirus software. This software allows you to scan traffic as it passes directly through the firewall, thus mitigating the risks of viruses spreading throughout your network.

The latest product set for the firewall line from Juniper is the SSG. The SSG product line was designed with key ideas in mind. First, it provides at high speeds advanced security features such as antivirus protection, antispam protection, IPS capabilities, and integrated URL filtering. Second, all the SSG products allow you to use WAN interfaces on the firewall, thereby enabling you to connect your firewall directly to a T1, digital subscriber line (DSL), or ISDN (Integrated Services Digital Network) link, to name a few. It gives you the capability to bypass the need to have a router on every WAN link. Because the SSG products are also built for future services, the architecture on the devices has changed from that of the traditional NetScreen firewall. SSG firewalls do not contain ASICs as the NetScreen firewalls do. However, this does not mean that the SSG firewall does not offer the same levels of performance as its cousin, the NetScreen product.

The Juniper firewall platform provides you with three management options:

- **Command-Line Interface (CLI)** The CLI provides the most granular control over the platform through straightforward interaction with the operation system (ScreenOS).
- **Web User Interface (WebUI)** The WebUI is a streamlined Web-based application with a user-friendly interface that allows you to easily manage the NetScreen appliance. Both WebUI and CLI are consistent among all the NetScreen firewall products—this means that once you have experience using one firewall model (for example, 5GT), you can easily apply your knowledge to other models (such as 208) in the NetScreen firewall product line.
- **NetScreen Security Manager (NSM)** A centralized enterprise-class solution that allows you to manage your entire NetScreen firewall infrastructure. The NSM provides not only a central console to manage your firewalls and Juniper IDP products but also consolidated logging and reporting. This is a great option that allows you to see all your network's activity from a central location.

SSL VPN

The need for remote access to a company's resources is at an all-time high. The traditional mode of the past was to use IPSec VPN clients. However, in many situations, the process of deploying and managing remote VPN clients is impractical, especially considering that you must maintain the software over time. Juniper Networks offers a product known as Secure Access SSL VPN. This product allows you to secure your internal resources behind a single entry point device. Remote users require only a Web browser capable of SSL encryption. The user connects to the SSL VPN gateway and begins his or her secure session. At this point, the user can access many different types of resources. This product provides secure ubiquitous client access, and because you do not have to deploy a client, you can easily deploy access to thousands of users in a matter of hours.

An important feature of the SSL VPN is client-side security. The SSL VPN offers several solutions that provide additional security to the end user's system. First, the Secure Access product offers the *host checker*. The host checker performs client-side checks for specific options. It can check to ensure the existence and the validity of any file on the client's system, such as an antivirus scanner or a personal firewall. It can check for the existence of specific registry settings as well. Finally, the host checker can tie into other third-party products and talk to the applications running on the client's system. One example would be the Sygate host integrity-checking client. It can ensure that the client meets or exceeds the company's defined standard for a remote software load. Based on the options that pass or fail, you can give the end user various levels of access.

If users had an antivirus program running on their systems, you could allow the users to access network file-shares, as well as first-level access of Web mail. If the users had a personal firewall running as well, they could safely be allowed to access servers using Microsoft terminal services—as well as the two other levels of access. The host-checking functions allow you to provide granular access to network resources based on the client's own security.

A second security application that users can run is called *cache cleaner*. Cache cleaner identifies all files cached by the Web browser and deletes them after the client's session is completed, thereby ensuring that no trace of the client's session remains on the remote system.

There are three levels of application access available through the SSL VPN device:

- **Tier 1** The first tier of access allows you to access Web-based and file-based resources. When you are accessing Web sites or Web-based applications, all HTML, JavaScript, and Java are rewritten to direct access through the SSLVPN gateway. This ensures that access to all resources can be secured, and not directed to another location without the administrator's explicit configuration. In the first level of access you can also access both common Internet fire system/server message block (CIFS/SMB) Microsoft file shares, as well as UNIX standards network file system (NFS) shares. Access to these resources is all done through a Web interface where you can upload and download files to modify them. Also included is a Java-based component that allows you to access systems via either Telnet or secure shell (SSH).
- **Tier 2** The second tier of access uses a component called secure application manager. Secure application manager runs as either a browser-based Java component or an Active-X component. This allows you to access resources via a client-initiated Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) unicast connection. You can use this to access various popular applications such as Microsoft Exchange, Citrix, and Microsoft Terminal Services.
- **Tier 3** The third tier of access allows full network connectivity, allowing the clients to connect into the network as if they were directly on the network. This allows systems to connect back to the client using applications such as X-Windows.

The SSL VPN product also provides a secure collaboration tool, *secure meeting*. Secure meeting allows users to share applications and collaborate from any Internet-connected computer. This is similar to services such as WebEx. It is a powerful tool to be used on an SSL VPN device, or as a stand-alone solution. The Secure Access products provide application-level logging, allowing you to record each user's activities. The Secure Access product line offers the enterprise the capability to integrate a remote access solution.

Intrusion Detection and Prevention

Juniper Networks' IDP product is a network appliance designed to provide intrusion detection and prevention for today's enterprise networks. It can be deployed in several different configurations to accommodate needed functionality. First, it can be deployed as a nonintrusive network traffic sniffer to detect incoming attacks, and to record, and alert on them. In this mode, you can easily install them to provide a network baseline, or to quickly replace

your other IDS devices. Second, you can install the device in gateway mode. In this mode, you can use the IDP as an active defense mechanism. In this mode, it provides several ways to stop network attacks. It can close or drop the connection on both the client and server sides. It can also capture the session for subsequent analysis. This product is great for doing forensic research on the attack.

The IDP platform provides a trademarked *multi-method detection* (MMD), which combines several detection mechanisms. Various applications require distinctively different methods of detecting attacks. The applied detection methods ensure that critical attacks are detected, providing you the information you need to identify network threats in your environment. The IDP currently provides nine different mechanisms to detect attacks.

- **Stateful signatures** Detect known attack patterns. This mechanism allows you to detect a greater number of attacks, and reduce the incidence of false positive alerts. A false positive is an alert that falsely detects an attack. A false positive can waste valuable time and resources.
- **Protocol anomaly detection** Reviews the different types of connections that go through the IDP and acknowledges any connections that deviate from the proper protocol standards. This can be used to detect new attacks and expose vulnerabilities.
- **Backdoor detection** Looks through interactive traffic for possible malicious communications. A backdoor is an application that resides on a host system unknown to the end user. These malicious applications, when installed on a user's system, can allow attackers to access your network's resources. When using the backdoor detection mechanism on your IDP, you can identify these intrusive connections and then block these connections to eliminate this harmful traffic.
- **Traffic anomaly detection** Allows you to look further than a single packet or a single session. It allows you to look across multiple sessions and identify anomalous traffic. An example of this type of traffic is a reconnaissance attack such as a port scan. A port scan is a series of sessions or connections that individually may not raise a red flag. However, when you add many of these probing packets together, this constitutes a traffic anomaly, which can be detected with this mode of operation.
- **Network honeypot** Mimics a system's services pretending to be a vulnerable system. This entices an attacker to access these services first, drawing attention away from your critical systems.
- **Layer-2 detection** Monitors network traffic on the second layer of the Open Systems Interconnect (OSI) model. This allows you to detect address resolution protocol (ARP) attacks on your network.

- **DoS detection** Allows you to detect certain types of Denial of Service (DoS) attacks. Denial of Service attacks can bring your network to its knees and early detection is critical to mitigate these attacks.
- **Spoofing detection** Provides the capability to detect spoofed IP packets. A spoofed IP packet is a packet that seems to be coming from a host, but really is coming from a malicious attacker.
- **Compound signatures** Combine multiple detection methods for complex attack detection. Juniper Networks combines stateful signatures and protocol anomaly to create a powerful detection mechanism.

Managing the IDP is simplified with its integrated management system, or by using Juniper's NSM (IDP software version 4.0 and later). Logging of attacks is detailed, providing extensive information in order to determine what is happening on your network.

The policy editor component, *Policy Editor,* is a graphical interface that allows for granular control over what type of traffic you want to detect and defend. Your configuration and signature information is readily available to use from within the policy editor. This allows you to easily create effective policies, thereby providing detection of, and prevention from, network attacks. The IDP product is an excellent product to use in your network to provide a new layer of security for your organization.

Unified Access Control (UAC)

Many security products focus on the security of the network at its perimeter. However, the need to focus on securing the accessibility of each client on the network is important. Imagine having the capability dynamically allow or deny access based upon each individual client, anywhere on the network. The UAC product set allows limited access to a network based on user credentials. This can be accomplished in one of two ways. In any network topology, you can use a Juniper firewall, or *Infranet Enforcer*, to provide access control at choke points in your network. Another option that you can use or use in conjunction with an Infranet Enforcer is an 802.1x switch infrastructure.

Using an 802.1x switch infrastructure allows you to authenticate users before the user even gets an IP address. This is highly secure because if the user fails to authenticate then they do not get an Internet Protocol (IP) address to allow access on the network. The UAC product uses a combination of both strategies to secure access on your network. Infranet Enforcers are used in locations of the network where you do not have 802.1x switching available. This provides a choke point type strategy where users' access is limited beyond the Infranet Enforcer. Combining 802.1x and the Infranet Enforcer strategy, you can secure your client infrastructure.

The Juniper Firewall Core Technologies

The Juniper firewall platform was designed from scratch, allowing the developers to develop new concepts for how a firewall works by combining both conventional, and original security approaches. These concepts are repeated throughout the book.

Zones

Zones are the core of the Juniper architecture. A zone can be defined as a logical area. There are several types of zones that can exist on a Juniper firewall. The first and most commonly used zone is the *security zone*. A security zone is a segment of network space where security measures are applied. These are used to determine the different network locations assigned to a Juniper firewall. For example, the most commonly used security zones are *trust* and *untrust*. The *trust zone* is assigned to the internal local area network (LAN) and the *untrust zone* is assigned to the Internet. The name of the zone is arbitrary, but is used to help the administrator determine what the zone is used for. Security zones are used in policy configuration and are a key component of them.

Another zone type is the *tunnel zone*. Tunnel zones are used in conjunction with tunnel interfaces. They are defined as a logical segment to which a VPN tunnel interface is bound. The last type of zone is a *function zone*. An example of a function is the *management zone* (MGT). It specifies that an interface is to be used only for management traffic and will not allow traffic to be routed over it. A function zone is defined as a physical or logical entity that performs a specific function. The use of zones allows you to clearly define the separation between two or more areas. The Juniper firewall product line provides for various and multiple usage of zones.

Virtual Routers

A firewall is nothing more than a security router. It essentially sends traffic from one location to another, determining the best path based on its routing table. The firewall has the capability to allow or deny traffic. The NetScreen firewall provides simple routing services, as you would expect, but it also offers much more. A normal device that uses IP has a single routing table. The routing table contains all of the known or learned routes. A NetScreen device uses the concept of the *virtual router* (VR).

A virtual router is a logical construct within a NetScreen device. It provides you with multiple routing tables on the same device. The virtual router has many uses. Virtual routers are bound to zones, and the zones are bound to interfaces. The NetScreen router will function much like a standard firewall device with one routing table. However, using two separate routing tables gives you the capability to separate your routing domain. For example, if you were to run Open Shortest Path First (OSPF) internally and Border Gateway Protocol (BGP) externally, you would have two separate routing domains. This allows you to securely separate your internally trusted routes from your externally untrusted routes. Later in the

book, we will discuss the configurations and virtues of virtual routers on Juniper firewalls in much more depth.

Interface Modes

As we have discussed, a Juniper firewall, by default, operates as a router. It allows each physical interface to use an IP address, allowing traffic to be forwarded between each interface. A Juniper firewall, however, is not limited to this traditional type of firewall configuration.

A Juniper firewall allows its physical interfaces to run in a special mode, *transparent mode*. Transparent mode allows you to put the Juniper firewall into layer two mode, which operates at the network layer. This allows a Juniper firewall to act similarly to a switch, while providing normal firewall filtering. This serves many purposes. If, for example, you have a flat network with one subnet, and no routing, but you want to separate your network and provide security for critical devices, you can install a Juniper firewall in transparent mode.

Policies

A *policy* is a statement that allows or denies traffic based on a defined set of specifications. The base specifications are source IP address, destination IP address, source zone, destination zone, and service or port. With this information, you can create a policy. There are three types of policies and a policy is classified as one or the other based on which zones are used in the policy: *intrazone, interzone*, and *global*. By default, there is an invisible global policy that denies all traffic from passing through the firewall. So if the traffic is not implicitly allowed by another policy, it is denied. Creating policies allows you to perform one of four actions on the traffic: allow the traffic, deny the traffic from passing, reject the traffic, or tunnel the traffic into a VPN. Allowing the traffic is the action you would want to use when the matching traffic is traffic you want to pass through the firewall. You would want to deny traffic when you want to prevent traffic from passing through your firewall. Rejecting the traffic stops the traffic; however, it allows the firewall to send a TCP reset, or a message stating, "Internet Control Message Protocol (ICMP) destination port unreachable." Finally, you would tunnel traffic when you want to permit traffic as well as place the traffic into a VPN tunnel. Each Juniper device has a limited number of policies. This can be a license restriction, capacity restriction or both. You cannot create new policies once you reach the maximum number of policies per device. Juniper enforces this to ensure the performance numbers on the specification sheets. It would not make sense to allow a low-end 5-GT appliance to run 40,000 policies, only to have the performance reduced to less then one Mbps. These restrictions are not modifiable, and they apply to each platform. There are many different elements involved in configuring an advanced policy. This includes traffic shaping, user authentication, network address translation (NAT), alarms, uniform resource locator (URL) filtering, and scheduling. There are a great number of configuration options.

Administering policies can be done in one of three ways: from the WebUI, CLI, or the NetScreen Security Manager (NSM). Each method creates the same result, but performing

each task is slightly different. On some competitive firewall products, using access lists can be frustrating. It can be a huge pain to reorder, view, and manage access lists. When the NetScreen platform was designed, it was calculated with those pains in mind. Once you start looking at the configurations in the next chapter, you will begin to understand the power of the NetScreen and its CLI.

VPN

All Juniper firewalls are also VPN devices. They can facilitate both site-to-site VPNs as well as client-to-site VPNs, or as Juniper calls them, *dial-up VPNs.* Juniper's NetScreen firewall supports all the standard elements that you would expect a VPN device to include. It supports Internet key exchange (IKE), authentication header (AH), encapsulating security payload (ESP), tunnel mode, transport, aggressive mode, quick mode, main mode, message-digest algorithm 5 (MD5), secure hash algorithm 1 (SHA-1), data encryption standard (DES), triple data encryption standard (3DES), advanced encryption standard (AES-128), and perfect forward secrecy (PFS), to name only a few. Juniper gives you several options when configuring a firewall on a Juniper appliance. There are two different methodologies: a route-based, or a policy-based VPN.

A *policy-based* VPN allows the creation of a VPN through a policy or rule. This is the traditional method and it is similar to other VPN products. This gives you a simplified method to create VPNs.

A *route-based* VPN uses a special type of virtual interface to connect via a VPN. This virtual interface, a *tunnel interface*, allows you to provide special types of services. It allows you to run routing protocols between these two virtual interfaces. You could run OSPF, which requires two devices to be directly connected. This, of course, would not normally be possible over the Internet, but if you create a route-based VPN between two NetScreen firewalls, this limitation for OSPF is removed because of this special virtual interface.

Intrusion Prevention

Today's firewalls have to provide much more then just your regular Layer 3 and Layer 4 inspection. Filtering your ports, protocols, and IP addresses no longer provides the security necessary for preventing sophisticated attacks. You need the capability to look inside the packet for specific data that indicates an attack. A packet product, such as an IDP, is far more capable of pointing out an attack than a basic firewall. Typically, any device designed to specifically provide a service would do a much better job than a multifunction device. There are many instances where the implementation of application layer inspection can be a great benefit to a network.

A smaller network may not have the same management needs, or financial means, to install an IDP device. The integration of application-level inspection may be a better fit. Application-level scanning of an integrated device can provide a second level of protection to your network by blocking specific attacks.

Deep packet inspection technology is the next step in the evolution of firewalls. Deep inspection allows you to inspect traffic at the application layer, relying on regular expressions (Regex) to determine malicious content in a packet. For example, if a worm spreading over the Internet attempts to exploit your Internet Information Server (IIS) Web server vulnerabilities by sending a harmful string of characters to your Web server. A custom signature can identify that attack string and stop it. By applying the custom signature to a policy, the traffic in that policy is inspected for that specific string. Deep inspection is truly the next jump in evolution for the firewall. Look to the future to provide much more strength in this field for development.

In recent years, Juniper has also released IDP modules for the NetScreen Integrated Security Gateways (ISG). These modules are hardware cards that are installed on the backplane of an ISG chassis. The firewall can then direct sessions into the IDP modules. This allows a single device to provide firewall, and IPS features in a single unit. More importantly, this is done on independent hardware modules providing the best possible throughput for your network.

Are you 0wned?

Application-Level Inspection

Firewalls have conventionally focused on layer three and layer four filtering. This means that the connection is filtered based only on IP addressing, TCP, UDP ports, and options set at those layers. This can prevent unwanted systems from accessing your servers. What do you do when an attacker uses your firewall configuration against you? Suppose, for example, that your firewall is blocking all ports except for the HTTP port.

The attacker slips through your allowed port and manipulates your Web application without detection by your firewall. It is unaware of attacks at the application level. Regardless of the fact that your Web server is on a separate demilitarized zone (DMZ) than your database server, the attacker uses your Web application to access the secured database, and steals your customers' credit card information, and identities. If you think that this is nothing more than a *good* story, think again. This method of attack goes on everyday, and many organizations are not yet aware of the threat. Skilled persons who understand Web applications and design can easily snake through your applications and extract data from your database.

So does this mean that you need to disable all access to your Web server, and dismantle your e-commerce efforts? Of course not. You must, however, use security products that provide application-level inspection to identify these

Continued

attacks. The best method is to perform a penetration test on your application to determine what type of vulnerabilities your applications may be susceptible to. Next, begin implementing products that can determine attacks from normal traffic. The deep inspection software integrated into the NetScreen firewall can help protect against many of the unstructured attacks that can potentially do damage via your Web server. However, structured attacks require a stronger tool such as the IDP to mitigate the risks of these attacks.

Using tools such as IDPs and the deep inspection technology is not something that you just turn on and hope for the best. To make this type of application-level inspecting technologies work effectively you need to tune them for your network. This can take a great deal of effort and time to ensure that your network is using these devices effectively. Many times organizations purchase external devices in an attempt to secure a system of poorly written applications. Many times safe programming techniques can enhance the security of your applications.

Device Architecture

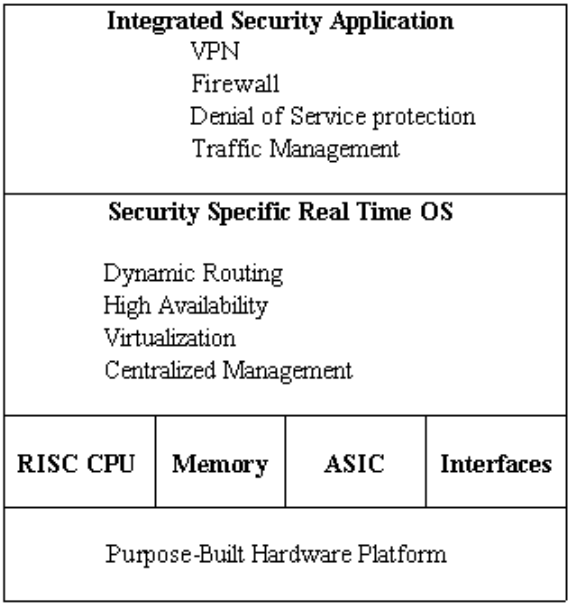
The device's hardware architecture was developed as a purpose-built device. Developed from the ground up to provide exceptional throughput, the firewall devices provide a level of security that leads the pack in firewall design. Juniper's NetScreen firewall product line is a layered architecture depicted in Figure 2.1. It is designed to provide optimal performance for critical security applications. The top layer of the NetScreen firewall architecture is the integrated security application. This application is integrated with the operating system to provide a hardened security solution. The integrated security application provides all of the VPN, firewalling, denial-of-service, and traffic management.

The second layer in the NetScreen firewall platform is the operating system. The operating system for the NetScreen firewall product, *ScreenOS*, was designed as a real-time operating system (RTOS). A real-time operating system is defined as an operating system that can respond to external world events in a time frame defined by the external world. Because only one task can run at a time for each CPU, the idea is to minimize the time it takes to set up and begin executing a task. A large challenge for RTOS is memory allocation. Allocating memory this takes time, which can slow down the OS from executing a task. ScreenOS preallocates memory to ensure that it will have enough memory to provide a sustained rate of service. ScreenOS is more secure than open source operating systems because the general public cannot inspect the source code for vulnerabilities. ScreenOS also does not have the exposure of Microsoft Windows. Consequently, fewer people have been exposed to SecureOS, thereby denying them the opportunity to learn about the operating system (OS), or possible uses for it. The OS on a NetScreen firewall provides services such as dynamic routing, high-availability, management, and the capability to virtualize a single device into multiple virtual devices.

The third layer in the NetScreen architecture comprises the hardware components themselves. The firewall connects all of its components together with a high-speed multibus configuration. The bus connects each ASIC with a RISC processor, synchronous dynamic random access memory (SDRAM), and the network interfaces. An ASIC is a chip designed for a single purpose. A single-purpose chip performs tasks much faster than a general purpose microprocessor chip. The NetScreen firewall architecture has been designed to provide the features that a firewall running on a general-purpose operating system cannot. It is not limited by connection table restrictions, and processing limitations found in firewall designed for general-purpose hardware, and general-purpose operating systems.

In the new SSG firewall product line, Juniper chose not to include ASIC processors in the devices. Using other components Juniper can provide the same high level of throughput and reliability without a specialized chip. This design allows devices to be more cost-effective for the consumer yet provide the same solid performance as the older platforms. The SSG architecture is designed to best perform while providing the new security features, Unified Threat Management (UTM). These four features: Antivirus, Antispam, IPS/DI, and Web Content Filtering are available on each member of the SSG platform at maximum possible throughput. All SSG products have the option of field upgradable memory. Each product has two memory configurations: a base memory and max memory configuration. The max memory option is required to provide the UTM features.

Figure 2.1 The NetScreen Device Architecture



The NetScreen and SSG Firewall Product Line

The Juniper Firewall product line has several tiers of products that span over its entire product line. One of the great parts of the NetScreen firewall product line is that no matter what tier of device, the configuration of each device remains similar. This allows you to configure each device as you would the other. Every device supports the same three management options; the WebUI, CLI, and NSM configuration of each device is relatively similar. However, the higher up the firewall product line, the more ports and options you will get to use.

Each firewall device is configured by using the same methods, no matter what tier the device is in. Other vendors offer inconsistent configurations among their devices, but the NetScreen remains unvarying. Each device is a purpose built platform to provide specific functions for which it is designed. All of the devices use flash memory as the long-term storage option. None of the firewalls relies on a hard disk to run.

The NetScreen-Security manager provides lasting storage for the firewall devices, eliminating the need for long-term storage on the devices for logs. You can also stream logs to a syslog server for storage.

Product Line

In this section, we review the products in the current Juniper firewall line, starting with the low-end devices, and finishing with the high-end products. At the end of this section, we review enterprise management options that Juniper Networks has to offer. In Table 2.1, you can see the layout of the product line from the low end to the high end.

Table 2.1 Juniper Networks' Firewall Product Line Overview

Product Name	Product Class	Max Interfaces Ethernet / Wan	Throughput
NetScreen-Remote VPN Client	Remote Client Software	N/A	N/A
NetScreen-Remote Security Client	Remote Client Software	N/A	N/A
NetScreen-Hardware Security Client	Small Office Home Office	5 / 0	50 Mbps
NetScreen-5-XT	Small Office Home Office	5 / 0	70 Mbps
NetScreen-5-GT	Small Office Home Office	5 / 1 ADSL	75 Mbps
SSG 5	Small Office Home Office	7 / 1 (v.92, ISDN, RS232)	90 Mbps

Continued

Table 2.1 continued Juniper Networks' Firewall Product Line Overview

Product Name	Product Class	Max Interfaces Ethernet / Wan	Throughput
SSG 20	Small Office Home Office	6 / 2 (v.92, ISDN, DSL,T1,E1)	90 Mbps
NetScreen-25	Mid Range	4	100 Mbps
NetScreen-50	Mid Range	4	170 Mbps
SSG 140	Mid Range	10 / 8 (2xT1, 2xE1, 2xSerial, 1xISDN)	350 Mbps
NetScreen-204	High Range	4	400 Mbps
NetScreen-208	High Range	8	550 Mbps
SSG 520	High Range	12 / (2xT1, 2xE1, 2xSerial, 1xDS3)	600 Mbps
NetScreen-500	Enterprise Class	8	700 Mbps
SSG 550	Enterprise Class	20 / (2xT1, 2xE1, 2xSerial, 1xDS3)	1 Gbps
NetScreen-ISG 1000	Next Gen Enterprise Class	20	1 Gbps
NetScreen-ISG 2000	Next Gen Enterprise Class	24	2 Gbps
NetScreen-5200	Carrier Class	26	10 Gbps
NetScreen-5400	Carrier Provider Class	78	30 Gbps

Tools & Traps…

Choosing the Right Tool for the Job

When you plan to purchase a NetScreen device, consider your future needs as well as current needs, because most devices cannot be upgraded. Realistically, you should look at the life of the product over the next three years. This will provide you for the right amount of growth for your network. The NetScreen-208 product would serve most companies well. Equipped with eight total interfaces and featuring up to 700Mbps throughput, it can provide solutions for most networks.

Continued

In many lower-end networks where you have just an internal LAN and an Internet connection only four interfaces, a lower level of throughput would be required. Even the lowest end NetScreen firewall device can easily handle even a hefty DS3 circuit to the Internet providing 45 Mbps. This said, choosing a firewall can be hard work. Because of the low upgrade capability of these devices, many people looking at a device such as NetScreen might think twice. However, as you can see with the large selection, a proper selection of a device can easily overcome your cognitive dissidence when choosing a NetScreen firewall.

The SSG product line also supports WAN interfaces. You have the option of dispensing with the need for a router. There are a number of options in Table 2.1. Juniper's firewalls support simple serial interfaces through to a DS3 interface.

NetScreen-Remote Client

Juniper Networks' NetScreen-Remote Client line includes two products:

- NetScreen-Remote VPN Client
- NetScreen-Remote Security Client

Remote access to company resources is a requirement for most organizations. Company resources *must* be secured. For remote access security, Juniper offers NetScreen-Remote VPN client, and NetScreen-Remote Security client. These products provide an easy-to-use interface to configure and connect to IPSec gateway endpoints. You are not limited to client access of the NetScreen-based VPN firewalls. It is capable of connecting to any IPSec gateway. Providing standards-based IPSec connectivity is just part of the NetScreen-Remote VPN client. The XAuth Extended Authentication protocol is also supported by NetScreen Remote. XAuth supports delivery of IP addresses and DNS (domain name system) settings to a virtual interface on the client. The Remote VPN client is capable of supporting up to 100 concurrent IPSec VPN tunnels. NetScreen-Remote VPN and Security clients provide easy, secured access to your mobile workforce.

NetScreen-Remote Security client has an integrated client firewall to protect the remote users system. This client allows the end user to connect securely to the enterprise network over the industry-standard IPSec. The interface of the client allows the user to easily configure a VPN connection. It also provides the administrator with the capability to create and then export a VPN policy that can be deployed to all remote users. The crowning feature of the security client is the integrated firewall. This allows you to protect the end user's system from intrusions and network attacks. Not only does this protect the end user's system, but it protects your company's network by preventing malicious attackers from connecting through a VPN client's system through to the company's network.

Small Office/Home Office (SOHO)

Juniper Networks' SOHO line includes the following products:

- NetScreen-Hardware Security Client
- NetScreen 5XT
- NetScreen 5GT
- SSG 5
- SSG 20

For remote locations or remote users with a need for a dedicated security appliance, the SOHO line of NetScreen firewall appliances provide enterprise-class security at a relatively affordable price. This product provides great power in a small footprint. These devices support the easy-to-use CLI and WebUI management interfaces that the high-end appliances and systems do. The SOHO product line is illustrated in Table 2.2.

Table 2.2 Juniper Networks' SOHO Product Line

	Hardware Security Client	5-XT	5-XT Elite	5-GT 10 User	5-GT Plus	5-GT Extended	SSG 5	SSG 20	SSG 5 Extended	SSG 20 Extended
Interfaces	5 10/100 Ethernet	5 10/100 Ethernet	5 10/100 Ethernet	5 10/100 Ethernet	5 10/100 Ethernet	5 10/100 Ethernet	7 10/100 Ethernet	5 10/100 Ethernet	7 10/100 Ethernet	5 10/100 Ethernet
Wan Interfaces (Optional On 5GT)	N/A	N/A	N/A	ADSL	ADSL	ADSL	1 (v.92, ISDN, RS232)	2 (v.92, ISDN, ADSL, T1,E1)	1 (v.92, ISDN, RS232)	2 (v.92, ISDN, ADSL, T1,E1)
Wireless (Optional On 5GT)	N/A	N/A	N/A	802.11 a/b	802.11 a/b	802.11 a/b	802.11 a/b/g	802.11 a/b/g	802.11 a/b/g	802.11 a/b/g
Max IP address behind	5	10	No limit	10	No limit	No limit	No Limit	No Limit	No Limit	No Limit
Maximum Throughput	50 Mbps					75 Mbps	90 Mbps	90 Mbps	90 Mbps	90 Mbps
Firewall	10 Mbps	70 Mbps	70 Mbps	75 Mbps	75 Mbps	20 Mbps	50 Mbps	50 Mbps	50 Mbps	50 Mbps
VPN		20 Mbps	20 Mbps	20 Mbps	20 Mbps					
Maximum Sessions	1,000	2,000	2,000	2,000	2,000	4,000	4,000	4,000	8,000	8,000
Maximum VPN Tunnels	2	10	10	10	10	25	25	25	50	50
Maximum Policies	50	100	100	100	100	100	200	200	200	200
Virtual Systems	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Security Zones	3	3	3	3	3	3	10	10	10	10
Virtual Routers	2	2	2	2	2	2	3	3	3	3
VLANs **(Trust-Untrust port mode only)**	N/A	N/A	N/A	10	10	10	10	10	50	50

Continued

Table 2.2 continued Juniper Networks' SOHO Product Line

	Hardware Security Client	5-XT	5-XT Elite	5-GT 10 User	5-GT Plus	5-GT Extended	SSG 5	SSG 20	SSG 5 Extended	SSG 20 Extended
Routing Protocol Support										
RIP v2	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
OSPF	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
BGP	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
High Availability	N/A	N/A	N/A	N/A	N/A		N/A	N/A		
HA Lite										
Active/Passive						Yes			Yes	Yes
Active/Active						No			Yes	Yes
Active/Active						No			No	No
Full Mesh						No			No	No
Antivirus Scanning										
Embedded	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
External	No	No	No	No	No	No	No	No	No	No
Deep Inspection	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Antispam	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Web Filtering	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

The NetScreen Hardware Security Client is at the low end of NetScreen's firewall product line. This device is designed as a hardware-based version of the remote software client. It still provides huge throughput for being the lowest performing device. This device passes a maximum of 50 Mbps for its firewall performance and 10 Mbps for a 3DES VPN. These numbers can easily support the fastest residence-installed broadband connection. Protecting home users from viruses is easy with this device. It includes Kaspersky's scan engine embedded directly into the device. This allows you to scan Post Office Protocol 3 (POP3), Internet Message Access Protocol (IMAP), File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), Hypertext Transfer Protocol (HTTP) Web mail in real time to protect users from viruses. This is a great way to prevent infected home users from spreading viruses to the company's network. The newest firewall technology, deep inspection, is supported to help protect against application-level attacks and vulnerabilities. The NetScreen-Hardware security client has to be managed from a NetScreen-Security Manager. Policies cannot be created on the device absent the NetScreen-Security Manager.

The next product in the small office, home office (SOHO) product line is the NetScreen 5-XT. This device has many more capabilities then the NetScreen-Hardware security client. It allows up to 70Mbps firewall performance, as well as 20Mbps 3DES VPN performance. The 5-XT supports the deep inspection application level, scanning for attacks at up to a maximum of 55 Mbps. It has a total of five 10/100 Ethernet ports. Two ports can be connected to the Internet to provide redundant Internet connectivity. However, if you require total uptime, the 5-XT also supports the capability to connect an external modem providing dial backup capabilities. The 5-XT does not support any sort of antivirus capabilities, though. This is an older product that is being replaced by the 5-GT.

The NetScreen-5-GT is the answer to your needs if you need a low-end remote appliance. Truly, the only two low-end things about this device are the price and the model number. The 5-GT provides a speedy 75Mbps firewall throughput, a full 75Mbps deep inspection scanning, and 20Mbps 3DES VPN performance. This device is similar in design to the 5-XT and has five 10/100 Ethernet ports. Again, two Internet-connected interfaces to provide redundant connectivity in case one Internet service provider experiences a failure. A modem port is provided to allow for dial-up Internet connectivity as well. In ScreenOS release 5.3 and later, the 5GT supports the creation of up to 10 VLAN subinterfaces. This is only supported in Trust-Untrust mode.

The 5-GT has Kaspersky's antivirus engine embedded, allowing for inline virus scanning of the POP3, IMAP, FTP, SMTP, and HTTP Web mail protocols. This is a separately licensed subscription. The 5-GT has a separate model, which is the same as the aforementioned features, but it also contains an asynchronous digital subscriber line (ADSL) port. This model is the 5-GT ADSL. The 5-GT has several different licensing choices to give you a range of options on this low-end appliance. On a 5GT appliance, you can configure several different port modes. A port mode configures the device to use a group of ports together into a single zone. This gives you the flexibility to better use zones on the device. Instead of just using four ports for the trust zone and one for untrust, you can use them in various ways. Port modes are covered in more depth in Chapter 3.

- **10-User License** Allows for only 10 users behind the 5GT to access through the device at one time
- **Unlimited User License** Allows for an unlimited about of users to access through the device at one time
- **5GT Extended** This licensing option provides up to 4,000 concurrent sessions. It allows you to create a DMZ on the firewall through the use of a dedicated DMZ port. The extended license also allows for a two-unit high-availability cluster with two 5GT Extended firewalls.

The SSG firewall product line is the next step up in the product line for the small office/home office product set. These products really are much more powerful then should fit into this low category. These products provide much more flexibility then the predecessor products. The first of these two products is the SSG 5. The SSG 5 and the SSG 20 are very similar to the 5GT in design. However, there have been some significant improvements in the design of the products. The first focus of the SSG products is the WAN interfaces. On the SSG 5, there are three options for WAN interfaces built into the unit. These interfaces are noted in Table 2.2 and are fixed ports that are installed in the factory. The SSG 20, however, has two separate WAN interface slots. The interface types are also listed in Table 2.2.

In the SSG firewall line, they also include the capability to create bridge groups. This is a more flexible implementation of the port mode option found in the 5GT. Bridge groups allow you to take one or more ports and place them into a group. This group can then have a zone bound to them. This allows for more flexibility on your device. Between the bridge groups, and the additional physical ports and VLAN support the SSG products are extremely versatile devices. The performance of the SSG 5 and the SSG 20 is not much more then the performance found in the 5GT product line. However, in most situations where these devices are deployed you will not require more performance. You also have the option to purchase the extended license key. This extends the capability of the device to increase the number of sessions, VPN tunnels, and VLANs. If you do need more performance, however, you can move into the midrange product line for the firewalls. The SSG 5 and SSG 20 come with two memory configurations: 128MB and 256MB. You can purchase this preinstalled version or install it after the fact. You are required to have the 256MB configuration to use the UTM features.

Midrange

Juniper Networks' midrange line includes the following products:

- NetScreen-25
- NetScreen-50
- SSG 140

The NetScreen-25, NetScreen-50, and SSG 140 are the next step up the Juniper Firewall ladder (see Table 2.3). These devices are a perfect fit for branch and remote offices, or for medium-sized and small companies. The only difference between the NetScreen-25 and NetScreen-50 devices is the performance that they provide. Both devices are physically

identical otherwise. The SSG 140 is a new class of device that provides more performance and the flexibility of WAN interfaces as well.

Table 2.3 Juniper Networks' Midrange Line of Firewalls

	NS-25 Baseline	NS-25 Advanced	NS-50 Baseline	NS-50 Advanced	SSG 140
Interfaces	4 10/100 Ethernet	4 10/100 Ethernet	4 10/100 Ethernet	4 10/100 Ethernet	8 10/100 Ethernet 2 10/100/1000 Ethernet 4 WAN Slots
WAN Interfaces					ISDN, Serial, T1, E1
Maximum Throughput					
Firewall	100 Mbps	100 Mbps	170 Mbps	170 Mbps	300 Mbps IMIX
VPN	20 Mbps	20 Mbps	45 Mbps	45 Mbps	100 Mbps IMIX
Maximum Sessions	24,000	32,000	48,000	64,000	32,000
Maximum VPN Tunnels	50 Shared	25 Site to Site			
100 Dial-Up	150 Shared	500 Shared	125 Shared		
Maximum Policies	500	500	1,000	1,000	500
Virtual Systems	N/A	N/A	N/A	N/A	N/A
Security Zones	4	4	4	4	40
Virtual Routers	3	3	3	3	3
VLANs	0	16	0	16	100
Routing Protocol Support					
RIP v2	Yes	Yes	Yes	Yes	Yes
OSPF	No	Yes	No	Yes	Yes
BGP	No	Yes	No	Yes	Yes
High Availability					
HA Lite	Yes	Yes	Yes	No	Yes
Active/Passive	No	No	No	Yes	Yes
Active/Active	No	No	No	No	No
Active/Active Full Mesh	No	No	No	No	No
Antivirus Scanning					
Embedded	No	No	No	No	Yes
External	No	No	No	No	No
Deep Inspection	Yes	Yes	Yes	Yes	Yes
Antispam	Yes	Yes	Yes	Yes	Yes
Web Filtering	Yes	Yes	Yes	Yes	Yes

The NetScreen-25 is the weaker of the two devices in the midrange category. The NetScreen-25 provides a total of 100Mbps firewall performance, 20Mbps 3DES VPN per-

formance, and up to 75Mbps deep inspection performance. It has a total of four 10/100 Ethernet ports. It also has a console port and modem port. The console port provides access for console CLI management. The modem port allows you to connect a modem for out-of-band management capabilities. The NetScreen-25 (and all later model devices) allows you to configure network ports tailored to your needs. This gives you control over your network, and it provides multiple configuration options. You can have four separate security zones for these interfaces. The NetScreen-25 device allows only for high-availability (HA) Lite mode. This mode will provide failover in case of a hardware failure. However, it will not allow you to fail all of your active sessions. All active sessions are lost when a device fails over to the backup device when you are using an HA Lite configuration. The NetScreen-25 comes in two licensed models: baseline and advanced. In Table 2.3, you can see the performance benefits of using the advanced feature set.

NetScreen-50 is the high-performer of the two devices in the midrange category. The NetScreen-50 provides a total of 170Mbps firewall performance, 45Mbps 3DES VPN performance, and up to 75Mbps deep inspection performance. It has a total of four 10/100 Ethernet ports that you can use. It also has a console port and modem port. The console port provides access for console CLI management. The modem port allows you to connect a modem for out-of-band management capabilities. The NetScreen-50 device allows for high-availability in Active/Passive mode. This mode would provide for failover in case of a hardware failure. It would also failover all your sessions for a seamless failover. The NetScreen-50 has two licensed models: the baseline and advanced. In Table 2.3, you can see the performance differences between the baseline and the advanced models.

The new SSG 140 product is truly a leader in its class. The SSG 140 provides unmatched connectivity matched with feature rich software options. The SSG 140 provides eight 10/100 Ethernet ports and two 10/100/1000 Ethernet ports. Besides the 10 Ethernet ports the firewall also has the capability to use up to four pluggable interface modules (PIMs). Each PIM is capable of using a combination of T1, E1, ISDN, or Serial ports. The SSG 140 product provides up to 300Mbps throughput for the firewall and 100Mbps VPN throughput. A much-needed upgrade from the NS-25 and NS-50 is the additional VLAN and security zone support. An amazing 40 security zones and up to 100 VLANs are available. As the SSG 5 and SSG 20 discussed the SSG 140 also supports the UTM features. The SSG 140 comes in two memory configurations: 256MB and 512MB. You can purchase this preinstalled, or you can install it after the fact. You must have the 512MB configuration to use UTM features.

High End

Juniper Networks' high-end line includes the following products:

- NetScreen-204
- NetScreen-208
- SSG 520

The high-end line of NetScreen products is shown in Table 2.4.

Table 2.4 Juniper Networks' High-End Line of Firewalls

	NS-204 Baseline	NS-204 Advanced	NS-208 Baseline	NS-208 Advanced	Virtualization License	SSG 520
Interfaces	4 10/100 Ethernet	4 10/100 Ethernet	8 10/100 Ethernet	8 10/100 Ethernet	N/A	4 10/100/1000 Ethernet 4 PIM Slots 2 EPIM Slots
WAN Interfaces (EPIM or PIM Slot)						Serial, T1, E1, DS3
EPIM Slot Card Support						Serial, T1, E1, DS3, 10/100/1000, SFP, FE Serial, T1/E1, DS3
Maximum Throughput Firewall VPN	 400 Mbps 200 Mbps	 400 Mbps 200 Mbps	 550 Mbps 200 Mbps	 550 Mbps 200 Mbps	 N/A	 600 Mbps IMIX 300 Mbps IMIX
Maximum Sessions	64,000	128,000	64,000	128,000	N/A	64,000
Maximum VPN Tunnels	500	1,000	500	1,000	N/A	500
Maximum Policies	4,000	4,000	4,000	4,000	N/A	1000
Virtual Systems	N/A	N/A	N/A	N/A	N/A	N/A
Security Zones	4	4	8	8	+10	60
Virtual Routers	2	2	2	2	+5	5
VLANs	0	32	0	32	+32	125

Continued

www.syngress.com

www.syngress.com

Table 2.4 continued Juniper Networks' High-End Line of Firewalls

	NS-204 Baseline	NS-204 Advanced	NS-208 Baseline	NS-208 Advanced	Virtualization License	SSG 520
Routing Protocol Support					N/A	
RIP v2	Yes	Yes	Yes	Yes		Yes
OSPF	No	Yes	No	Yes		Yes
BGP	No	Yes	No	Yes		Yes
High Availability					N/A	
HA Lite	No	No	No	No		Yes
Active/Passive	Yes	Yes	Yes	Yes		Yes
Active/Active	No	Yes	No	Yes		No
Active/Active Full Mesh	No	No	No	Yes		No
Antivirus Scanning					N/A	
Embedded	No	No	No	No		No
External	Yes	Yes	Yes	Yes		Yes
Deep Inspection	No	Yes	No	Yes	N/A	Yes
Throughput	**N/A**	180 Mbps	N/A	180 Mbps		300 Mbps
Antispam	No	No	No	No	No	Yes
Web Filtering	No	No	No	No	No	Yes

The NetScreen-200 series is the first model of high-end NetScreen features. This is the first series of devices that support an active/active high-availability configuration. This allows both NetScreen appliances in a high-availability cluster to be active at the same time, which allows for higher throughput, and maximum capacity. This class of firewall is typically required for one of three reasons. First, you require the use of more than four interfaces, similar to NetScreen-208. Second, you require higher throughput on these devices. Third, you require the more advanced features available for the NetScreen-200 series.

The NetScreen-204 has the same genetic make up as the NetScreen-25 and NetScreen-50. The NetScreen-204 provides double the performance of the NetScreen-50, providing 400Mbps firewall performance, 200Mbps 3DES VPN performance, and 180Mbps deep inspection capability. Much like the other devices of the same form factor, this device provides four 10/100BaseT ports. It also has a console port and modem port. The console port provides access for console CLI management. The modem port allows you to connect a modem for out–of-band management capabilities. This is the first platform that allows a function in Active/Passive mode or Active/Active mode. Antivirus scanning is performed via an external Trend Micro antivirus server. The NetScreen-204 comes in two licensed models: baseline and advanced. In Table 2.4, you can readily see the performance advantages gained by using the advanced featured set. You can also purchase a virtualization license for this platform. This provides 32 additional virtual LANs (VLANs), 10 additional security zones, and five additional virtual routers.

The NetScreen-208 comes with a similar one-rack unit form factor but it is the first device to have more than four physical interfaces. The NetScreen-208 offers impressive performance by all security standards. Providing 550Mbps firewall performance, 200Mbps 3DES VPN performance, and 180Mbps deep inspection capability, it also offers support for up to 128,000 concurrent sessions. The NetScreen-208 has the capability to easily support an e-commerce deployment. This device provides eight 10/100BaseT ports. It also has a console port and modem port. The console port provides access for console CLI management. The modem port allows you to connect a modem for out of band management capabilities. This enables you to use a Personal Computer Memory Card International Association (PCMCIA) flash card to back up your configuration. This is the first platform that allows you to have an Active/Passive, Active/Active, and Active/Active Full mesh configuration. Antivirus scanning is performed via an external Trend Micro antivirus server. The NetScreen-208 comes in two licensed models: baseline and advanced. In Table 2.4, you can see the performance advantages gained by using the advanced featured set. You can also purchase a virtualization license for this platform. NetScreen-208 provides 32 additional virtual LANs (VLANs), 10 additional security zones and five additional virtual routers (VRs).

The SSG 520 differs from the other products in this category. However, in today's day and age the need for flexibility at every location in the network is a requirement. The SSG 520 provides you with four 10/100/1000 ports that are built into the product. In addition, the devices come with the capability to add a mix of additional LAN and WAN ports. The SSG 520 comes with four PIM and two Enhanced Pluggable Interface Module (EPIM) slots. A PIM slot can use a WAN interface (T1, E1, DS3, or Serial) module. In an EPIM slot,

you can use a LAN (10/100/1000, SFP, FE Serial, T1/E1, DS3), or a WAN module. This allows the firewall to aggregate WAN interfaces while still providing the capability to provide high throughput LAN firewall features. The SSG 520 just as the rest of the SSG products provides the full UTM feature set. The SSG 520 can support up 125 VLAN interfaces and 60 security zones. That is more than double the number supported by the NS-208 firewall. The SSG 520 has two memory options: 256MB and 1GB. The 1GB memory option is required to support UTM features.

Enterprise Class

Juniper Networks' enterprise class includes the following products:

- NetScreen-500
- SSG 550
- NetScreen-ISG 1000
- NetScreen-ISG 2000

If you are looking for a high-performance, readily available, and expensive platform, then the enterprise class of NetScreen products is where you should begin browsing (see Table 2.5). There are two devices similar in design, with one outclassing the other in number of features. Both systems are the first devices in the NetScreen firewall line to provide redundant power supplies. This is a great option when uptime is crucial. Both devices have interchangeable interface modules. These modules allow you to have either 10/100BaseT ports, or Gigabit fiber ports. Copper Gigabit ports are not supported at this time: only fiber connections are supported.

Table 2.5 Juniper Networks' Enterprise-Class Product Line

	NS-500 Baseline	NS-500 Advanced	SSG 550
Interfaces	8 10/100 Ethernet or 8 Mini-GBIC or 4 GBIC	8 10/100 Ethernet or 8 Mini-GBIC or 4 GBIC	
WAN Interfaces (EPIM or PIM Slot)			Serial, T1, E1, DS3
EPIM Slot Card Support			Serial, T1, E1, DS3, 10/100/1000, SFP, FE Serial, T1/E1, DS3
Maximum Throughput			
Firewall	700 Mbps	700 Mbps	1 Gbps IMIX
VPN	250 Mbps	250 Mbps	500 Mbps IMIX
Maximum Sessions	128,000	250,000	128,000

Continued

Table 2.5 continued Juniper Networks' Enterprise-Class Product Line

	NS-500 Baseline	NS-500 Advanced	SSG 550
Maximum VPN Tunnels		1,000	5,000 and
5,000 Dial-Up	1000		
Maximum Policies	20,000	20,000	4000
Virtual Systems	Up to 25	Up to 25	N/A
Security Zones	8	8	60
Virtual Routers	2	2	8
VLANs	100	100	150
Routing Protocol Support			
RIP v2	Yes	Yes	Yes
OSPF	No	Yes	Yes
BGP	No	Yes	Yes
High Availability			
HA Lite	No	No	Yes
Active/Passive	Yes	Yes	Yes
Active/Active	No	Yes	Yes
Active/Active Full Mesh	No	Yes	Yes
Antivirus Scanning			
Embedded	No	No	Yes
External	Yes	Yes	Yes
Deep Inspection	No	Yes	Yes
Throughput	N/A	180 Mbps	600 Mbps
Antispam	No	No	Yes
Web Filtering	No	No	Yes

The NetScreen-500 is truly an enterprise-class device. This tool is capable of providing a highly available firewall scenario. First, it allows you to have redundant power supplies. This is essential when managing a network that requires 100 percent uptime. Secondly, components, like fans are also redundant to ensure that this device does not overheat. Finally, you can have high-availability interfaces to ensure you never have downtime. As far as high-availability modes go, the NetScreen-500 supports all three modes: Active/Passive, Active/Active, and Active/Active Full Mesh. When using a NetScreen device in high-availability mode, you need to have ports dedicated to enable both a heartbeat and the passing of session synchronization information. The NetScreen-500 provides two ports dedicated only to this purpose.

The NetScreen-500 has very large performance numbers, providing 700Mbps firewall performance, 250Mbps 3DES VPN throughput, and 300Mbps performance while doing deep inspection. It supports up to 250,000 concurrent sessions and up to 18,000 new ses-

sions per second. This is the first device in the NetScreen firewall line that can have a modular interface configuration. The NetScreen-500 can provide up to eight 10/100BaseT Ethernet ports, eight mini-GBIC (SX or LX) ports, or four GBIC (SX or LX) ports. This is not an overly dense port configuration, but it is the lowest-end device to provide for Gigabit ports. Virtual Systems (VSYS) are supported on this appliance and on all later-model appliances. A VSYS allows you to segment a device into several virtual systems. These virtual systems allow you to have a completely separate management domain to provide a virtual firewall.

The NetScreen-500 has two separate licensing modes: baseline and advanced. Table 2.5 includes the differences in the devices. You may also purchase virtual systems in three separate options: an upgrade to five virtual systems, upgrades from five virtual systems to 10 virtual systems, and from 10 virtual systems to 25 virtual systems are all available. NetScreen-500 GPRS is a subsequent version of NetScreen-500. This device allows you to secure the general packet radio services (GPRS) protocol as well. The performance of the device is similar to NetScreen-500.

The SSG 550 is similar in design to the SSG 520. However, the SSG 550 also provides you with four 10/100/1000 ports that are built into the product. The devices come with the capability to add a mix of additional LAN and WAN ports. The SSG 550 comes with two PIM, and four enhanced pluggable interface module (EPIM) slots. A PIM slot can use a WAN interface (T1, E1, DS3, or Serial) module. In an EPIM slot, you can use a LAN (10/100/1000, SFP, FE Serial, T1/E1, and DS3) or WAN module. This allows for a generous mix of LAN or WAN interfaces. The SSG 550 is capable of providing a mix of LAN or WAN ports but it might be considered more of a LAN device because of the higher throughput of the unit. Another added bonus of the SSG 550 versus the SSG 520 is the fact the 550 can have dual hot-swappable power supplies. The SSG 550 just as the rest of the SSG products provides the full UTM feature set. The SSG 520 can support up 150 VLAN interfaces and 60 security zones. That is more than double the number supported on the NS-208 firewall. The SSG 550 has two memory options: 256MB and 1GB. The 1GB memory option is required to support the UTM features.

The Integrated Security Gateway (ISG) firewall products are more then just firewalls. They include all of the features of a Juniper Firewall/VPN device. Secondly, the ISG firewalls also allow you to provide full IDP integration as well. The IDP integration is enabled by installation of an additional system board, a security module (SM). The security module offloads the IDP traffic processing from the rest of the firewall. The ISG product line includes the ISG 1000 and ISG 2000 devices (see Table 2.6).

Table 2.6 Juniper Networks' Enterprise-Class ISG Firewalls

	NetScreen-ISG 1000 Baseline	NetScreen-ISG 1000 Advanced	NetScreen-ISG 2000 Baseline	NetScreen-ISG 2000 Advanced
Interfaces	4 fixed CG plus up to 4 Mini-GBIC (SX or LX), or up to 8 10/100/1000, or 20 10/100	4 fixed CG plus up to 4 Mini-GBIC (SX or LX), or up to 8 10/100/1000, or 20 10/100	Up to 8 Mini-GBIC (SX or LX), or up to 8 10/100/1000, or up to 28 10/100	Up to 8 Mini-GBIC (SX or LX), or up to 8 10/100/1000, or up to 28 10/100
Security Modules	0	2	0	3
Maximum Throughput				
Firewall	1000 Mbps	1000 Mbps	2000 Mbps	4000 Mbps
VPN	1000 Mbps	1000 Mbps	1000 Mbps	2000 Mbps
IPS	N/A	1000 Mbps	N/A	2000 Mbps
Maximum Sessions	250,000	250,000	256,000	512,000
With IPS Configuration	N/A	500,000	N/A	1,000,000
Maximum VPN Tunnels	1,000	2,000	1,000	10,000
Maximum Policies	30,000	10,000	30,000	30,000
Virtual Systems	Up to 10	Up to 10	Up to 50	Up to 50
Security Zones	20 up to 20 more with VSYS	20 up to 20 more with VSYS	26	26 up to 100 more with VSYS
Virtual Routers	3 up to 10 more with VSYS	3 up to 10 more with VSYS	3	3 up to 50 more with VSYS
VLANs	50	1000	100	2000
Routing Protocol Support				
RIP v2	Yes	Yes	Yes	Yes
OSPF	No	Yes	No	Yes
BGP	No	Yes	No	Yes
High Availability				
HA Lite	No	No	No	No
Active/Passive	Yes	Yes	Yes	Yes
Active/Active	No	Yes	No	Yes
Active/Active Full Mesh	No	Yes	No	Yes

Continued

Table 2.6 continued Juniper Networks' Enterprise-Class ISG Firewalls

	NetScreen-ISG 1000 Baseline	NetScreen-ISG 1000 Advanced	NetScreen-ISG 2000 Baseline	NetScreen-ISG 2000 Advanced
Antivirus Scanning				
Embedded	No	No	No	No
External	Yes	Yes	Yes	Yes
Deep Inspection	No	Yes	No	Yes
Full IPS/IDP Capability	N/A	Yes	N/A	Yes

The first product in the ISG line is the ISG 1000, providing 2 Gbps firewall throughput, 1 Gbps 3DES VPN performance, and up to 1 Gbps IPS performance. The NetScreen-ISG 1000 has two interface modules that allow you to combine any of the following: four-port 10/100 Ethernet module, eight-port 10/100 Ethernet module, or a dual-port mini-GBIC module. Onboard the ISG 1000 is four 10/100/1000 copper Ethernet ports.

The ISG 1000 allows you to install up to two security modules. This allows you to provide up to 1 Gbps throughput for IPS inspection. The security modules are individual computing boards added to the main location of the chassis. If you use the security modules, you must purchase an IDP upgrade kit. The kit includes a memory upgrade for the management modules, tools, NSM 5 device license, and an IDP license for the ISG. NSM is required to manage an ISG 2000 with security modules. You must purchase support separately with the included NSM 5 support license.

The ISG 1000 does not have a modular power supply configuration; it has one fixed alternating current (AC) power supply. In the advanced license model, the device supports the Active/Passive, Active/Active, and Active/Active Full Mesh high-availability configurations. However, with a baseline license, the device supports only an Active/Passive mode HA configuration. A NetScreen device in high-availability mode requires two dedicated ports to enable both a *heartbeat* and the passing of session synchronization information.

The NetScreen-ISG 2000 provides 4 Gbps firewall throughput, 2 Gbps 3DES VPN performance, and 2 Gbps IPS performance. This is a tremendous amount of throughput for a firewall device. The second important feature is port density. The NetScreen-ISG 2000 has four expansion slots that allow you to combine any of the following: four-port 10/100 Ethernet module, eight-port 10/100 Ethernet module, or a dual-port mini-GBIC module. That means you could have a maximum of 28 Ethernet ports or eight mini-GBIC modules. For a firewall appliance, that is a huge number of total ports. This gives you and plethora of options for this device on your network. As discussed earlier the ISG 2000 allows you to install up to three security modules. This allows you to provide up to 2 Gbps throughput for IPS inspection.

The NetScreen-ISG 2000 includes two hot-swappable AC power supplies to start your device for total redundancy. In the advanced license model, the device supports the Active/Passive, Active/Active, and Active/Active Full Mesh high-availability configurations.

However, with a baseline license, the only device supports an Active/Passive mode HA configuration. A NetScreen device in high-availability mode requires you to have ports dedicated to enable both a heartbeat and the passing of session synchronization information. The NetScreen-500 provides two dedicated ports specifically for this purpose. It can also support up to 50 virtual systems, half a million sessions for the firewall, one million sessions with the IDP modules, and up to 10,000 concurrent VPN tunnels.

Service Provider Class

Juniper Networks' Service Provider class includes the following products:

- NetScreen-5200
- NetScreen-5400

Welcome to the top of the NetScreen firewall product line. These are the true Service Provider class firewall. These firewall devices are some of the highest performing firewalls in the world. With a colossal level of throughput and port density, these devices are exactly what you need for a company that has a massive network infrastructure. The Service Provider Class Line features are listed in Table 2.7.

Table 2.7 Juniper Networks' Service Provider Class of Firewalls

	NetScreen-5200 (Management One Modules)	NetScreen-5400 (Management One Modules)	NetScreen-5200 (Management Two Modules)	NetScreen-5400 (Management Two Modules)
Interfaces	8 Mini-GBIC or 2 Mini-GBIC and 24 10/100 Ethernet	24 Mini-GBIC or 6 Mini-GBIC and 72 10/100 Ethernet	8 Mini-GBIC or 2 10 GigE ports	8 Mini-GBIC or 2 10 GigE ports
Maximum Throughput				
Firewall	4 Gbps	12 Gbps	10 Gbps	30 Gbps
VPN	2 Gbps	6 Gbps	5 Gbps	15 Gbps
Maximum Sessions	1,000,000	1,000,000	1,000,000	1,000,000
Maximum VPN Tunnels	25,000	25,000	25,000	25,000
Maximum Policies	40,000	40,000	40,000	40,000
Virtual Systems	Up to 500	Up to 500	Up to 500	Up to 500
Security Zones	16 up to 1000 additional	16 up to 1000 additional	16 up to 1000 additional	16 up to 1000 additional

Continued

Table 2.7 continued Juniper Networks' Service Provider Class of Firewalls

	NetScreen-5200 (Management One Modules)	NetScreen-5400 (Management One Modules)	NetScreen-5200 (Management Two Modules)	NetScreen-5400 (Management Two Modules)
Virtual Routers	2 up to 500 additional	2 up to 500 additional	2 up to 500 additional	2 up to 500 additional
VLANs	4,000	4,000	4,000	4,000
Routing Protocol Support				
RIP v2	Yes	Yes	Yes	Yes
OSPF	Yes	Yes	Yes	Yes
BGP	Yes	Yes	Yes	Yes
High Availability				
HA Lite	No	No	No	No
Active/Passive	Yes	Yes	Yes	Yes
Active/Active	Yes	Yes	Yes	Yes
Active/Active Full Mesh	Yes	Yes	Yes	Yes
Antivirus Scanning				
Embedded	No	No	No	No
External	Yes	Yes	No	No
Deep Inspection	Yes	Yes	Yes	Yes
Throughput	500 Mbps	500 Mbps		

Both devices are almost identical except for two things: port density and throughput. The core component of the devices is the chassis. The chassis contains two slots in the NS-5200, and four slots in the NS-5400. These slots allow you to add management modules; one is required per chassis, and interface card. There are two versions of modules: M1 and M2. The two models cannot be intermingled.

The first device the NetScreen-5200 series appliance is allowed a maximum of eight mini-GBIC ports, two mini-GBIC ports, and 24 10/100BaseT Ethernet ports, or two 10 GigE ports. It has a maximum throughput of 10Gbps firewall inspection. For VPN performance, it provides 5 Gbps 3DES throughput. The other enterprise-class device, the NetScreen-5400, has even more impressive performance and port density. This device can have either a maximum of 24 mini-GBIC ports, six mini-GBIC ports, and 72 10/100BaseT Ethernet ports, or six 10 GigE ports. Without little explanation required, you can appreciate the astounding statistics on these two devices. The NS-5400 can provide up to a 30Gbps throughput firewall and 15Gbps VPN.

For the most part, these two appliances have identical overall performance statistics. The NetScreen-5000 product line can support up to 1,000,000 concurrent sessions. In addition,

they can support up to 25,000 VPN tunnels, 500 virtual systems, and up to 4,000 virtual LANs (VLANs). Both devices can support all three modes of high-availability: Active/Passive, Active/Active, and Active/Active Full Mesh. Both devices support the HA ports to provide both heartbeat and session synchronization.

Enterprise Management

Using Juniper Networks' firewalls in your enterprise provides you with these benefits:

- A unified management interface
- Lower administrative costs
- Centralized logging
- Simplified VPN deployment

Juniper offers you two easy-to-use methods for managing a firewall. You can operate the easy-to-use WebUI, or you can use the command-line interface to control your Juniper firewall. This is a great way to administer multiple devices. However, what if you need to manage 10, 100, or maybe even 1,000 devices? Managing each individual firewall turns into a giant chore. Merely accounting for the logging from multiple devices can be daunting. Is it practical to use a simple syslog server to manage all devices. This points to the need for a centralized management console. Enter the NetScreen-Security manager (NSM). This product is an all-in-one solution to manage up to 1,000 NetScreen firewall appliances concurrently. The NetScreen-Security manager is *the* solution to control multiple devices.

Each individual device is entered into the NSM. Once the device has been imported, you can manage each individual aspect of the firewall directly from the NSM. You can add and delete security zones, create new policies, and tweak existing policies. If you have dozens of locations requiring the same policy, you can easily deploy that policy to all of devices at once. If you need to make a change to that policy, rather than accessing each device individually, you can change the policy once, and update all the other device policies. This simplifies large-scale deployments, and allows the administrator to gain control over the unwieldy enterprise security as a whole. The NSM incorporates logging to one central location and stores it for historical purposes, as well as providing real-time monitoring capability. Sorting through log information can be hassle. Juniper Networks provides a quick reporting system to summarize the priority information you need, when you need it. This helps to quickly identify network areas requiring your focused attention.

Consolidating all devices into a tightly knit VPN solution can be complicated when you have multiple devices. Verifying that each device it is properly configured can lead to big headaches—especially if you need to make configuration changes. However, the NetScreen-Security Manager makes deploying large-scale VPNs a snap. You simply define all protected resources for the level of access that you desire. Then configure your VPN topology, designating which are hubs and which are spokes. Then deploy that configuration to all of your devices at once, and your VPN is up and running. You can monitor your enterprise's VPNs

using the VPN monitor built in to NSM. From one screen, you can determine which locations have their VPNs up and which are connected to each other. This takes the guesswork out determining what is happening within your secured infrastructure.

There are a number of reasons to use NSM to take control of your NetScreen infrastructure. If you have all of your devices deployed, but you have enterprise management issues to address, this is an easy task for NSM. Simply import all existing configurations into the NSM, and then you can begin to use the helpful features of NSM. All policies, address objects, and VPNs are imported directly into the NSM. This allows you to retain these configurations. Secondly, if you are performing a new deployment, you can simply preconfigure your devices to contact the NSM for details of configuration. Once the device is online, and can contact the NSM, you can conduct all management from the NSM. Finally, if you are using the legacy Global Pro product, your configuration can be easily imported into NSM. This allows you to take advantage of the newer technologies of the NSM product. For any NetScreen deployment, small or large, the NSM can easily empower the administrator into gaining full control over your network.

In its latest release the NSM server can manage two new types of device classes. The inclusion of IDP management is one device type. It allows you to utilize the NSM architecture to manage your IDP deployment. The management interface is streamlined to allow for this new device, and it offers many benefits over the older IDP software manager. Secondly, you can manage the integrated ISG/IDP devices. This allows you to create a firewall policy, and an IDP policy for the device. It utilizes a simple workflow to ease the management of complex solutions.

Summary

In Chapter 2, we explored the various components that a Juniper firewall comprises. Juniper consistently brings to market a secure, speedy, and cost effective solution. In the past two years, Juniper has rounded out its firewall product line to fit many of the needs you, and your organization, require in other areas of security: such as remote access, Intrusion Prevention, and Unified Access Control.

The NetScreen security product line contains an amazing collection of security products. The three core product lines offer the enterprise customer a good selection of products for their networks. The firewall product line offers a core set of products to secure network focal points. To minimize your network risks, the IDP product provides the capability to perform a detailed inspection of your traffic. With the proper configuration, you can block malicious traffic before it has a chance to affect your systems, or to compromise them to create a loss of data. The Secure Access SSL VPN product is a new solution to an old problem. Remote access to the company's network has been a long, tedious, and strife-filled journey to provide an easy-to-deploy, yet secure, solution. The NetScreen SSL VPN solution can deploy to thousands of users without the actual deployment of software. This helps organizations because it does not require a large staff to manage the software. These security products provide secure options for any company's size.

We explored the core technologies that make up the Juniper firewall product line. Zones are a core part of the NetScreen firewall. Zones allow the administrator to divide networks into logical divisions. This allows you to simplify the policy creation process by clearly enabling, or denying, access to the various network segments based on zones. Juniper truly bends the idea of a firewall with the incorporation of virtual routers. Virtual routers allow you to separate routing domains into separate logical entities. This allows a firewall to utilize the firewall as a true router, without compromising security. The Juniper firewall product again bends the traditional look of a firewall by acting as a transparent device on your network, yet it continues to provide the full spectrum of firewall features. Policies in the NetScreen firewall are the rule base, security policy, or access list of competitive products.

In addition to serving as a firewall gateway, the Juniper firewall is also a fully integrated VPN gateway, which provides the capability to act as a site-to-site gateway. It also provides remote VPN access to mobile users. The industry standard IPSec implementation provided by NetScreen gives the enterprise a truly enterprise-class VPN solution. Application-level security is necessary for each organization. It provides inspection capability of the application layer that, otherwise, could be provided only by a dedicated device such as the IDP product. The intriguing design of the hardware architecture proves that the single-purpose design can provide a high-end, high-performance firewall device.

The Juniper firewall product line provides a complete selection of firewall products that cover all network security needs. Each product is tailored to provide exactly what you need for a solution to enterprise firewall needs. The NSM product brings all firewalls together for centralized management solutions.

Solutions Fast Track

The Juniper Product Offerings

- ☑ The NetScreen firewall products have both the ICSA and Common Criteria certification.
- ☑ Kaspersky antivirus is used for virus scanning on the firewall product line.
- ☑ The secure access SSL VPN is a clientless solution that does not require the predeployment of software.
- ☑ The Secure Access SSL VPN product can be deployed to thousands of users in a matter of hours.
- ☑ The IDP product allows you to inspect traffic for malicious intruders.
- ☑ The IDP deploys nine mechanisms to detect attacks.

The Juniper Firewall Core Technologies

- ☑ Zones separate logical areas inside the firewall.
- ☑ Virtual routers allow multiple routing tables to exist in a single device.
- ☑ Juniper firewall, in transparent mode, allows the firewall to act as a switch while still providing its usual firewall functions.
- ☑ A policy is used to allow or deny traffic to pass through the firewall gateway.
- ☑ Juniper firewalls are integrated VPN devices.
- ☑ You can use a Juniper firewall in both site-to-site VPN configurations as well as client-to-site configurations.
- ☑ There are two different ways to create a VPN in a Juniper firewall, either route-based or policy-based.
- ☑ Deep inspection allows you to look inside of a packet for a malicious code.
- ☑ The Juniper firewall is based on ASICs in order to increase its performance.
- ☑ The Juniper VPN clients are supported only on Microsoft Windows operating systems.

The NetScreen and SSG Firewall Product Line

- ☑ The NetScreen-5-GT products contain an internal antivirus scanning engine.
- ☑ The NetScreen-25 and NetScreen-50 products are perfect solutions for small- to medium-size businesses.
- ☑ Both of the NetScreen-204 and NetScreen-208 products are good solutions for larger organizations.
- ☑ The entire SSG Product line supports the capability to perform multiple functions: antivirus, antispam, IPS/DI, and Web filtering
- ☑ The ISG product line supports the capability to function as a standalone firewall, and an integrated intrusion prevention solution
- ☑ The NetScreen-5400 is the highest performing NetScreen firewall providing 12 Gbps of firewall throughput.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: You mention several times that the NetScreen firewall is ICSA certified. Why does this matter?

A: The ICSA certification ensures that the firewall device meets a certain level of criteria. This is important when determining interoperability between various vendor devices. For example, automotive companies use a special network, the automotive network exchange (ANX). You are required to use an ICSA-certified device to ensure that your device will be interoperable with other trading partners on that network.

Q: Security zones are a confusing concept. Other vendors get along with out them, so why use them at all?

A: Zones are excellent tools to provide logical separation between multiple areas of your network. As you will see in later chapters that delve into creating policies, zones simplify the process by identifying the two separate areas of your network that you want to enable access to. This can prevent you from accidentally creating access rules that allow unintended access to specific sections of your network.

Q: Deep inspection seems like a great technology, but you seem to have a negative opinion about it. Why?

A: I am a firm believer in the deep inspection technology. It truly is the next step in the evolution of firewall devices. However, technologies like deep inspection should be used as a supplement rather than as a single solution for application-level security. Using deep inspection alone is a great solution for many companies, but it is not recommended as the only solution for a large e-commerce infrastructure. Security is best served as a layered model.

Q: Is it true that the SSG firewalls do not contain an ASIC based architecture?

A: The SSG product line is based upon the architecture of the very successful 5GT product line. An ASIC is not capable of providing all of the advanced features such as deep inspection and antivirus at rapid rates. Therefore, by utilizing a faster processing architecture we can provide these features at faster rates to our customers. The architecture is custom built to enable the fastest possible packet processing.

Q: Why would Juniper limit the number of policies that each device can have?

A: Each NetScreen device is designed to perform at a specified rate of performance. Each NetScreen device could probably support a greater number of policies, but the greater number of policies could degrade its performance. For each policy in the list, the NetScreen firewall checks from a top-down perspective. Therefore, the longer the list of policies, the more time it takes to traverse the line.

Chapter 3

Deploying Juniper Firewalls

Solutions in this chapter:

- **Managing Your Juniper Firewall**
- **Configuring Your Firewall for the First Time**
- **Configuring System Services**

- ☑ **Summary**
- ☑ **Solutions Fast Track**
- ☑ **Frequently Asked Questions**

Introduction

In this chapter we will look at the basics of deploying a Juniper firewall. The Juniper firewall has a large number of configuration options. Before you can deploy a device, you must first understand how to manage it, so in the first section of this chapter we look at the various methods of managing your Juniper firewall. Each option and best known procedure is discussed. Strong system security is important, but no more so than preventing intruder attacks.

There are many management options available on the Juniper firewall. Of these options, there are, effectively, two ways to manage the device directly. The first is from the *command line interface* (CLI). Many people still prefer this method of device management. Fully comprehending the command line interface allows you to better understand the Juniper firewall. There are specific functions that can only be done from the command line interface. Many of these commands are not commonly used, but are switches to enable or disable specific system features.

The second firewall management option is the *Web User Interface* (WebUI). This streamlined interface is user friendly and intuitive, allowing anyone to jump in and manage the firewall with ease. Even command line junkies will use the WebUI to reference the configuration, or to see a configuration more clearly.

Since a firewall is a core component of the network, we will focus heavily on how to configure your device to interact with the network. This covers *zone configuration* and Internet protocol *(IP) address assignment*. Properly configuring the network is crucial to the functionality of your network entity. Each type of zone and interface is documented to explain the available configuration options. Finally, we will configure various system services available from your Juniper firewall.

Managing Your Juniper Firewall

The first step in learning about firewalls is how to effectively manage them. In this section, we will look at the various management configuration options. The core configuration component for the firewall is the CLI. Even if you are using the WebUI it still ultimately generates the CLI configuration for you. While not required to memorize the CLI, it will greatly help if you do.

When managing your firewall you are required to authenticate to the device. Securing your management access is key to your network security. If you lose control of your access points, you lose control to your network. Creating a strong authentication policy for your administrators is essential for the effectiveness of your firewalls.

There may be times when you mistakenly erase parts of your configuration, or lose your configuration altogether. We will review how to recover from this type of mistake. Losing access to your device can be devastating. With so many different passwords to remember, you can easily forget how to gain access to your Juniper firewall. Even the most experienced administrators can find themselves in this predicament. However, several methods of recovery have been documented.

Finally, we will look at how to update the operating system on your Juniper device. Staying current with software revisions is very important. It provides you with security-related fixes as well as new software enhancements. For each type of management option, there is a specific way to update ScreenOS. Some options may be more effective then others, depending on your needs. At the completion of this section you should be familiar with WebUI and CLI. Knowing this is a requirement for managing your Juniper firewall.

Juniper Management Options

Every Juniper management option centers around two forms of management: the WebUI and the CLI. There is a third type of management, an enterprise class of security, called the NetScreen Security Manager (NSM). Because NSM's configuration options are extensive, NSM is outside of the scope of this book.

Serial Console

The *Serial Console* is a nine-pin female serial connection. This option gives you CLI access to the firewall. Serial Console is used to initially connect to your device, and to conduct *out-of-band management*. Out-of-band management is management that is not network based, such as access via modem. There are certain benefits to using a serial console that you do not get from using any other type of connection. The console provides a secure, physical, and dedicated access. Network connectivity issues cannot interrupt this type of connection, and no one can intercept your management traffic. It is completely secure because of its direct connection.

When configuring over a serial port, you are not using any type of network connectivity. In the case when you need to change Internet Protocol (IP) addressing on the firewall, and guarantee connectivity, using the serial console is an excellent option. With, and only with, serial console can you view and interact with the booting process. This cannot be accomplished remotely because the operating system (OS) has not started, and it is unable to provide management services. Many devices from UNIX servers, as well as other embedded devices, use serial consoles to provide serial console management. Most of the devices use an RJ-45 serial cable with a DB9 female connector. However some older devices use a DB9 female to DB9 male straight through serial cable. Table 3.1 outlines the proper connection settings when connecting with a serial terminal, or serial terminal emulator.

Table 3.1 The Serial Terminal Settings

Setting	Value
Speed	9600 bps
Character Size	8 Bit
Parity	None
Stop Bit	1
Flow Control	None

Telnet

A second form of CLI management is *Telnet*. Telnet is a protocol that has been used for years, and it is like a network based version of a serial console. However, it lacks many of the advantages of a serial console. First of all, it is a very unstable connection. The connection is made over the network in clear text format. This means that the transmitted data is not encrypted in any way, thereby allowing easy access to your login and password. Most client operating systems provide an easy to use Telnet client. A Telnet connection is not an ideal configuration for managing your device from a remote location. You can have a maximum of two active concurrent Telnet sessions. Most operating systems come with a built-in Telnet client. If not, you can use a program called *Tera Term*. Its download location can be found in the Resources section at the end of this chapter.

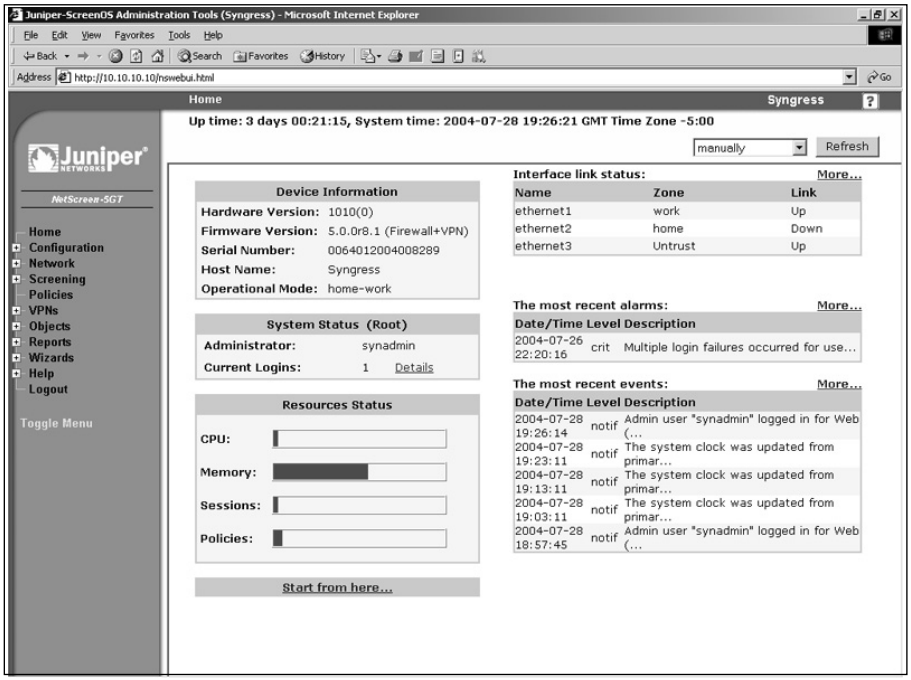
Secure Shell

The third form of command line management is *secure shell* (SSH). Like Telnet, SSH is a remote command line session. When using SSH, Telnet's security concerns are not an issue. Secure Shell provides an encrypted command line session to the Juniper firewall. It also provides protection from IP spoofing, and Domain Name System (DNS) spoofing attacks. SSH has two versions, v1 and v2. The versions are not backwardly compatible. Version two is more popular because of its higher level of security. You are required to have a client that is compatible with the version of SSH that you are using. Many UNIX based operating systems include clients, but Windows based operating systems do not. You can use a client named *PuTTY* for Windows. It is free, and it is easy to use. Information on the PuTTY client can be found in the Resources section at the end of this chapter.

WebUI

The Web user interface is the easiest type of management to use. Because of its simple point-and-select nature, it gives the end user a jumpstart into the management of the Juniper firewall. You can see in Figure 3.1 that the interface is very straightforward. On the left-hand side of the browser is the menu column. From here you can choose from the various configuration options. This menu can be either Dynamic Hypertext Markup Language (DHTML) based, the default, or Java based. The functionality is the same, but the look and feel is slightly different. By default, the WebUI is configured to work over only the Hypertext Transfer Protocol (HTTP). It can, however, be configured to work over Hypertext Transfer Protocol Secure (HTTPS). This provides a mechanism to secure your Web management traffic. Most of the popular Web browsers such as Internet Explorer, or Firefox work well with it.

Figure 3.1 Web User Interface



The NetScreen Security Manager

The NetScreen Security Manager (NSM) is a separate tool that can be used to manage a Juniper firewall device. The NSM is an application that runs on either a Solaris server, or a Red Hat Linux server. It requires a separate license, and it is licensed based on how many devices you want to manage. This product is used most effectively when you need to manage several devices at the same time. It uses an object-oriented management design.

Administrative Users

When connecting to a Juniper firewall for management purposes, you must always authenticate to the firewall. There are several types of users that you can employ to connect a Juniper firewall. The first user is the *root user*. This user is the principal user of the Juniper firewall device. The root user has the most power of any user on a Juniper firewall. There is only one root user per device. By default, the root user's name is *netscreen* and the default password is *netscreen*. It is highly recommended that you immediately change the login name and password. The root user has the greatest number of administrative privileges of any device. The *root user administrative privileges* are listed below:

- Add, remove, and manage all other administrators
- Create and manage virtual systems
- Create, delete, and manage virtual routers
- Add, delete, and manage security zones
- Assign security zones to interfaces
- Perform asset recovery
- Set the device to Federal Information Processing Standards (FIPS) mode
- Reset the device to default settings
- Manage the device's firmware
- Load configuration files
- Perform management on the root system

The next level of administrator is *read/write*. Read/write is very similar to the root user; however, read/write users cannot create other administrators. This type of access is most useful when you want to distribute administrative privileges to others, yet control access. The Juniper firewall provides a very detailed audit log of the actions of each administrator. You should capitalize on this by creating administrative users for each person who administers your firewall. This way you can identify the user with the modification. There is no reason to share an administrator user account between two users. The read/write administrative privileges include:

- Create and manage virtual systems
- Create, delete, and manage virtual routers
- Add, delete, and manage security zones
- Assign security zones to interfaces
- Perform asset recovery
- Set the device to FIPS mode
- Reset the device to default settings
- Manage the device's firmware
- Load configuration files
- Perform management on the root system

The next type of user is the *read-only* administrator. This user has limited access to the system. As the name suggests, the user can only view the configuration, and they are unable to modify the system in any way. This is useful if you want to assign a technical writer to document your configurations, or if you want to give anyone limited access to the device to

perform troubleshooting on the network. The following list includes the limited privileges of the read-only administrator.

- Read-only privileges in the root system
- Read-only privileges in all virtual systems

On some devices you can have *virtual systems*. A virtual system acts as its own separate security domain. Virtual system administrators have permission only on a specific system. The virtual system administrator privileges are shown in the following list.

- Create and manage auth, Internet Key Exchange (IKE), Layer 2 tunneling protocol (L2TP), Extended Authentication (Xauth), and Manual Key users
- Create and manage services
- Create and manage policies
- Create and manage addresses
- Create and manage virtual private networks (VPNs)
- Modify the virtual system administrator login password
- Create and remove virtual system read-only administrators

The last type of user is the *virtual system read-only administrator* who has almost the same privileges as a read-only administrator. The difference is that they can see only the configuration of a single, specified virtual system.

Becoming familiar with the privileges associated with the different types of administrator can give you the tools to create an efficient strategy for delegating authority on your system. Do not be afraid to create many different administrative users for your Juniper device. This will provide you with granular access to your system. Again, all users' actions are logged. This log provides a detailed list of access for each user. This can be helpful when determining issues related to a particular administrator, or in determining whether or not an administrator account has been compromised. Chapter 6 reviews the use of external authentication sources for administrative users. This can provide additional security in cases where you use technologies such as *SecurID* to remove the use of a single static password.

The Local File System and the Configuration File

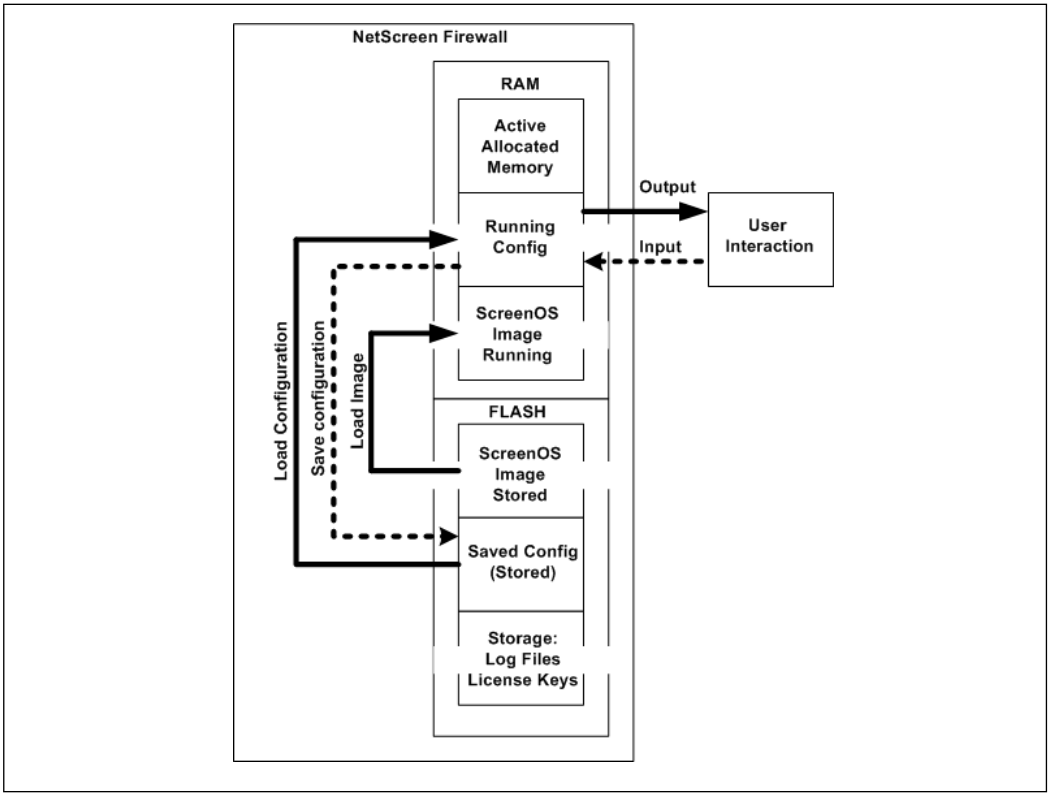
Each Juniper firewall device has a similar design for its internal system components. Long-term storage on the device is stored into *flash memory*. Flash memory is a non-volatile memory that retains information after the system is turned off. Some devices have a Compact Flash (CF), Secure Digital Memory (SD) card slot, or a universal serial bus (USB) port for external storage. This is flash memory, but it is removable. The internal flash is not

removable. All component information that Juniper needs to store is in flash memory, including ScreenOS log files, license keys, attack databases, and virus definitions.

Each Juniper device also contains random access memory (RAM). This is a volatile type of memory that is cleared whenever the system is powered off, or reset. When the Juniper device powers on, and after the power on self test (POST) is completed, the ScreenOS image is loaded into RAM. After ScreenOS is up and functional, it loads the saved configuration file from flash memory. The configuration that is stored in RAM is called the *running configuration*.

Whenever you make a change to the configuration, it is always saved to the running configuration. If you make changes to your configuration but fail to save it, the file would revert to the last saved configuration whenever you reset or rebooted your device. When you remove power to the device, and then restore power, it causes a return to previously saved configuration. When using the CLI, your configuration must be *manually* saved. This can be done by using the *save* command. The save command is simply **save**. By typing that command, your running configuration is saved as the *saved configuration*, which is stored in flash memory. The file system components are shown in Figure 3.2.

Figure 3.2 File System Components



Using the WebUI is even easier. The WebUI *automatically* saves your configuration after every change. However, when using the CLI, if you exit your session or attempt to reset the device, you will be notified that your configuration has changed. At that point you are given the option of saving the configuration. The Juniper device is much more user friendly than other devices when it comes to advising you that your configuration has changed, and offering you the option to save it.

There are times when flash may not provide you with the type of storage that you need. You may require long term storage of log files, or perhaps a backup of your configuration file. There are two ways to accomplish this:

- When using the command line, you can apply the command **get config** to view your configuration, then copy and paste it into a simple text document.
- From the command line, you can copy the configuration to a Trivial File Transfer Protocol (TFTP) server. TFTP is a simple type of File Transfer Protocol (FTP) server. It requires no authentication, but only specification of the filename you are placing on the server. To save your configuration to a TFTP server, use the command **save config to tftp <*a.b.c.d*> <*file*>,** where <*a.b.c.d*> is the IP address of the TFTP server, and <*file*> is the filename you want use for the save.

Depending on the data that is being transferred from the file system, you may prefer a more secure option than TFTP. You can use *secure copy* (SCP) to transfer files as well. Secure copy is similar to secure shell. It requires a special client in order to interact with it. Many UNIX systems include this feature. Windows has many clients. I prefer the **PuTTY Secure Copy (**PSCP) software, which is part of the *PuTTY* freeware secure shell clients. In the following example we will turn on SCP, and copy a file from the Juniper firewall to our UNIX system.

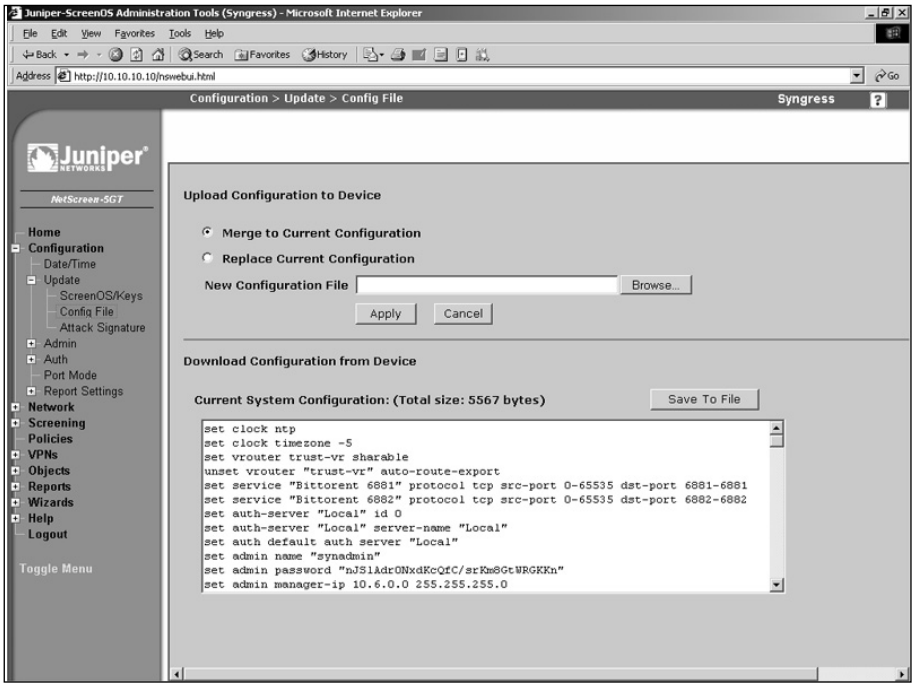
```
From the CLI:
Syngress-> set scp enable
Syngress-> get scp
SCP is enabled
SCP is ready
Syngress-> get file
    flash:/envar.rec                    98
    flash:/golerd.rec                 1220
    flash:/burnin_log1               10240
    flash:/burnin_log0               10240
    flash:/dhcpservl.txt                52
    flash:/ns_sys_config              1092
    flash:/dnstb.rec                     1
    flash:/license.key                 395
    flash:/$lkg$.cfg                   922
    flash:/expire.rec                   23
```

```
    flash:/attacks.sig                 198833
Syngress->
From the UNIX Host:
UNIX-Host:~ syngress$ scp synadmin@10.6.0.1:license.key license.txt
The authenticity of host '10.6.0.1 (10.6.0.1)' can't be established.
DSA key fingerprint is f9:a7:4c:53:4c:0a:cc:5a:50:6b:eb:df:42:42:63:c0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.6.0.1' (DSA) to the list of known hosts.
synadmin@10.6.0.1's password:
license.key                                   100%  395     4.8KB/s   00:00
UNIX-Host:~ syngress$ cat license.txt
1k=d2f5fb8aa5b9a000&n=capacity_key
k=2JQcSPh1ogana6h82NJeAfDwgb3aiOXT2UFcm9OFQDkuK4iT6YfKefMZjTODboIN2JQ0oWnWWX+nKkY
SMytB8gF1ID7tWXI9lvZ11JURDENckexZ7IwtmRmDEh+YT3dJvDSOAYeGuuWFtGYE5tVnPfZq6cnlO254
GPPm5HJ3qTG4sRBSRR/QFqL6WAnfnoSpByJu/Xr9vxx9GSU4fTMGLFkWsbRP5cVpTGWmyOBapFfn1qWzu
/bMLzDkox8zUHFZ2NcNCOSGOk5PvCMcZwOaADRIFqJj1oh4u7+toY37gdrEM5sQqmELemAlUi90dhLPl7
jsTy1R/V0/ourYn00XcMw==&n=di_db_key
UNIX-Host:~ Syngress$
```

As you can see, we enabled SCP, allowing us to view all of the files stored in flash memory. Next we went over to the UNIX host and copied the file from the Juniper device to the local UNIX system. Finally, we used the **cat** command to concatenate the contents of the file so you can see them. SCP can be effective and easy to use for removing files from Juniper devices.

If you are using WebUI, you can access **Configuration** | **Update** | **Config File** and then select the button labeled **Save To File**. This will allow you to save the configuration to your local PC as shown in Figure 3.3. Alternatively, from this same screen you can select the text in the text window, then copy and paste the configuration to a text file. As you have seen from these files, the config files are a collection of commands. The configuration file operates similar to manually typing these commands in line by line. This is great because it requires that you understand only one format. It also allows you to easily modify saved configuration files to reflect changes. Becoming familiar and comfortable with the use of the CLI cannot be stressed enough. In the next section, we will examine the configuration of the device, and the commands available to administer the device.

Figure 3.3 WebUI Save Screen

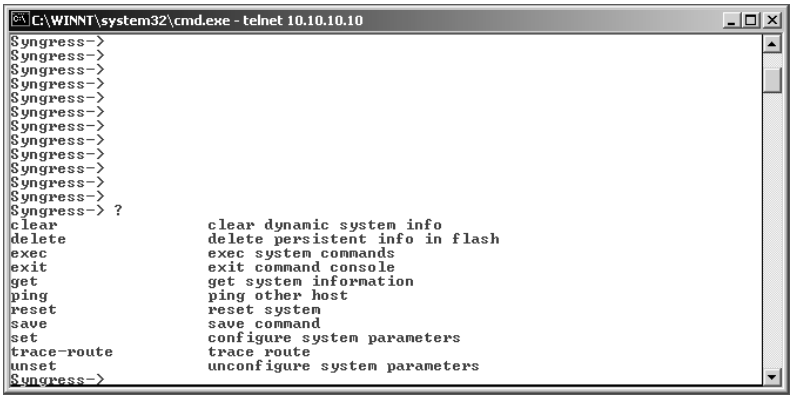


Using the Command Line Interface

The command line interface is at the core of configuring your Juniper firewall device. No matter which method you use to manage your firewall, the CLI commands control the device, and a thorough understanding the CLI is crucial to effective management. The NSM generates the same commands that you may manually enter via the CLI. CLI commands are straightforward, and easy to learn. Other devices use cryptic commands, or commands that seem to do one thing, but actually perform an unrelated action. When this firewall was designed, the engineers took the need for simplicity into consideration. In Figure 3.4, an example of the help screen is shown. This gives you an idea of the information provided by the *Help* command.

Figure 3.4 shows an example of the command line. The prompt shows the device's current *host* name. This is very useful if you have several devices that are not readily distinguishable from the command line. Starting at the root, there are literally thousands of command options. Memorizing this great number of commands could be a daunting task. However, there is an easy-to-use built-in help system. From anywhere on the command line, simply type **?** to access the Help system, which will list most available commands. Some are not listed; however, these specific commands will be discussed in later sections.

Figure 3.4 Command Line Session Using Help



From here there are several *base commands*, including *clear, exec, exit, get, ping, reset, save, set, trace-route,* and *unset***.** Under each one of these commands are subcommands.

An example is in order. We will explore the command used to retrieve information from the device, the **get** command. If we wanted to look at system information device such as *uptime*, *serial number*, and *configuration information*, we would use the **get system** command. At the end of any get command you can do one of three things.

- You can press **Enter** and have the information displayed in your terminal window.
- You can redirect the output to a TFTP server much as we did earlier when we saved the configuration. You would use this command **get system > tftp *<a.b.c.d>* *<string>*** to send the output to a TFTP server, where <*a.b.c.d*> is the IP address of the TFTP server, and <*string*> is the filename you want to save.
- You can also use the pipe (**|**) to match output. If you were to use the *get system* command to search for the serial number of your device, you would use the command: **get system | include "Serial Num"**. This would then display only the serial number, and omit the rest of the data. You can also exclude specific information. You would use the same procedure as described earlier, but substitute the term **exclude** for **include**. This helps filter the information provided from a get command.

The next command we will examine is the *set* command. This command is used to set a configuration in the current running configuration. Suppose you wanted to set the hostname of your Juniper device to *Syngress*. You would use the *set hostname Syngress* command to cause your prompt to appear as **Syngress->**. This prompt appears only in the *running* configuration. If you want to ensure that this is the default prompt for your device, simply save the configuration: use the command *save* to commit the running configuration to the saved configuration. The *set* command is used throughout this book; therefore, there will be ample exposure.

It is important that you familiarize yourself with the five, system-controlling commands: *save, exec, exit, delete,* and *reset*. Each of these commands performs a system task. The *save* command can be used to perform functions other than the obvious. The *save* command is used to save files to, and from, the local system. The *reset* command is used to *reboot* the Juniper device. There are several suboptions that allow you to reboot without being prompted to confirm the configuration. You can also force a reboot with a choice of saving the running configuration, or discarding it. This way, when you want to reboot the system you do not have to answer prompts before the reboot. This is helpful if placed inside a configuration script.

The *exec* command is powerful and multi-purposed. The *exec* command runs a command on the system. For example, the command **exec save software from flash to tftp 1.2.3.4 CurrentOS.bin** would save the current version of ScreenOS to a TFTP server. So it would be much like copying a file in DOS or UNIX shell from one location to another. This is an example of the type of function that the *exec* command can provide.

The *delete* command allows you to manage your local system by deleting several types of stored information. This can range from you local stored SSH information to files on the local flash file system. For example, if you wanted to delete a file named *old_data* that was stored in flash memory, you would use the following command: **delete file flash:old_data**. This would delete that file permanently from flash memory.

The *exit* command serves one purpose: to exit your current session. When you use this command, your current CLI session is terminated. If you have made unsaved configuration changes, you will be prompted to save them before you exit.

The *clear* command allows you to clear current data from memory. This can include dozens of options anywhere from the current local DNS cache to the current sessions passing though the firewall. This is useful if you want to remote this information, and to then to accumulate it again. Sessions are a perfect example of something that you may want to clear. You would want to clear you session table if you were troubleshooting a connectivity problem, and you wanted to see the session recreated in your debugging logs. This is as easy as typing **clear session** at the command line, and pressing **Enter** to clear all sessions. You could also selectively delete your sessions depending on your needs.

There are two commands that you can use to for troubleshooting purposes, *ping* and *trace-route*. Though you may have used these before on other operating systems, *ping* is a tool to test *connectivity* between two systems. You use *ping* to verify that your firewall can see a specific host. The *ping* command can be used with options other than host. You can also specify how many ping packets you want to send, as well as the size and the timeout for each packet. To use the *ping* command, just type *ping*, and then the hostname or IP address of the device you want to contact. The other command is *trace-route*. *Trace-route* is similar to *ping*, but it is designed to determine the IP addresses of all routers in the path from your network to the specified remote host.

When using the command line, there are a few special commands that you can use to make things easier for the end user. We previously covered the *?* command for getting help. This can be used for every subcommand, as well as partial commands, to list available options

for that command. The *help* command is very useful, and it should be used often. Next is the **Tab** key, which is used to provide command completion. For example, you can type **set add,** and then press **Tab** to have the command completed for you. This results in the command *set address*. If there is more then one match to the command, both matches will be listed, and you can select the appropriate one. You must continue to type the individual characters of the command until it becomes a unique entity in order for command completion to work. This is universal for the CLI on the Juniper device. This is the same functionality provided by the UNIX *bash shell*. Table 3.2 displays other special key combinations.

Tools & Traps…

Command Line Interface Quandaries

When you use the command line there are occasions where some functions do not appear to be functioning, or where some commands do not seem to cause the expected action. For example, sometimes **Tab** completion will not work. Though frustrating, luckily there are only a few situations in which this can happen. One such situation is when you attempt to use **Tab** completion with the name of an interface. Each time you press the **Tab** key, you see the same line again and again. You can use the question mark to bring up the interface list.

The other situation occurs when you use **Tab** completion to complete the name of a zone. You will get the same results as with interface completion. The command line allows use of truncated commands rather your having to type the complete command name.

For example, rather than typing the command *get interface ethernet3* you could use the command *g int e3*. For the first command we type only the letter g. The first command that it matches with the g is get. Since no other command matches it, ScreenOS interprets the g as the get command. The second command we typed was *int*, and the third was *e3*, which corresponded to ethernet3. The more you use the command line, the more familiar you will become with the short, or truncated, version of the commands.

As you can see, each command is separated by a space. However, if a space between two command line entries is *required*, you simply surround the space/text with quotes. For example, the command *set snmp location Dearborn, MI* would fail. However, if we used the command *set snmp location "Dearborn, MI"*, the text enclosed in double quotation marks would count as a single word.

Table 3.2 Special Key Combinations for the CLI

Special Key	Action
Up-arrow key	Recalls previous command
Down-arrow key	Recalls next command
Control+A	Brings cursor to beginning of the current line
Control+E	Brings the cursor to the end of the current line
Ctrl+C	This is the escape sequence
Left-arrow key	Move cursor back one position
Right-arrow key	Move cursor forward one position
Tab	Completes partially typed command
Question mark (?)	Displays Help and command options

The command line interface environment offers *you* the capability to tailor commands specifically for your purposes. In fact, the more advanced options, such as *debugging*, can only be carried out from the CLI. Administrators generally find the WebUI easier to use at first; however, they soon realize the power of the CLI.

Using the Web User Interface

The Web User Interface (WebUI) is a simple to use tool for managing your Juniper firewall. It is intuitive, and it allows those with little firewall experience to easily control a Juniper device. Figure 3.1 shows the main WebUI page following authentication. The menu bar on the left is where you select configuration options. The current status is displayed on the right-hand side of the screen. On this screen, there are six different boxes: *Device Information, System Status, Resource Status, Interface Link Status, The most recent alarms, and The most recent events.*

Each box reports the status of current events. Current uptime, and the current system time are displayed at the top of the screen. The *Device Information* box shows information such as the hardware version, current firmware version, serial number, host name, and its current operations mode. The *System Status* box performs as its name suggests. It shows the current number of logins to the device, and it shows the login identities. The *Resources Status* displays in a bar graph format, four device resources: CPU, memory, sessions, and policies. If you hover the mouse pointer over any of the bars in the graph, it will display the numerical values for that bar. These are the core performance metrics of the Juniper device. As we discussed earlier, the memory bar graph will read higher then you would expect it to do, because ScreenOS preallocates memory for performance.

If you look at the box entitled *Interface Link Status*, you will see the status of all interfaces. This is handy for determining which interface is up, and which is down. *The most recent alarms* list performs as its name suggests. Finally, as its name implies, *The most recent events* box lists the most recent events. Some boxes in the upper right-hand corner have more hyperlinks, which takes you directly to the detail page for each item.

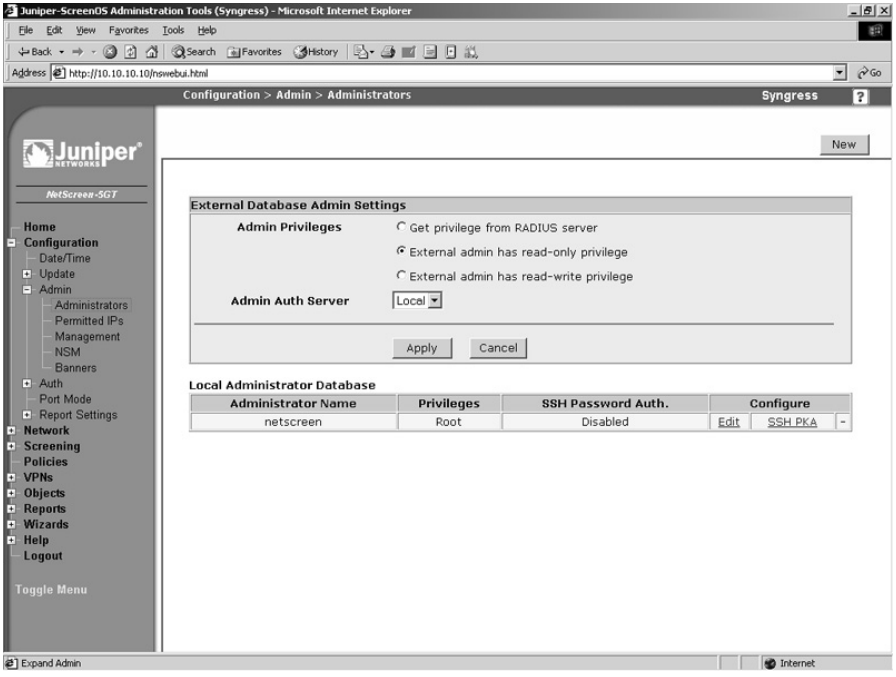
Securing the Management Interface

Now that you understand management of the Juniper firewall device, it is time to *secure* management access to your device. The last thing you want to do is leave the door wide open for an intruder to control your device. There are some easy steps that you can take to prevent this. First, you should change the *root username* and *password*. Everyone who owns a Juniper firewall is aware of the default login and password to the device.

Use the following steps to change the root username and password via the WebUI.

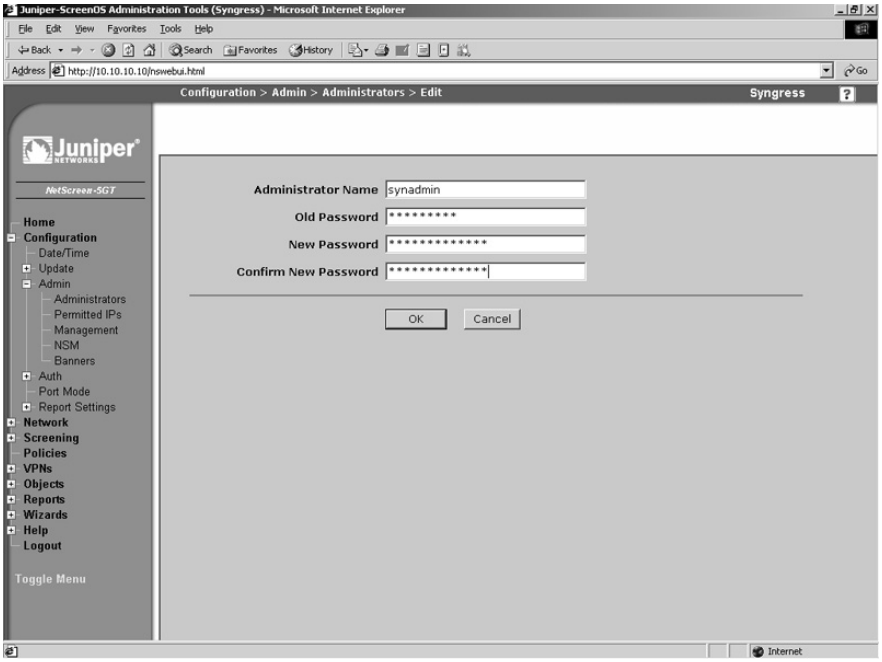
1. Select **Configuration** | **Admin** | **Administrators**. A screen similar to Figure 3.5 will be displayed.

Figure 3.5 WebUI Administrators Screen



2. Press the **Edit** link for the user with *root* privileges. In our example, the *root* user is the only username entry. A screen similar to that in Figure 3.6 will be displayed. Figure 3.6 is identical to Figure 3.5, with the exception that Figure 3.6 must be replaced with a screenshot of the Edit screen.

Figure 3.6 Edit Administrator



3. Change the **Administrator Name** from **Juniper** to **synadmin**.
4. Enter **Juniper** in the **Old Password** field.
5. Enter the new password in the **New Password** and **Confirm New Password** fields.
6. Press **OK**

Use the following steps to change the root username and password via the CLI:

1. Enter the following command to change the admin name:

```
Syngress-> set admin name synadmin
```

   You will see the following message:

```
Password has been restored to default "Juniper". For security reasons,
please change password immediately.
```

2. Enter the following command to change the password:

```
Syngress-> set admin password password
```

3. Use the following command to verify the changes:

```
Syngress-> get admin user
```

You will see an output similar to the following:

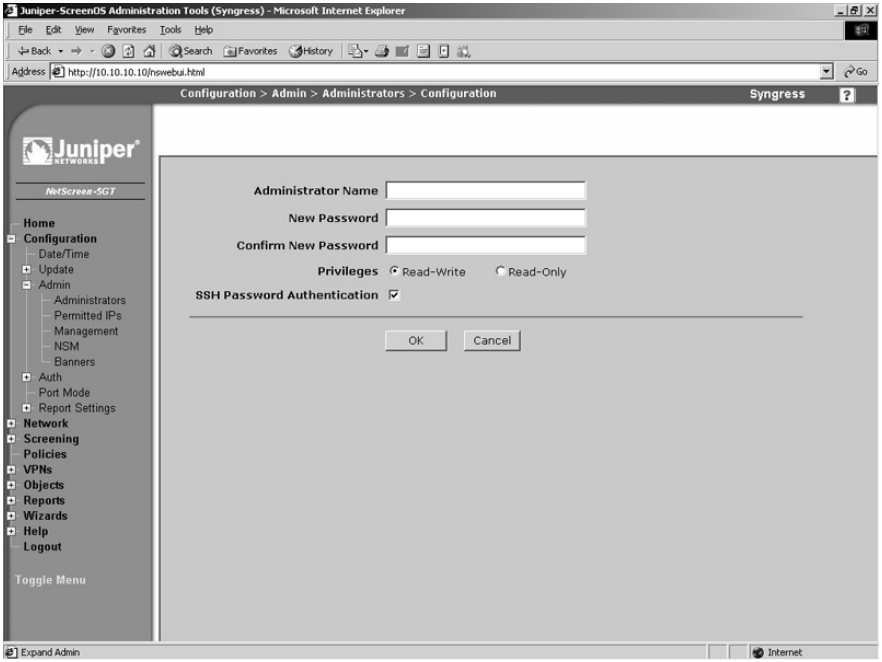
```
Name                            Privilege
------------------------------- --------------
synadmin                        Root
Syngress->
```

The device now has its root users name set to **synadmin,** and its password has been changed. It is suggested that you create a password of a minimum of eight characters. The maximum number of characters allowed in the password is thirty-one.

It is also suggested that you create a read-write administrator to use for regular maintenance. If that administrator is compromised, there will be no direct root access to the device. Use the following steps to create a read-write administrator via the WebUI:

1. Select **Configuration** | **Admin** | **Administrators** | **New**. The screen shown in Figure 3.7 will appear.

Figure 3.7 Administrator Configuration



2. Use the **Administrator Name** field to enter the new name. In this example, **backupadmin**.

3. Enter this user's password in the **New Password** and **Confirm New Password** fields.
4. Enable the **Read-Write** option.
5. Press **OK**.

Use the following to create a read-only administrator via the WebUI.

1. Select **Configuration** | **Admin** | **Administrators** | **New**.
2. Use the **Administrator Name** field to enter the new name. In this example, **roadmin**.
3. Enter this user's password in the **New Password** and **Confirm New Password** fields.
4. Enable the **Read-Only** option.
5. Press **OK**.

Enter the following command to create a read-write administrator via the CLI:

```
Syngress-> set admin user backupadmin password %so%back privilege all
```

Verify the entry by using the *get admin user* command. The output will look like the following:

```
Name                            Privilege
------------------------------- ---------------
synadmin                        Root
backupadmin                     Read-Write
```

Enter the following command to create a read-only administrator via the CLI:

```
Syngress-> set admin user roadmin password n0tru$t privilege read-only
```

Verify the entry by using the *get admin user* command. The output will look like the following:

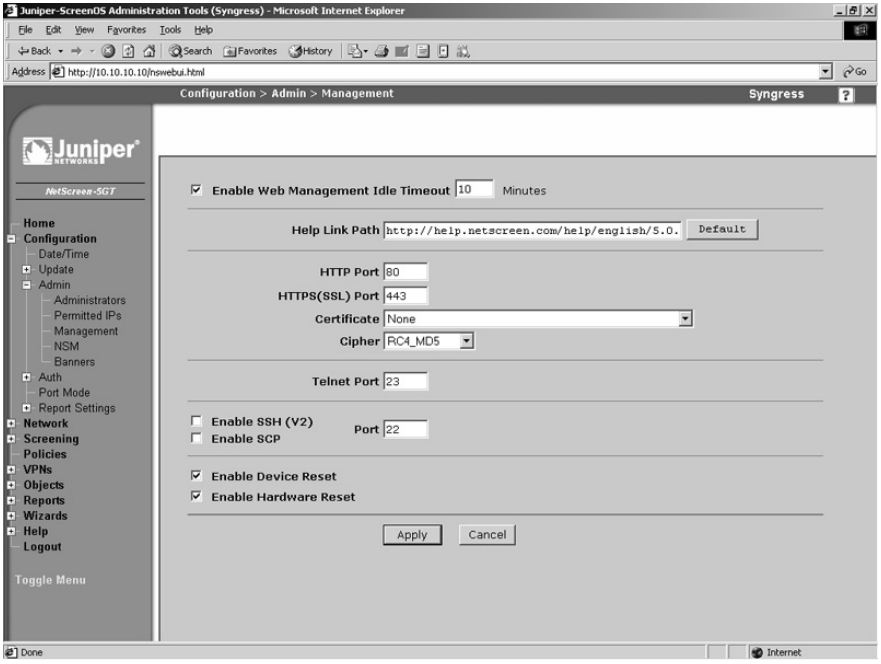
```
Name                            Privilege
------------------------------- ---------------
synadmin                        Root
backupadmin                     Read-Write
roadmin                         Read-Only
```

Another option that you should configure is the *idle timeout*. I have been to many locations where you only have to connect to the console to have a *privileged* account ready and waiting for you. This opportunity exists because the previous user left the console unattended, and they failed to log out. This is a common setup for a serious security breach. Anyone with a little know-how can cause trouble on your network if allowed to connect to your system with readily available privileged access. In order to avoid this situation, set the

idle timeout to a reasonable amount of time. The default is ten minutes for the console, Telnet, SSH, and WebUI sessions. Use the following steps to set the console, Telnet, and WebUI sessions to timeout after five minutes via the WebUI:

1. Select **Configuration** | **Admin** | **Management**. A screen similar to the one shown in Figure 3.8 will appear.

Figure 3.8 Admin Management



2. Ensure the **Enable Web Management Idle Timeout** option is enabled and type **5** in the corresponding text field.
3. Press **Apply**.

You can also modify the console timeout option via the CLI by typing **set console timeout 5**. Note that a timeout value of **0** will disable the timeout feature. Use the **get console** command to verify the change. The output will resemble the following:

```
Console timeout: 5(minute), Page size: 22/22, debug: buffer
privilege 250, config was changed and not saved!
ID State  Duration Task Type   Host
 0 Login       660 13433716 Telnet 10.254.5.32:49401
 1 Logout        0 13435768 Local
 2 Logout        0 13424824 Local
 3 Logout        0 13410460 Local
```

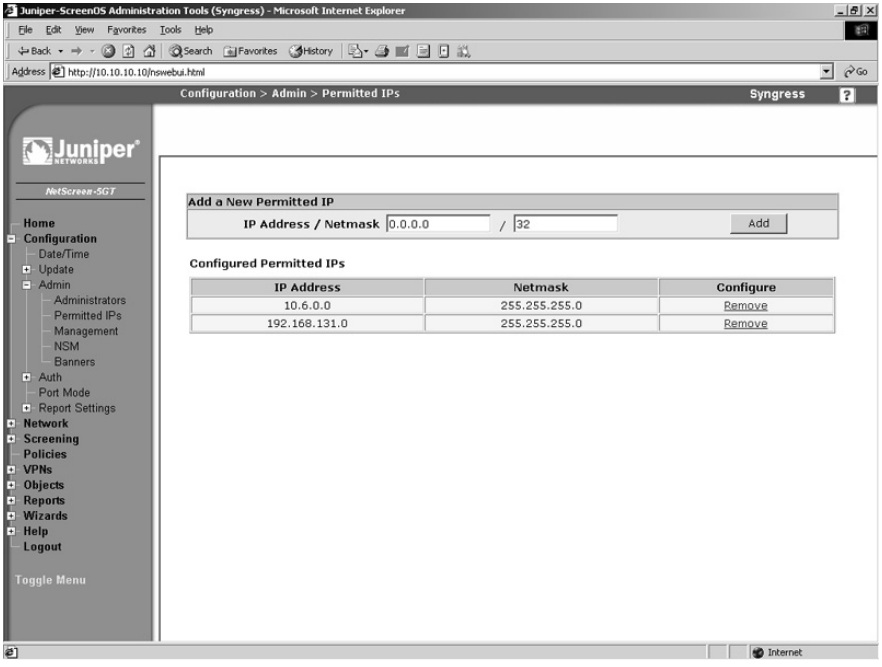
To set the admin authentication timeout, type **set admin auth timeout 5**. Use the *get admin auth* command to verify the setting. The output will resemble the following:

```
Admin user authentication timeout: 5 minutes
Admin user authentication type: Local
```

The next step is to limit system access to your firewall. By specifying *permitted* IP addresses, you can limit which IP addresses are authorized to perform management services. You are limited to a total of six entries for both network and host entries. Once you enable this setting, it immediately takes effect. If you set this up remotely, ensure that you add your own IP address and/or source network. Use the following steps to create a permitted IP address entry via the WebUI:

1. Select **Configuration** | **Admin** | **Permitted IPs**. A screen similar to that shown in Figure 3.9 will be displayed.

Figure 3.9 Permitted IPs



2. Use the available text fields to enter the IP address and netmask, and then select **Add**. You can remove an IP address from the list by selecting its **Remove** link. Note that if the list contains no IP addresses, any IP address will be able to access the firewall.

To add a permitted IP address via the CLI, type the command **set admin manager-ip *ipaddress***, where *ipaddress* is the full IP address using dotted quad (###.###.###.###) notation. You can verify the setting by entering **get admin manager-ip**. To remove an IP address entry via the CLI, type the command **unset admin manager-ip *ipaddress***.

Secure Shell is highly suggested over Telnet, as we discussed earlier when we were looking at our different management options. However, SSH must be enabled before you can use it. Again earlier we looked at using SSH version two. In the following code snippet we enable SSH version two in either the CLI, or the WebUI. After enabling SSH it may take several minutes for the SSH servers to be enabled. This is because the SSH keys are generating during this time.

Use the following steps to enable SSH via the WebUI:

1. Select **Configuration** | **Admin** | **Management**.
2. Enable the **Enable SSH (v2)** option.
3. Press **Apply**.

To enable SSH via the CLI, type the command **set ssh version v2**. To set version 1 rather than version 2, simply replace **v2** in the command with **v1**.

It is strongly recommended that you use SSL when using the WebUI. In general, it is very easy to set up and configure. Included in ScreenOS 5.2 and later is a self-signed certificate. WebUI allows you to turn on SSL right out of the box. You can also generate a certificate signing request (CSR) and submit it back to a certificate authority (CA) to get the certificate signed. Once you have the signed certificate, you can load it back onto your Juniper device. We will review how to generate the CSR, and how to load the certificate. However, signing a certificate varies based upon which certificate authority you choose. If you are using your device from your company's network, you should use a certificate purchased from a reputable Web site such as www.verisign.com or www.godaddy.com. Either site can provide you with a certificate. However, if you want to get a signed certificate for testing purposes, go to www.cacert.org to get a free one.

Use the following steps to generate a certificate request. Note that this example includes company-specific information that you should substitute with your own information.

1. Access **Objects** | **Certificates**. The screen will display the existing certificates (Figure 3.10).
2. Press **New**. The New Request screen will be displayed as shown in Figure 3.11.
3. Enter your **Name**, **Phone**, **Unit/Department**, **Organization**, **County/Locality**, **State**, **Country**, **Email**, **IP Address**, and Fully Qualified Domain Name (**FQDN**).
4. Select the Rivest, Shamir, and Adelman (**RSA)** option.
5. Select **1024** or **2048** from the **Create new key pair** drop-down list: the higher the number, the more secure the certificate.

Figure 3.10 Certificates

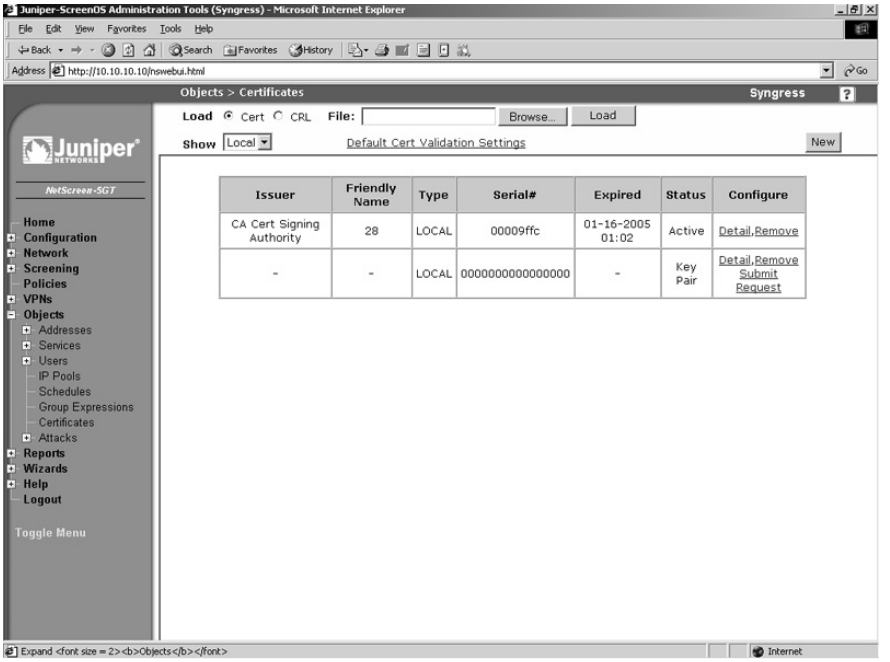
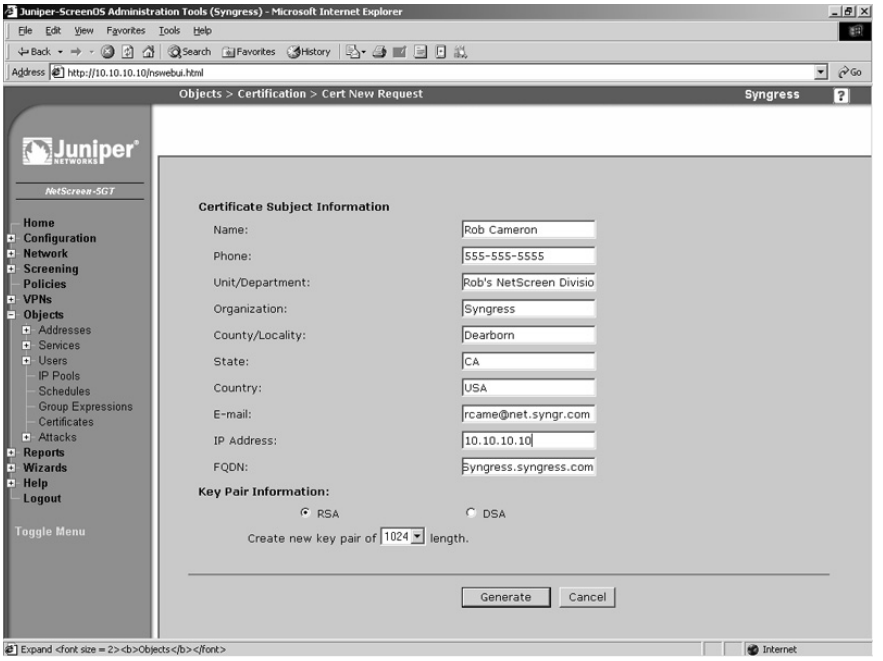


Figure 3.11 New Certificate Request



6. Press **Generate**. In several minutes a new page will displayed that contains a section of text.
7. Copy the text contents from "-----BEGIN CERTIFICATE REQUEST----" to "-----END CERTIFICATE REQUEST-----".
8. Supply this to your certificate authority. They, in turn, will supply you with a certificate file.
9. Access **Objects** | **Certificates** and select **Browse**. Choose the certificate file from the CA and select **Load**. The certificate is now active and loaded.
10. Access **Configuration** | **Admin** | **Management**. Select the certificate from the **Certificate** field.

Use the following steps to request and set up a certificate via the CLI using your own personal and company information.

1. Enter the following commands to request a certificate:

```
Syngress-> set admin mail server-name 123.123.123.100
Syngress-> set pki x509 dn country-name US
Syngress-> set pki x509 dn email rob@Juniper.com
Syngress-> set pki x509 dn ip 123.123.123.123
Syngress-> set pki x509 dn local-name "Dearborn"
Syngress-> set pki x509 dn name "Rob Cameron"
Syngress-> set pki x509 dn org-name "Rob's Juniper division"
Syngress-> set pki x509 dn org-unit-name Books
Syngress-> set pki x509 dn phone 555-555-5555
Syngress-> set pki x509 dn state-name CA
Syngress-> set pki x509 cert-fqdn manage.Juniper.com
Syngress-> set pki x509 dn default send-to rob@Juniper.com
Syngress-> exec pki rsa new-key 1024
```

2. The certificate will be e-mailed to the address you originally specified. Copy the contents starting with "-----BEGIN CERTIFICATE REQUEST----" and ending with "----END CERTIFICATE REQUEST----".
3. Supply this information to your certificate authority. They, in turn, will supply you with a certificate file. The CA may also supply you with a local certificate and a certificate revocation list (CRL). A CRL contains a list of all revoked certificates. These are certificates that the CA has signed that are no longer valid.
4. To import these files, use the following commands:

```
Syngress-> exec tftp 123.123.123.100 cert-name newcer.cer
Syngress-> exec tftp 123.123.123.100 cert-name localpro.cer
Syngress-> exec tftp 123.123.123.100 crl-name notrust.crl
```

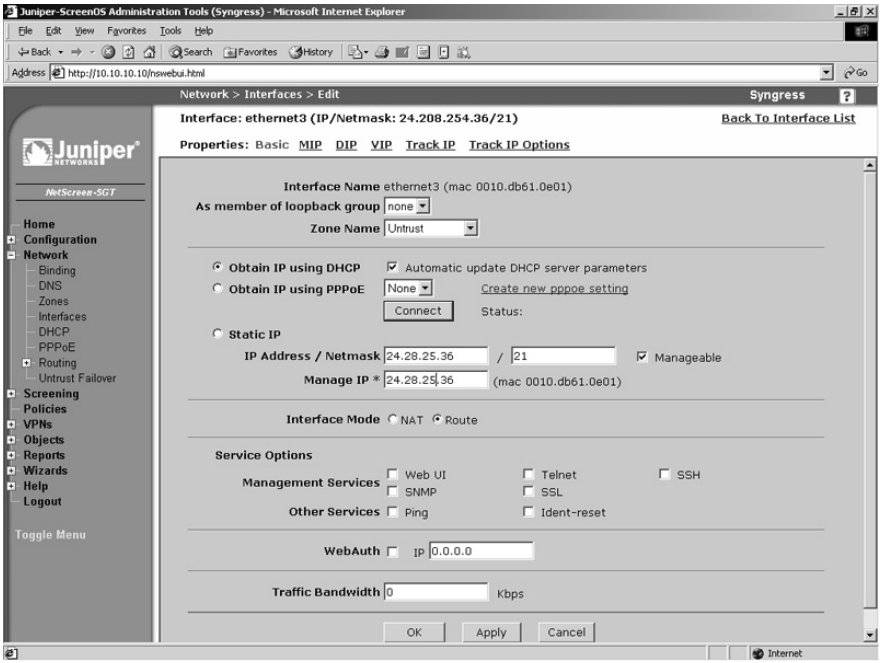
```
Syngress-> set ssl encrypt 3des sha-1
Syngress-> set ssl cert 1
Syngress-> set ssl enable
```

Now that we have the access restricted to specific hosts, there are several more options we can utilize to enhance the security. The first option is to disable unnecessary management services. Management services are bound to individual interfaces. It is important to restrict them to the bare minimum. This can be done easily from either the WebUI or the CLI. In this case, we are using a Juniper-5GT so we will be modifying the *untrust* interface. We are going to enable the WebUI, SSL for the WebUI, and SSH. We will use only the WebUI with SSL and SSH because they are secured.

Use the following steps to disable unnecessary management services via the WebUI:

1. Access **Network** | **Interfaces**. Press the **Edit** link for the entry titled **untrust.** A screen similar to Figure 3.12 will be displayed.

Figure 3.12 Editing Network Interfaces



2. Ensure that **WebUI**, **SSH**, and **SSL** are all enabled, and ensure the remaining option are disabled.
3. Press **Apply**.

To disable unnecessary management services via the CLI, type the following commands:

```
Syngress-> unset interface untrust manage ping
Syngress-> unset interface untrust manage snmp
Syngress-> unset interface untrust manage telnet
Syngress-> set interface untrust manage ssh
Syngress-> set interface untrust manage web
Syngress-> set interface untrust manage ssl
```

Use the *get interface trust* command to verify the settings. The output should resemble the following:

```
Interface untrust:
  number 1, if_info 88, if_index 0, mode route
  link up, phy-link up/full-duplex
  vsys Root, zone Untrust, vr trust-vr
  dhcp client enabled
  PPPoE disabled
  *ip 123.208.123.254/24   mac 0010.db61.1231
  gateway 123.208.123.1
  *manage ip 123.208.123.254, mac 0010.db61.1231
  route-deny disable
  ping disabled, telnet disabled, SSH enabled, SNMP disabled
  Webenabled, ident-reset disabled, SSL enabled
  webauth disabled, webauth-ip 0.0.0.0
  OSPF disabled  BGP disabled  RIP disabled
  bandwidth: physical 100000kbps, configured 0kbps, current 0kbps
             total configured gbw 0kbps, total allocated gbw 0kbps
  DHCP-Relay disabled
  DHCP-server disabled
```

Next, you can change the local port that your management services listen on. This can help prevent your services from being detected if someone were to scan for open services. Telnet (TCP 23), SSH (TCP 22), WebUI (TCP 80), and WebUI SSL (TCP 443) can each be changed to a different port number. Use the following steps to change the ports via the WebUI:

1. Access **Configuration** | **Admin** | **Administrators**.
2. Specify new port numbers for Telnet, SSH, WebUI and WebUI SSL. Note that port numbers must be in the range 1024-32767.
3. Press **Apply**.

Enter the following commands to set the port numbers via the CLI:

```
Syngress-> set admin ssh port 1024
Syngress-> set admin port 32000
```

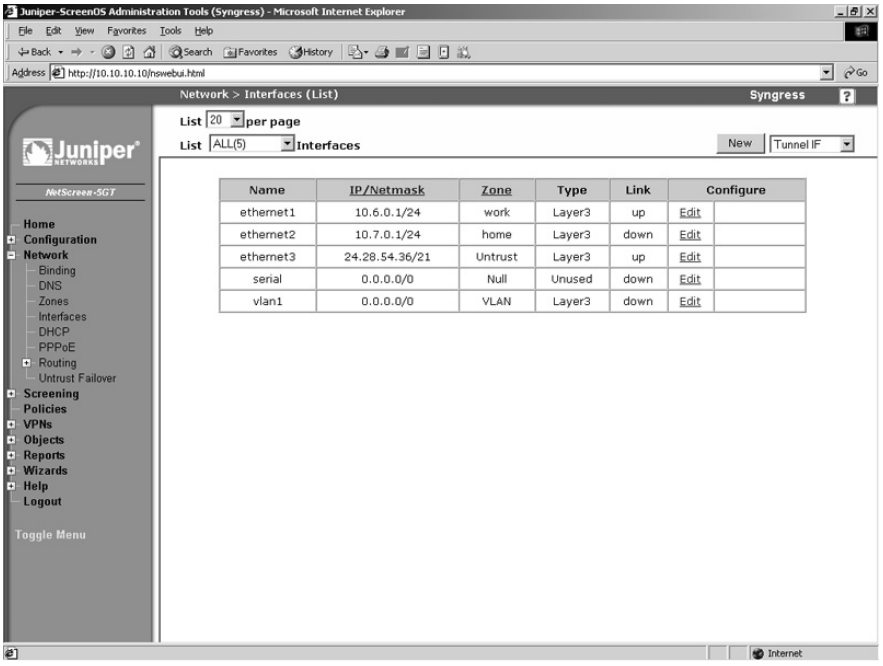
```
Syngress-> set admin telnet port 4000
Syngress-> set ssl port 5000
```

So far, we have explored interface IP address management, and it is simple to determine the IP address of the firewall. If the IP address is known, it can be used to connect to it and to manage your device. However, you can set up a management IP, which is configured directly on the interface. For this example we will be using a Juniper-5GT, and we will be modifying the *untrust* interface.

Use the following steps to set up a management IP via the WebUI:

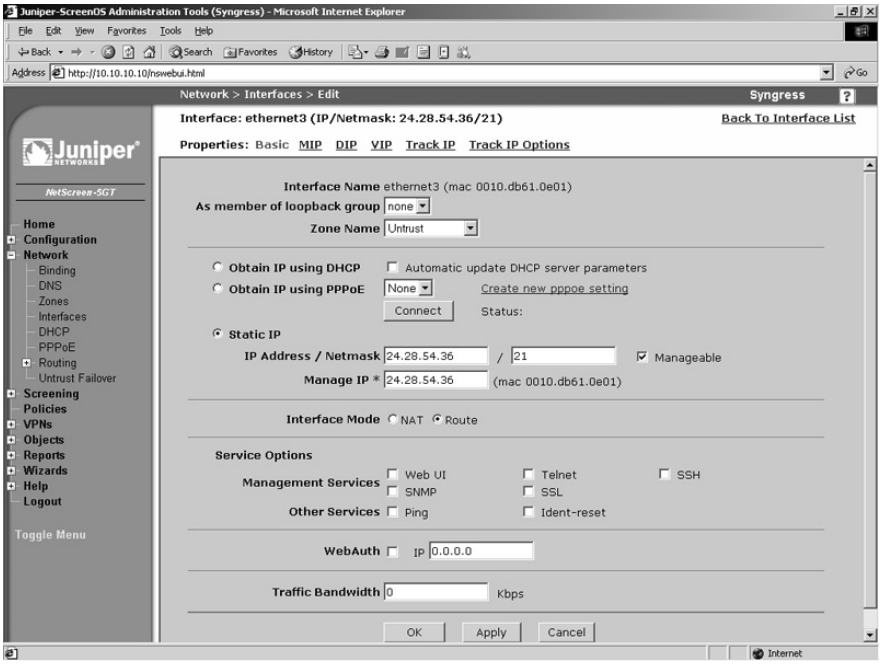
1. Access **Network** | **Interfaces (List)**. The screen shown in Figure 3.13 will be displayed.

Figure 3.13 Network Interfaces List



2. Press the **Edit** link for the *untrust* entry. A screen similar to the one shown in Figure 3.14 will be displayed.
3. Use the **Manage IP *** field to enter the new IP address.
4. Press **Apply**.

Figure 3.14 Edit Network Interface



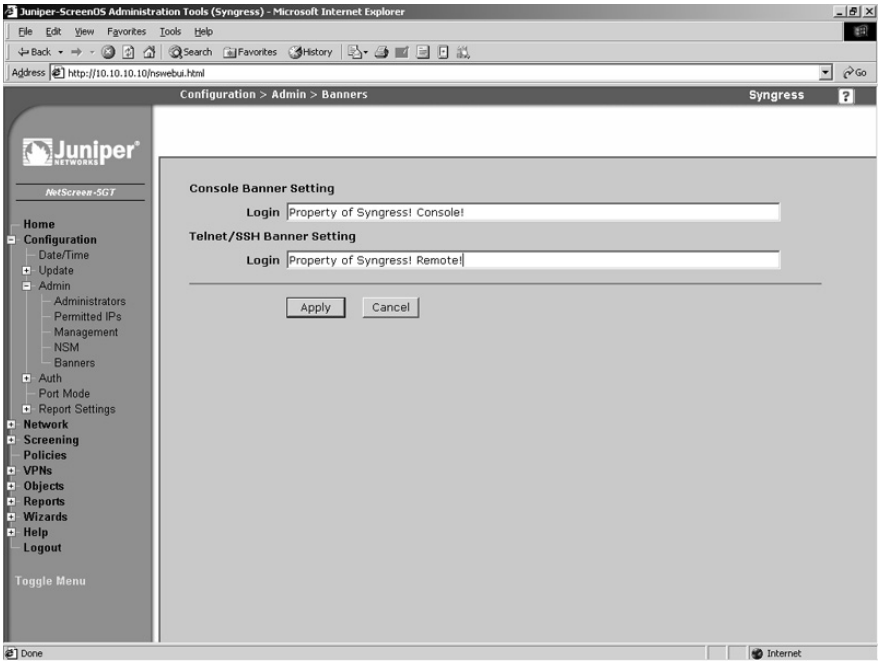
To set up a management IP via the CLI, type the command **set interface untrust manage-ip** ***ipaddress***.

For remote command line access you can set up customized login banners. This is useful to provide a legal warning, or a help message. This can also identify specific penalties for unauthorized access. There are two limitations to using banners. First, you are limited to a single line. Second, you are limited to 127 characters. A banner can be configured for both console and remote Telnet sessions. This option can be configured from either the CLI, or the WebUI.

From the WebUI:

1. Access **Configuration** | **Admin** | **Banners**. A screen similar to Figure 3.15 will be displayed.
2. Use the **Console Banner Setting Login** field to enter the login banner text that will be displayed for users using the console.
3. Use the **Telnet/SSH Banner Setting Login** field to enter the login banner text that will be displayed for users using Telnet or SSH.
4. Press **Apply**.

Figure 3.15 Banners



Use the following CLI command to set the banner for console users.

```
Syngress-> set admin auth banner console login "Only permitted individuals are
allowed to use this access. If you are not permitted please disconnect!"
```

Use the following CLI command to set the banner for Telnet users.

```
Syngress-> set admin auth banner telnet login "Authorized users only!!! All
actions are logged!!!"
```

Finally, there are three options that can be configured only from the command line that can enhance security. Two of these options will not save your system, but since they are new to the 5.0 ScreenOS release, they are worth mentioning. First, you can enforce a minimum length for administrative user passwords. Second, you can restrict how many unsuccessful login attempts that a user can have before they are kicked out of the system. The default is three and it does not lock out the user. The same person could Telnet back in to try again. Finally, you can restrict the root user to access from the console only. This can prevent anyone from gaining root access to the device unless they have physical access to it.

Use the following CLI commands to set a minimum password length, limit access attempts, and restrict root user access to the console, respectively.

```
Syngress-> set admin password restrict length 8
Syngress-> set admin access attempts 2
Syngress-> set admin root access console
```

The ideas in this section will help to secure your device. Security is all about mitigating risk. With these management security procedures in place, you significantly lower the chances of incurring a security breach. You can mix and match the configurations that work best for your environment.

Updating ScreenOS

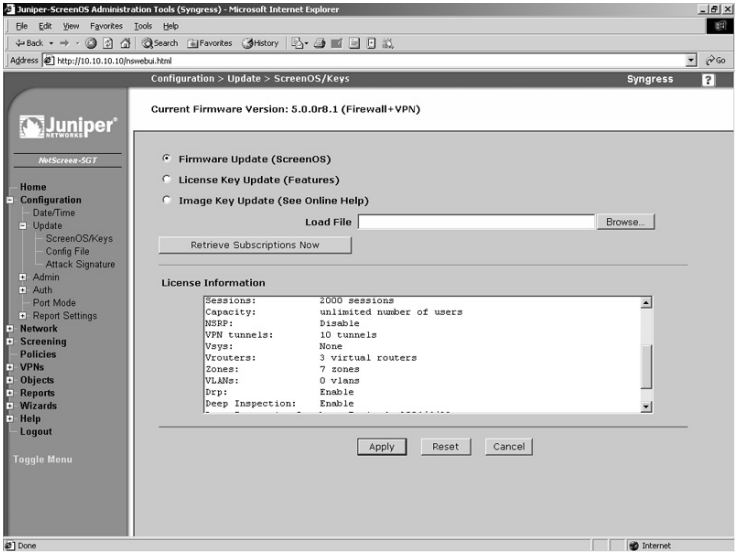
Juniper Networks is committed to providing a secure and robust operating system for Juniper firewall products. From time to time Juniper will publish a new version of ScreenOS. This may include security updates, feature enhancements, or both. It is very important that you maintain the currency of the software on your firewall. It is a core component of your network security platform, and it has to be secure. There are several methods available to upgrade ScreenOS. First, we will focus on the command line methods where you can not only update your OS, but you can back up your operating system as well. You are required to use a Trivial File Transfer Protocol (TFTP) server when you use the CLI. Use the following command to back up your software:

```
Syngress-> save software from flash to tftp ipaddress 5.0.0r8.1-5GT.bin

       Use the following command to update the software:

Syngress-> save software from tftp 1.2.3.4 5.0.0r8.1-5GT.bin to flash
```

You can also use the WebUI to update the firmware. However, as we mentioned before, you cannot download the current software from the WebUI.

1. Access **Configuration** | **Update** | **ScreenOS/Keys**. A screen similar to Figure 3.16 will be displayed.

Figure 3.16 ScreenOS/Keys



2. Enable the **Firmware Update (ScreenOS)** option.
3. Press **Browse** and locate and select the previously downloaded firmware file, which is stored on the local system.
4. Press **Apply**. It may take several minutes to update the system with the new OS.

System Recovery

There may be times when your Juniper firewall runs into problems from which you cannot recover. Three scenarios are covered in this section. One of the major issues is *configuration management*. There may be scenarios that cause you to make changes where you are unsure of the repercussions. For example, you may be adding a new route, or a new policy that could wreak havoc on your network, though you are actively running on a successful configuration. In cases where you need a backup copy of a correctly functioning configuration file, you can use the *configuration rollback feature*.

The configuration rollback feature allows you maintain a backup configuration file that you can use in case your primary configuration file, saved or running, runs into problems. The configuration rollback cannot be performed from the WebUI. Use the following steps to save your system configuration.

1. Use the command *get file* to get a list of files in flash memory.
2. Enter the command **save config to last-known good**. A new file called lkg.cfg will be created. This file is your rollback configuration file. It is a saved copy of the *running* configuration at the time you executed the command. That file stays on the system unless you explicitly call the *delete* command to remove it. This means that even if you reset the configuration to the defaults, you still have this configuration available for use.

To restore a previously saved system configuration, type the command **exec config rollback**. Note that this process *forces* your device to reboot.

As long as the file exists, you can use this restoration process at any time. There is one additional way to use configuration rollback. If you are working on a new configuration that could possibly cause you to lose access to your system for any reason, configuration rollback can be placed in *watching* mode. In this mode, if the device is reset, it will automatically reset the configuration to the stored rollback configuration. This is a life saver in cases where you need to ensure the safe restoration of your device's provided networking services.

To put the rollback in watching mode, type the command **exec config rollback enable**. The command prompt will include the text "rollback enabled". To turn this mode off, type **exec config rollback disable**.

Now that we have discussed how to recover your configuration, we need to look at another scenario. What if you lose your root password? This is a tough situation to recover

from, because you have lost all access to the system. There are two methods to recover from this error. Both methods require you to have console access to the device. In the first scenario, you would log into the serial console using the *serial number* of the device as the username and password. Once you do this, you will be notified that you will lose your configuration and all your settings. If you have performed proper configuration management, you will be fine. Note; even the configuration rollback file is deleted. So you must have saved your configuration somewhere other than the system if you want to be able to use it to restore service in an emergency.

The following shows a typical serial number login and the resulting messages.

```
login: 00642120012308289
password:
!!! Lost Password Reset !!! You have initiated a command to reset the device to
factory defaults, clearing all current configuration and settings. Would you like
to continue?  y/[n] y


    !! Reconfirm Lost Password Reset !! If you continue, the entire
    configuration of the device will be erased. In addition, a permanent
    counter will be incremented to signify that this device has been reset.
    This is your last chance to cancel this command. If you proceed, the
    device will return to factory default configuration, which is: System IP:
    192.168.1.1; username: netscreen, password: netscreen. Would you like to
    continue?  y/[n] y
```

Another way to access a system when you have forgotten the root password is to use the *reset button* located on the exterior of the system. To use this type of configuration use the following procedure:

1. Use a pin, place it in the resent hole, push and hold for at least four to six seconds. The status LED will blink amber once per second.
2. Wait for the status LED to begin blinking, and then remove the pin from the reset hole.
3. Wait one to two seconds, and replace the pin in the reset hole, push and hold for at least four to six seconds.
4. Wait for the status LED to turn red, and then eventually to begin blinking green before you release the pin from the reset hole.

Doing this will reset the system, and you will lose all your configurations. This is done for security purposes. These are both powerful methods available to recover your device; however, you may want to disable these options. You may not want someone to be able to walk up to your device and reset your configuration. Both methods can be disabled. However, if you disable them, the device will be unrecoverable if you lose the root password. Therefore, do not lose your root password unless you want to physically return the device to Juniper Networks.

To disable the ability to log in using the serial number, type **unset admin device-reset**. To re-enable this feature, type **set admin device-reset**. To disable the device's reset button, type **unset admin hw-reset**. To re-enable this feature, type **set admin hw-reset**.

In the previous section we looked at ways to upgrade ScreenOS. However, there are many ways in which the image can be corrupted during upload. More than likely, the file was damaged *before* you uploaded it. To restore your system to a functional configuration, you must have serial console access to the system, and a TFTP server on the local network to the device. During the boot process, a prompt will be displayed four times. The prompt will say, "Hit any key to run loader. Press any key, and you will be asked for the file you want to load, the IP address you want to assign to your device, and the IP address of the TFTP server. The interface that receives the IP address you assign is one of the following depending on what type of device you have: Trust, E1, or E1/1. If the file can be found on the TFTP server, it will be loaded into flash, and your device will reboot. When the device reboots it will load the new OS image.

```
Juniper NS-5GT Boot Loader Version 2.1.0 (Checksum: 61D07DA5)
Copyright (c) 1997-2003 Juniper Technologies, Inc.

Total physical memory: 128MB
    Test - Pass
    Initialization.... Done

Hit any key to run loader
Hit any key to run loader
Hit any key to run loader

Serial Number [0123012123008289]: READ ONLY
HW Version Number [1010]: READ ONLY
Self MAC Address [0010-db61-1230]: READ ONLY
Boot File Name [ns5gt.5.0.0r8.1]:
Self IP Address [192.168.1.1]:
TFTP IP Address [192.168.1.31]:

Save loader config (56 bytes)... Done
```

Configuring Your Firewall for the First Time

Now that you are familiar with the basics of managing your Juniper firewall, it is now time to configure your firewall. This section discusses basic configuration requirements to make your system functional on your network. There are three basics for getting your device up and running on the network. The first thing you need is a *zone.* We touched on zones in the previous chapter. In this section we will explore how to use existing zones, create new zones, and how to bind zones to interfaces. The primary type of zone that exists is the *security zone*, but there are several other types of zones that can be used. It is important to know how each type of zone functions, because it determines how an interface will function. Some zones may never be used; however, being aware of their existence is important.

There are several types of interface on a Juniper firewall. You will always have physical interfaces because they are required in order to connect to the network. Juniper also offers several other types of interfaces. These interfaces provide different functions, and they are not all physical devices. These types of interfaces include *subinterfaces*, *management interfaces*, *high availability interfaces*, and *tunnel interfaces*. Each type of interface was designed to provide a specific function on the Juniper device. We will look at each interface type, its function, and how you can leverage their special abilities on your network.

Your newly configured interface will require an IP address if you want it to interact with your network. In Chapter 1 we discussed IP addressing. It is assumed that you are already familiar with IP addressing, and that you have used it on at least one type of system. The process is similar for every device because each system operates on the IP standard. A Juniper firewall is no exception.

Some Small Office Home Office (SOHO) class devices have a configuration mode called *port mode*. The SOHO devices have five physical interfaces. By default, there is one external *untr*ust interface and four *trus*t interfaces. However, you can change the port mode number to modify the distribution of ports. This feature can be used to extend the value of the SOHO class devices. In this section we will also look at the various options you can use when configuring a network interface using the built-in PPPoE client.

Types of Zones

There are three types of zones on a Juniper firewall. Each zone provides its own specific function, and each is used for a specific purpose. The *security zone* is the most commonly used zone type. The other two zone types are used much less commonly. One of these types is the *tunnel zone*. This type of zone is used for creating route-based VPNs. The other type of zone is the *function zone*. This zone is used for special purposes in high availability. Each type of zone is used to bind to an interface.

Security Zones

A security zone is used to break your network into logical segments. At a minimum, you need to define two security zones. Most Juniper firewall devices come with predefined zones that you can use. These zones are usually trust, untrust, and demilitarized zone (DMZ); however, this varies from device to device. You need to use two zones because this will allow you to separate your network into two parts. Each Juniper firewall can use only a limited number of zones. On some devices you can only have a few, while on the higher-end firewalls you could have several hundred zones. There is another type of security zone called a layer two zone, which is covered in a later chapter.

Tunnel Zones

Tunnel zones are used with tunnel interfaces. Tunnel interfaces are a special type of virtual interface that can terminate VPN traffic. Tunnel interfaces are first bound to the tunnel zone.

Then the tunnel zone is bound to a security zone, which is in turn bound to a physical interface. Tunnel zones are covered in depth in Chapters 11 and 14.

Function Zones

There are five types of function zone, and each is used to provide a single, unique function. The first type is the *null zone*. The null zone is used as a placeholder for interfaces that are not bound to a zone. The next type of function zone is the management (MGT) zone. This zone is used on out-of-band management interfaces. The high availability (HA) function zone is used for high availability interfaces. There are no configurable options for the HA zone. The *self zone* is used to host management connections. When using the remote management protocols to connect to, and manage, your Juniper device, you are connecting to the self zone. The last type of zone is the *virtual local area network* (VLAN) zone. It is used to host the VLAN1 interface. The VLAN1 interface is used to manage a Juniper firewall that is running in transparent mode.

Virtual Routers

As we have discussed, any device that uses the IP protocol must have a routing table that determines how to send information from one place to another. Juniper takes this idea to a whole new level by allowing you to have multiple routing tables, or virtual routers. Each virtual router has its own routing table that is complete and separate routing domain from other virtual routers. In this chapter, we will discuss the trust virtual router, and how to configure routes in it. A full explanation of routing is covered in Chapter 7.

Types of Interfaces

A Juniper firewall can contain several types of interfaces. An interface allows traffic to enter a zone and leave a zone. If you want an interface to pass traffic, you need to *bind* it to a zone. Once you bind an interface to a zone, you can apply an IP address to it. There are four types of interfaces: security zone interfaces, function zone interfaces, tunnel interfaces and loopback interfaces. As you can see, each type of interface has a corresponding zone type, except for the loopback interface, which is a special type of interface.

Security Zone Interfaces

Security zone interfaces are used primarily for passing traffic from one zone to another. In this category any type of interface related to physical interfaces or virtual interfaces belongs in this category. This is the interface that you will more commonly work with.

Physical Interfaces

Every Juniper firewall has some kind of physical interface. Physical interfaces are used to connect the firewall to the network. The naming convention of the physical interfaces varies

based on the platform used. On the SOHO class of Juniper appliances, the interface names are based upon the zones. For example, the internal interface is named trust and the external interface is named untrust. On the Juniper-25 through the Juniper-208 products, the interfaces are named beginning with the media type, *Ethernet,* and then specified by the port number, such as *Ethernet1*. Juniper firewalls that are systems are named using the media type, slot number, and then the port number. For example, *Ethernet2/1* would be an Ethernet interface in slot number two, and port number one. The Juniper-500, ISG-2000, Juniper-5200, and Juniper-5400 belong to this category. Physical interfaces can be assigned a single primary IP address.

There are some situations where you may need to have multiple IP address on an interface. You can add multiple secondary IP addresses on each physical interface. When a secondary IP address is added, the Juniper firewall automatically adds a route between the two IP address segments. In this way you can connect the two segments. The route will automatically be removed if you delete the secondary IP address. If you want to segment these two networks, you can disable routing between the two. This will drop packets between the two, but the routing table will not be modified.

Secondary IP addresses have some restrictions as well. First, subnets between the multiple secondary interfaces *cannot* overlap. Secondly, interfaces in the untrust zone are unable to use multiple secondary IP addresses. If you choose to manage your firewall with the secondary IP address, it inherits the management properties of the primary interface. The secondary interface is unable to have a gateway, which means anything connecting to that interface *must* be on that local network.

Subinterfaces

Subinterfaces are used primarily with VLANs. For example, if you had a network that contained several VLANs, a Juniper firewall could act as a central point to connect between the separate VLANs. Each subinterface acts like a physical interface. All of the subinterfaces that are bound to a physical interface can use only the bandwidth that is provided by that interface. So if you have a single 100Mbps interface and several subinterfaces, they can only share the maximum bandwidth of that 100Mbps interface. The properties of a subinterface are otherwise identical to that of a physical interface. However, each subinterface *must* be assigned to a different VLAN and they *must* have a different IP subnet than all of physical interfaces, and the other subinterfaces defined on the firewall.

Aggregate Interfaces

When you create an aggregate interface you are binding multiple physical interfaces together to create one super interface. This interface acts as if it were a single physical interface. It provides cumulative bandwidth. So if you bound two 1-gigabit interfaces together, you would have a combined throughput of 2Gbps for that interface. If one of the interfaces were to fail, the remaining interface would continue to carry the traffic. However, that remaining interface can only carry as much traffic as the interface is rated for. So if you had two gigabit

interfaces bound together, and you lost one, you would lower your maximum throughput to 1Gbps. This feature is only available on the Juniper-5200, and the Juniper-5400 system.

Redundant Interfaces

The redundant interface is much like the aggregate interface, but has only one of the two benefits of the aggregate interface. Redundant interfaces are *unable* to combine their bandwidth, and they provide redundancy only in case of a failure.

VLAN1 Interface

The VLAN1 interface is used for one purpose. When you configure a Juniper firewall to operate in transparent mode, the physical interfaces do not have IP addresses. You need a way to manage the firewall, and to terminate VPNs. The VLAN1 interface is a virtual security interface that can have an IP address assigned to it. This allows you to remotely manage your firewall, and to have an IP address to terminate VPNs. Using a Juniper firewall in transparent mode is covered in Chapter 9.

Virtual Security Interfaces

The last type of security interface is the virtual security interface (VSI). This type of interface is used when two Juniper devices are used in a high availability configuration. The two firewalls are combined to create a single entity called a virtual security device (VSD). Each device in the cluster defines a physical interface to create a VSI. This VSI has its own MAC address, its own IP address, and it operates like a physical interface. Configuring and using VSIs and VSDs are covered in Chapter 14.

Function Zone Interfaces

Function zone interfaces are special interfaces that are used for a single purpose, or task. These interfaces are dedicated to that task, and they cannot be used to do anything else.

Management Interfaces

Some Juniper firewalls contain an interface dedicated for management of the device. This interface is called the MGT interface. It allows you to separate the management of the device from the rest of the network by using this special interface. It is ensures that you will have bandwidth for management applications. Because the interface does not pass general-purpose traffic, it provides additional security by being dedicated only to management.

HA Interfaces

On Juniper systems, Juniper-500 and later models, each device contains two HA interfaces, HA1, and HA2. These interfaces are used exclusively for high availability. One interface passes control messages to each device. The second HA interface is used for traffic synchro-

nization. If one of the interfaces fails, the remaining HA interface would provide both services. You must use a minimum of 100Mbps interfaces for high availability interfaces.

Some devices that can function in a HA cluster do not have dedicated interfaces for high availability. You can use a *virtual* HA interface, which is bound to a physical interface. This allows you to use the high availability configurations even though you do not have a dedicated interface to do so.

Tunnel Interfaces

A tunnel interface is used as a gateway to a VPN. This allows you to create a VPN configuration, and to bind that VPN to the tunnel interface. If you want to pass traffic to the VPN, you simply create a route on your firewall to point to the tunnel interface for the remote network. The VPN will be automatically established, and traffic will be encrypted before being sent to the remote gateway. Tunnel interfaces are used only for VPNs. VPNs are explained in Chapter 11.

Loopback Interfaces

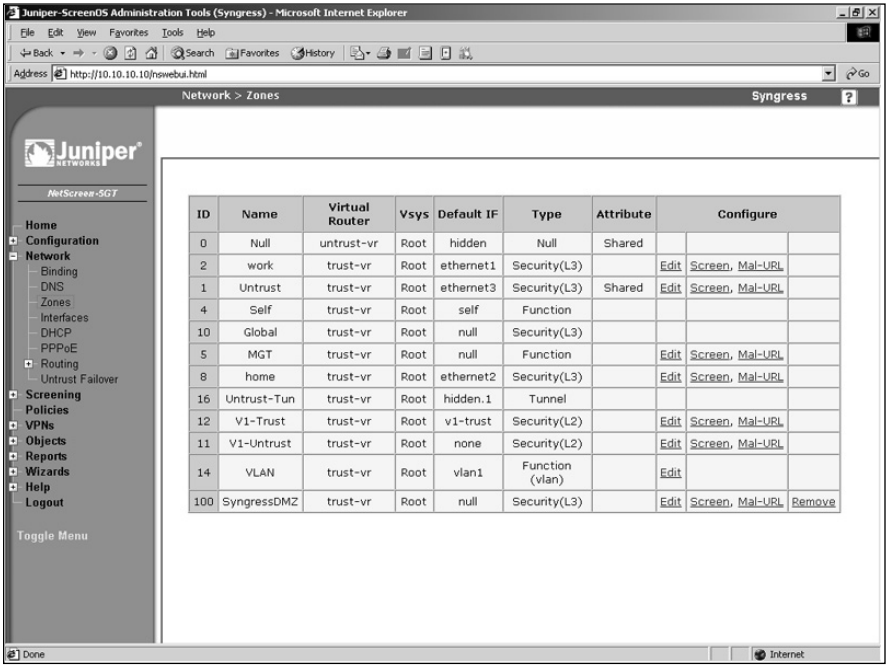
The last type of interface is the loopback interface. The loopback interface is a special interface that exists logically inside the firewall. A loopback interface is assigned to a zone, and it is not accessible from other zones unless you specify a policy to permit that traffic. A loopback interface can be used to manage your firewall.

Configuring Security Zones

Security zones are the core for creation of policies in the Juniper firewall. Policies are discussed in the next chapter. It is important that you become an expert on managing security zones. Once you have the security zones created and configured, it will be much easier for you to effectively create policies. As mentioned before, there will be several predefined security zones on your firewall. These are typically trust, untrust, and DMZ. The trust zone is designed for the internal protected network. The untrust zone is designed typically for the Internet or other undesirable places. The DMZ zone is used for your DMZ network. The trust zone and untrust zone have some unique properties that will be discussed later in this chapter. The predefined zones cannot be deleted, but they can be *modified*. In ScreenOS version 5.4 and later, these zones no longer count toward the upper limit of your device. Previously, you were allowed eight zones on the device, and three (Trust, Untrust, and DMZ) were already taken. You can now create eight user zones for the device.

First, we will inspect zone configurations on our device. This can be done from both the command line as well as the WebUI. To view the zones using the WebUI, access **Network** | **Zones**. A screen similar to the one shown in Figure 3.17 will be displayed.

Figure 3.17 Network Zones



ID	Name	Virtual Router	Vsys	Default IF	Type	Attribute	Configure		
0	Null	untrust-vr	Root	hidden	Null	Shared			
2	work	trust-vr	Root	ethernet1	Security(L3)		Edit	Screen, Mal-URL	
1	Untrust	trust-vr	Root	ethernet3	Security(L3)	Shared	Edit	Screen, Mal-URL	
4	Self	trust-vr	Root	self	Function				
10	Global	trust-vr	Root	null	Security(L3)				
5	MGT	trust-vr	Root	null	Function		Edit	Screen, Mal-URL	
8	home	trust-vr	Root	ethernet2	Security(L3)		Edit	Screen, Mal-URL	
16	Untrust-Tun	trust-vr	Root	hidden.1	Tunnel				
12	V1-Trust	trust-vr	Root	v1-trust	Security(L2)		Edit	Screen, Mal-URL	
11	V1-Untrust	trust-vr	Root	none	Security(L2)		Edit	Screen, Mal-URL	
14	VLAN	trust-vr	Root	vlan1	Function (vlan)		Edit		
100	SyngressDMZ	trust-vr	Root	null	Security(L3)		Edit	Screen, Mal-URL	Remove

To view the zones using the CLI, type the command **get zone**. You will see each zone listed in an output similar to the following:

```
Total 10 zones created in vsys Root - 5 are policy configurable.
Total policy configurable zones for Root is 5.
------------------------------------------------------------------------
  ID Name                Type     Attr     VR          Default-IF   VSYS
   0 Null                Null     Shared untrust-vr   hidden       Root
   1 Untrust             Sec(L3) Shared trust-vr     untrust      Root
   2 Trust               Sec(L3)         trust-vr     trust        Root
   4 Self                Func            trust-vr     self         Root
   5 MGT                 Func            trust-vr     null         Root
  10 Global              Sec(L3)         trust-vr     null         Root
  11 V1-Untrust          Sec(L2)         trust-vr     None         Root
  12 V1-Trust            Sec(L2)         trust-vr     v1-trust     Root
  14 VLAN                Func            trust-vr     vlan1        Root
  16 Untrust-Tun         Tun             trust-vr     hidden.1     Root
------------------------------------------------------------------------
```

Both the WebUI and the CLI look very similar regarding the way that zones are displayed. Both show the following information:

- **ID** The ID is used when doing debugging. It is important to understand where to locate the zone ID.
- **Name** The name is used as a label for the zone.
- **Type** This tells you what type of zone this is. As you can see, there are several of the zone types we have mentioned.
- **Attr** This specifies any additional attributes for the zone. *Shared* means that the zone is shared among all local virtual systems. By default, untrust and null are shared.
- **VR** This specifies which virtual router that the zone is operating in.
- **Default-IF** This identifies which interface is bound to the zone by default.
- **VSYS** This lists which vsys, or virtual system, the zone is bound to.

It is a simple task to create a new zone. However, before doing so, you should know the following information:

- **Name** A descriptive name for your zone. If you have a DMZ for Webservers, naming it WebDMZ is more helpful than if you chose DMZ02. This is a personal preference; however, if you are creating a layer two security zone, the zone must be prefixed with ***L2-***
- **Type of zone** You can create three types of zones: security layer three zones, security layer two zones, and tunnel zones.

This is the minimum information you would need to configure a zone. There are some additional options that can be configured on a zone.

- **Screen** Screen options are defense options that protect against specific attacks, and malicious traffic. Chapter 10 covers this topic in more detail.
- **Malicious URL protection** This feature provides pattern matching for HTTP traffic. It allows you to identify malicious universal resource locators (URLs) and to block those requests.
- **Block Intra-Zone Traffic** If this option is selected, it will allow you to block traffic between two interfaces bound to the same zone.
- **If TCP non SYN, send RESET back** This option is valid only for layer three security zones and tunnel zones. If this option is enabled, the Juniper firewall will send a RESET TCP packet to any host that sends a TCP segment with a TCP flag set to something other than SYN, and that does not belong to an existing session. If you have SYN checking enabled, from CLI type **set flow tcp-syn-check**, the unsolicited SYN packet is dropped, and the session initiator is notified to reset the TCP connection without initializing a new session in the session table. If the Juniper firewall were to skip sending the RESET notice, the system attempting to

initiate the session would continually send SYN packets until its connection attempt timed out. If SYN checking is disabled, the Juniper firewall passes the SYN packet to the end system if a policy permits it. This is useful for blocking packets that can be used in different types of network scans. If you are unsure if this will help you, it is best to leave it at the default setting.

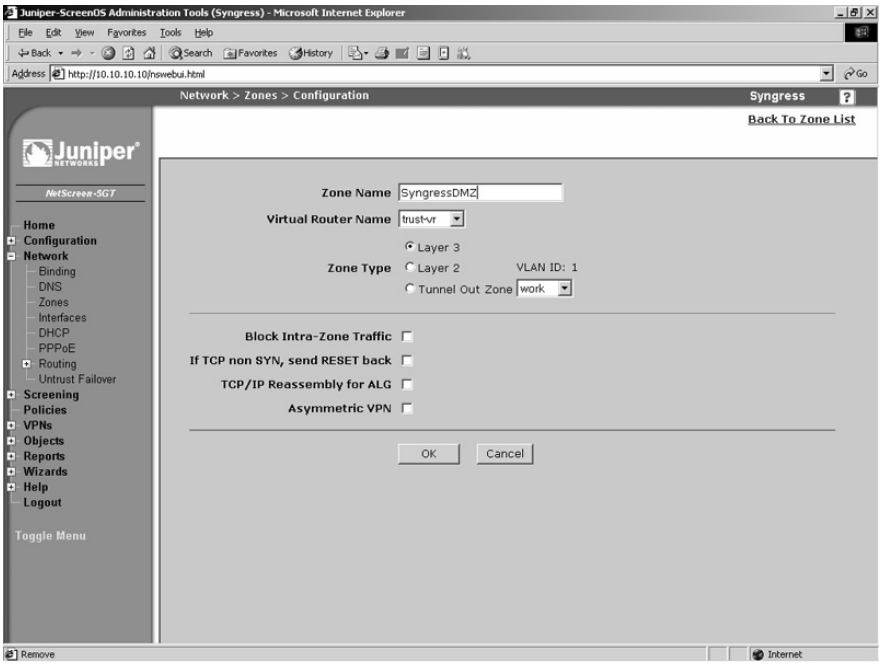
- **IP/TCP Reassembly for ALG (Application Layer Gateway)** If this option is selected, the Juniper firewall will reassemble fragmented HTTP and FTP packets before they are inspected. This will allow for more efficient enforcement for the Mal-URL engine to inspect the traffic. If you are not using the Mal-URL feature, leave this option off.
- **Shared Zone** This option is only available if you have a Juniper device that supports virtual systems. This option enables the zone to be shared among all virtual systems. Once you enable this option, you cannot disable it. You must either delete the zone, or disable all virtual systems, in order to disable it.
- **IP Classification** This option is used only with virtual systems. If this option is selected, the firewall will associate all traffic with this zone to a specific virtual system.
- **WebUI** (layer two zones only) Selecting this option enables management for the WebUI on this zone.
- **SNMP** (layer two zones only) Select this option to enable Simple Network Management Protocol (SNMP) services on this zone.
- **Telnet** (layer two zones only) Select this option to enable Telnet management on this zone.
- **SSL** (layer two zones only) Selecting this option enables SSL WebUI management on this zone.
- **SSH** (layer two zones only) Selecting this option enables SSH management on this zone.
- **NSM** (layer two zones only) Selecting this option enables NSM management on this zone.
- **Ping** (layer two zones only) Selecting this option enables *ping* from the firewall in this zone.
- **Ident-reset** (layer two zones only) Some services such as SMTP and FTP send an ident, or identification request. If you have Ident-reset enabled, it will reset this ident request and allow you access to that service.
- **WebAuth** (layer two zones only) Selecting this option enables Web authentication when traffic passes through the interface to which this zone is bound.

Generally, you would define the name for the new zone, and specify its type. However, it is always a good idea to familiarize yourself with available options when creating a new zone.

As we step through the zone creation process, we will focus on layer three zones, and the other zone types will be covered in later chapters. Use the following steps to create a zone using the WebUI:

1. Access **Network** | **Zones** and select **New**. A screen similar to Figure 3.18 will be displayed.

Figure 3.18 Create a New Zone



2. Enter the **Zone Name**.
3. Ensure **trust-vr** is selected in the **Virtual Router Name** drop-down list.
4. In the **Zone Type** section, select the **Layer 3** option.
5. Press **OK**.

To create a zone using the CLI, type the command **set zone name *name***, where *name* is the name for the zone.

Once a zone is created, you can modify all of its properties except for its name. To change the name, you must delete the zone, and then re-create it using the desired name. Use the following steps to delete a zone using the WebUI:

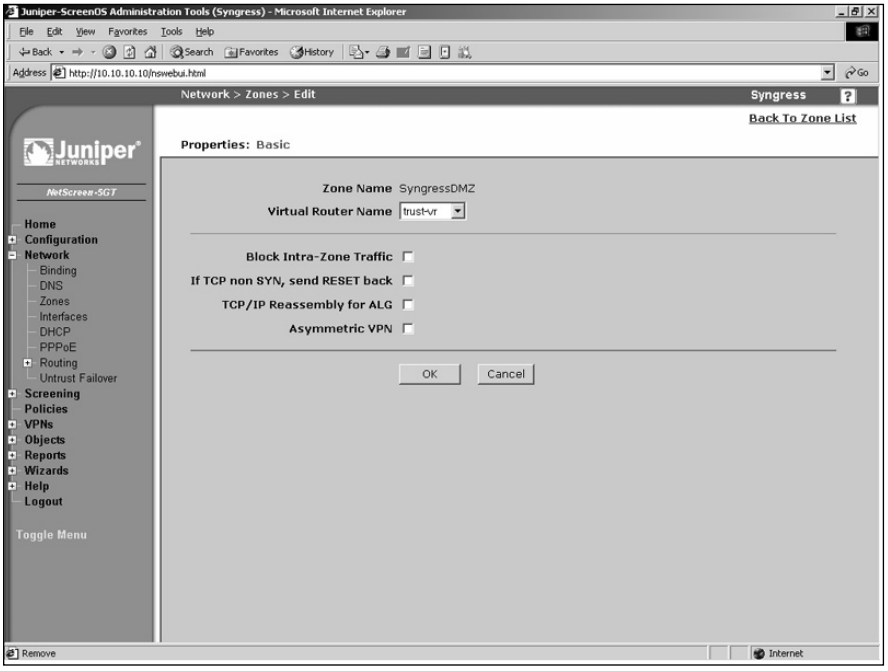
1. Access **Network** | **Zones** and select the **Remove** link of the zone you wish to delete.
2. Press **OK** to confirm.

To remove a zone using the CLI, type the command **unset zone *name***, where *name* is the name of the zone you wish to remove.

Use the following steps to modify an existing zone via the WebUI:

1. Access **Network** | **Zones** and select the **Edit** link of the zone you wish to modify. A screen similar to the one shown in Figure 3.19 will be displayed.

Figure 3.19 Edit a Zone

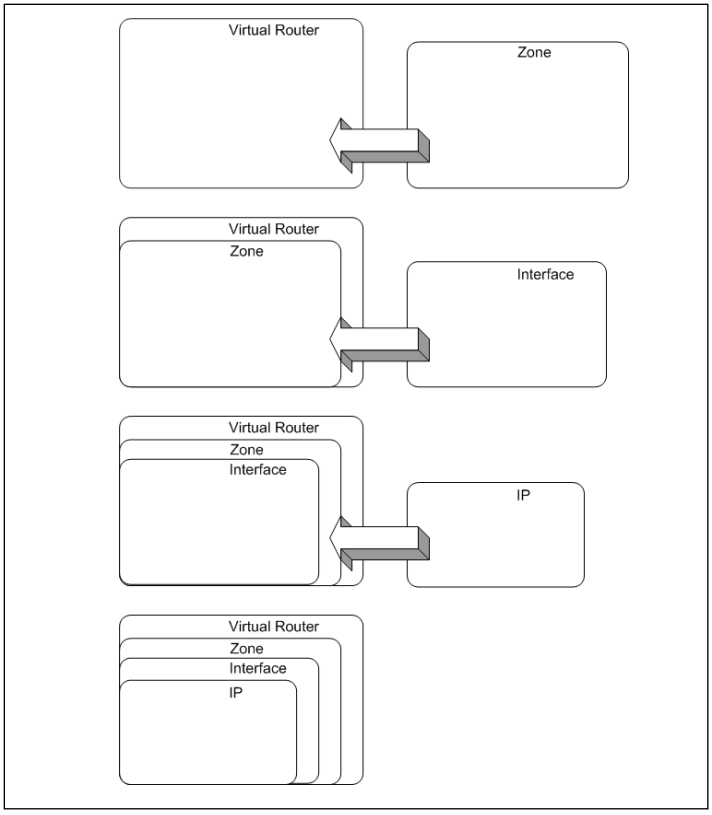


2. Change the desired fields and select **OK**.

Configuring Your Firewall for the Network

When configuring a Juniper device, there are several steps you should perform before it can interact with the network. A physical interface must first be bound to a zone before it can be assigned an IP address. Figure 3.20 depicts the relationship between a zone and an interface. A zone is a *parent* to a physical interface, and the IP address is a *child* to the physical interface.

Figure 3.20 Zone Interface/IP Relationship



Binding an Interface to a Zone

First we will bind an interface to a zone. In this case we will use a NetScreen-5GT, and we will bind the trust zone to the trust interface. This can be done using either the WebUI or the CLI. However, to change the zone you must first remove the IP address by setting it to **0.0.0.0/0.** Afterward, you can select a new zone.

From the WebUI:

1. Access **Network** | **Interfaces**.
2. Press the **Edit** link for the **trust** interface.
3. Select **Trust** from the **Zone Name** drop-down list.
4. Press **OK**.

To bind an interface to a zone using the CLI, type the command **set interface *interfacename* zone *zonename***, where *interfacename* is the name of the interface you wish to bind, and *zonename* is the name of the zone you wish to bind the specified interface to.

Setting Up IP Addressing

We will now assign an IP address of 192.168.0.1 with a twenty-four-bit subnet mask to the interface. This can be done using either the WebUI or the CLI. If you want to modify the IP address of an interface, it is the accomplished using the same steps that you would use to set it up for the first time.

From the WebUI:

1. Access **Network** | **Interfaces** and select the **Edit** link for the **trust** interface, or whichever interface you want to bind.
2. Select the **Static IP** option.
3. Enter **192.168.0.1,** or whichever IP address you want to assign, in the IP address text field, and type **24, or another numerical value to represent the bits,** in the netmask text field.
4. Press **OK**.

To assign an IP address to an interface using the CLI, type the command **set interface *interfacename* ip *ipaddress netmask***, where *interfacename* is the name of the interface, *ipaddress* is the IP address you want to assign, and *netmask* is the netmask.

Configuring the DHCP Client

Here we take the Juniper-5GT and set the untrust interface to receive an IP address from the Dynamic Host Configuration Protocol (DHCP). This will allow the Juniper firewall to be plugged into any cable modem, DSL, or internal network to seamlessly receive an IP address.

From the WebUI:

1. Access **Network** | **Interfaces** and select the **Edit** link for the **untrust** interface, or whichever interface you want to configure.
2. Select the **Obtain IP using DHCP** option.
3. Enable the **Automatic update of DHCP server parameters** option.
4. Press **OK**.

To set this configuration using the CLI, type the command **set interface *interfacename* dhcp client enable**, where *interfacename* is the name of the interface you wish to configure.

Using PPPoE

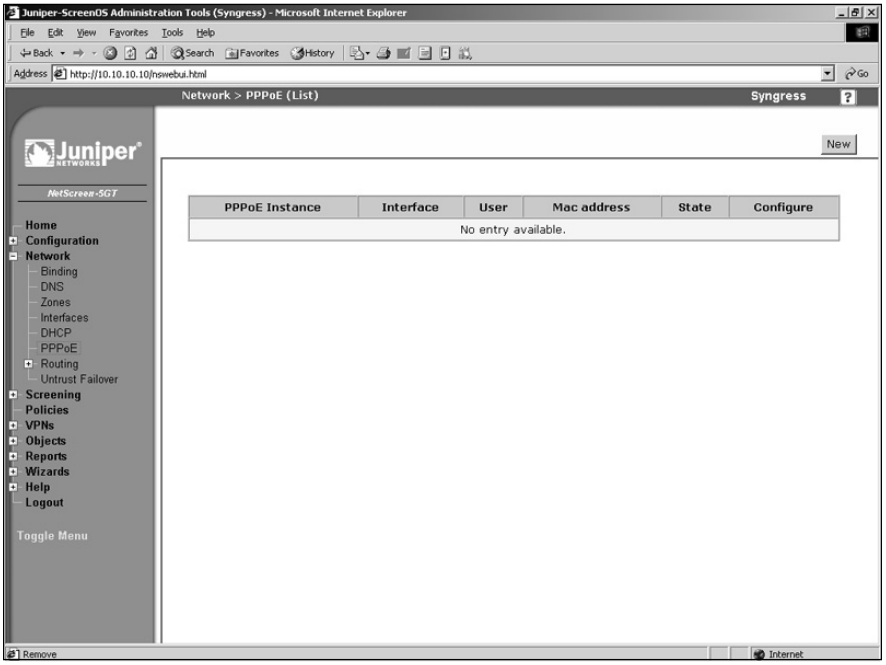
Some DSL service providers require the use of a protocol called Point-to-Point Protocol over Ethernet (PPPoE). This requires an additional configuration. You must configure a PPPoE instance, bind to an interface, and then configure the interface to use PPPoE to

negotiate the connection. You will then get an IP address from PPPoE, just as you would with DHCP.

From the WebUI:

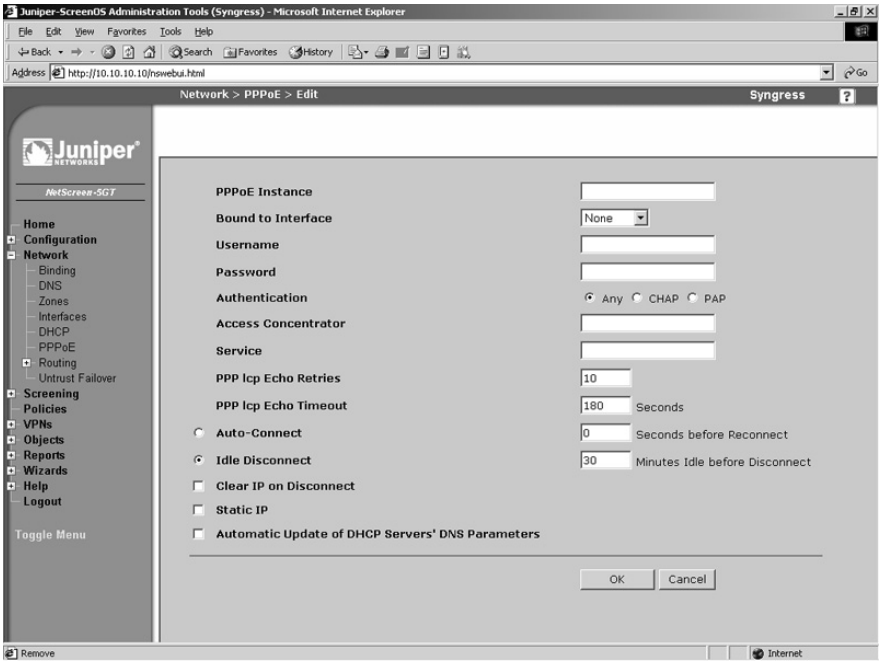
1. Access **Network** | **PPPoE**. A screen similar to Figure 3.21 will be displayed.

Figure 3.21 Network PPPoE



2. Press **New**. A screen similar to the one shown in Figure 3.22 will be displayed.
3. Use the PPPoE Instance field to enter the name.
4. Choose **untrust** from the **Bound to Interface** drop-down list, or whichever interface you wish to bind.
5. Enter your ISP-provided username and password in the **Username** and **Password** fields, respectively.
6. Select the **Any Authentication** option.
7. Enable the **Automatic Update of DHCP Servers' DNS Parameters** option.
8. Press **OK**.

Figure 3.22 Network | PPPoE | Edit



To create a PPPoE connection via the CLI, type the command **set pppoe name "*name*" username "*username*" password "*password*"**, ensuring that you include the quotation marks, and where *name* is the name of the interface, *username* is your ISP-provided username, and *password* is your ISP-provided password.

Interface Speed Modes

By default, all of the ports on your Juniper firewall are auto-sensing. This means they negotiate Ethernet settings such as speed and duplex. Regardless of the auto-sensing feature, you will want to hard code these settings to ensure that you are getting proper performance out of your network. This configuration can be done only from the CLI. In the following example, we will hard code the trust interface port 4 interface to 100Mbps full duplex.

```
Syngress-> get interface trust port phy
Port 1:  link is down, 10 Mbps, forced to half duplex
Port 2:  link is down, 10 Mbps, forced to half duplex
Port 3:  link is down, 10 Mbps, forced to half duplex
Port 4:  link is up, 100 Mbps, auto negotiated to full duplex
Syngress-> set int trust port 4 phy full 100mb
Syngress-> get interface trust port phy
Port 1:  link is down, 10 Mbps, forced to half duplex
```

```
Port 2:  link is down, 10 Mbps, forced to half duplex
Port 3:  link is down, 10 Mbps, forced to half duplex
Port 4:  link is up, 100 Mbps, forced to full duplex
```

Port Mode Configuration

Some devices in the SOHO product line support *port mode*. These devices contain one untrust, or external port, and four internal ports. By default, the four internal ports are called trust, and they are bound to the trust zone. However, there are four other modes you can use; however, the *extended* mode requires an additional license. When you change between port modes, this removes your existing configuration. If you clear your configuration by using the *unset all* command, the port mode setting will be unaffected. In Table 3.3 you can see the differences between the different modes.

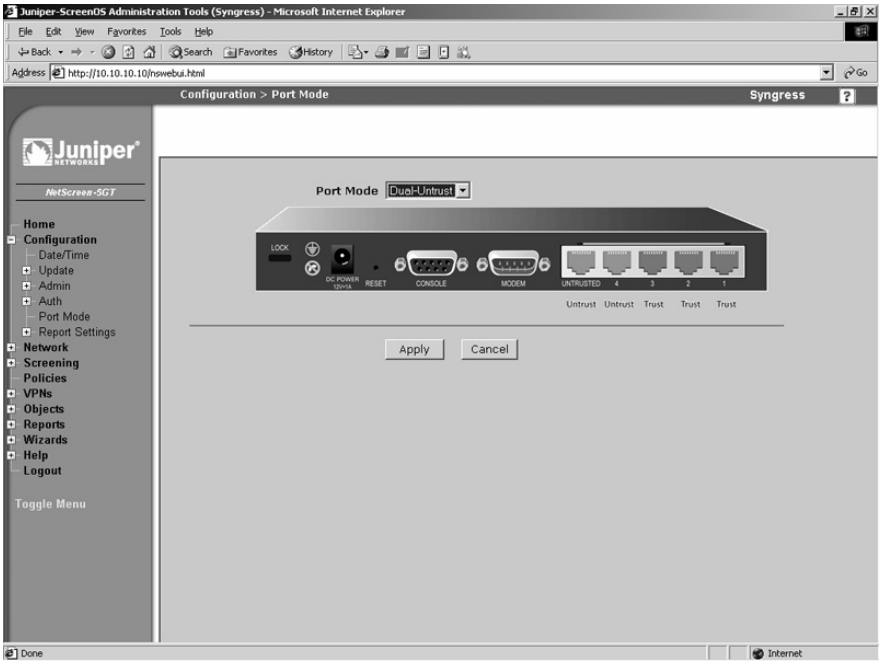
Table 3.3 Port Modes

Port	Trust-Untrust		Home-Work		Dual Untrust		Combined		Extended	
	Int	Zone	Int	Zone	Int	Zone	Int	Zone	Int	Zone
Untrusted	Untrust	Untrust	Eth3	Untrust	Eth3	Untrust	Eth4	Untrust	Eth3	Untrust
1	Trust	Trust	Eth1	Work	Eth1	Trust	Eth1	Work	Eth1	Trust
2	Trust	Trust	Eth1	Work	Eth1	Trust	Eth2	Home	Eth1	Trust
3	Trust	Trust	Eth2	Home	Eth1	Trust	Eth2	Home	Eth2	DMZ
4	Trust	Trust	Eth2	Home	Eth2	Untrust	Eth3	Untrust	Eth2	DMZ
Modem	Serial	Null	Serial	Null	Serial	N/A	N/A	N/A	Serial	Untrust

Port	DMZ-Dual-Untrust		Dual-DMZ	
	Int	Zone	Int	Zone
Untrusted	Eth4	Untrust	Eth5	Untrust
1	Eth1	Trust	Eth1	Trust
2	Eth1	Trust	Eth2	DMZ
3	Eth2	DMZ	Eth3	DMZ2
4	Eth4	Untrust	Eth4	Untrust
Modem	Serial	Null	Serial	Null

You can change the port mode settings from either the CLI or the WebUI. You can see the port mode WebUI configuration in Figure 3.23.

Figure 3.23 Port Mode Configuration



Use the following steps to change the port mode settings via the WebUI:

1. Access **Configuration | Port Mode**.
2. Select the desired mode from the **Port Mode** drop-down list.
3. Press **Apply**, then select **OK** to confirm. Your current configuration will be erased and the device will reboot.

To change modes using the CLI, type the command **exec port-mode combined** and press **y** to confirm. Your current configuration will be erased, and the device will reboot.

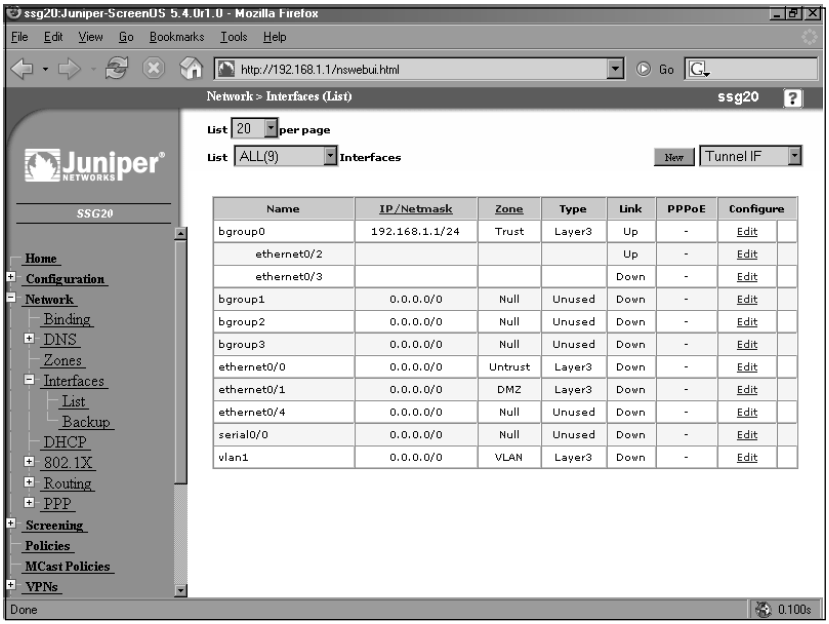
Bridge Groups

The SSG 5 and SSG 20 firewalls offer the option of configuring *bridge groups*. This new option replaces port modes. The bridge group option is more flexible because you are not subject to the limitations of the port mode. You can configure up to four bridge groups per unit. Another benefit is that you can also bind wireless interfaces to a bridge group. The wireless interfaces were previously independent of the LAN interfaces. You could not have the same IP subnet on the LAN and the Wireless LAN (WLAN) at the same. However, bridge groups allow you to bridge them together. The best part is you do not have to reboot the unit for this to take effect.

Use the following steps to change the bridge group settings via the WebUI:

1. Access **Network | Interfaces | List.**
2. Select the Bridge Group you want to edit and select **Edit** from the **Configure** column (see Figure 3.24).
3. Under the Bridge Groups **properties** section at the top select **Bind Port.**
4. You can now select and deselect the interfaces you want to add to the bridge group by selecting the checkbox to the right of the interface name.

Figure 3.24 Bridge Group Configuration



Use the following commands to add an interface to a bridge group via the CLI:

```
Syngress-> get int

A - Active, I - Inactive, U - Up, D - Down, R - Ready

Interfaces in vsys Root:
Name           IP Address          Zone        MAC             VLAN State VSD
serial0/0      0.0.0.0/0           Null        0014.f69b.e3cd    -   D    -
eth0/0         0.0.0.0/0           Untrust     0014.f69b.e3c0    -   D    -
eth0/1         0.0.0.0/0           DMZ         0014.f69b.e3c5    -   D    -
eth0/4         0.0.0.0/0           Null        0014.f69b.e3c8    -   D    -
bgroup0        192.168.1.1/24      Trust       0014.f69b.e3c9    -   U    -
```

```
  eth0/2       N/A                N/A         N/A               -   U   -
  eth0/3       N/A                N/A         N/A               -   D   -
bgroup1        0.0.0.0/0          Null        0014.f69b.e3ca    -   D   -
bgroup2        0.0.0.0/0          Null        0014.f69b.e3cb    -   D   -
bgroup3        0.0.0.0/0          Null        0014.f69b.e3cc    -   D   -
vlan1          0.0.0.0/0          VLAN        0014.f69b.e3cf    1   D   -
null           0.0.0.0/0          Null        N/A               -   U   0
Syngress-> set int bg0 port e0/4
Syngress-> get int

A - Active, I - Inactive, U - Up, D - Down, R - Ready

Interfaces in vsys Root:
Name           IP Address         Zone        MAC            VLAN State VSD
serial0/0      0.0.0.0/0          Null        0014.f69b.e3cd    -   D   -
eth0/0         0.0.0.0/0          Untrust     0014.f69b.e3c0    -   D   -
eth0/1         0.0.0.0/0          DMZ         0014.f69b.e3c5    -   D   -
bgroup0        192.168.1.1/24     Trust       0014.f69b.e3c9    -   U   -
  eth0/2       N/A                N/A         N/A               -   U   -
  eth0/3       N/A                N/A         N/A               -   D   -
  eth0/4       N/A                N/A         N/A               -   D   -
bgroup1        0.0.0.0/0          Null        0014.f69b.e3ca    -   D   -
bgroup2        0.0.0.0/0          Null        0014.f69b.e3cb    -   D   -
bgroup3        0.0.0.0/0          Null        0014.f69b.e3cc    -   D   -
vlan1          0.0.0.0/0          VLAN        0014.f69b.e3cf    1   D   -
null           0.0.0.0/0          Null        N/A               -   U   0
Syngress->
```

Bridge Group Caveats

As with any new features there are a few caveats with the bridge group feature. It is important to point these out these potential problem areas before you get tripped up by one of them:

- Policies cannot be configured between ports in a bridge group (Bgroup). Traffic is switched locally and is not processed by the ScreenOS engine
- You cannot bind Eth0 and wireless port into the same Bgroup interface. This restriction applies only to Eth0 port.
- Spanning tree is not supported. Make sure you do not create a loop in network; otherwise, there will be broadcast storm.

- Transparent mode is not supported *if* Wired and Wireless port are in the *same* Bgroup.
- ScreenOS will allow creation of a VLAN subinterface on a Bgroup while wireless port is bound to this Bgroup. However, after the VLAN subinterface is created it will include only Ethernet ports in the Bgroup. Wireless port will not be a member port.

Configuring Basic Network Routing

When you want to connect to a remote network, you must inform your firewall of its location. You do this by adding network routes to your firewall. These routes tell the firewall where the remote network can be found. In this section we will discuss adding a static route to access a remote network. We will also be adding a default route. A default route is also known as the *route of last resort*. So if a packet on a device needs to get to a location, and no other routes on the device are able to identify the next gateway to send it to, it will use the default gateway. When a system is determining which route to use, it will always use the most specific route first.

In this example we add a static route on our Juniper firewall to determine the next *hop* for the 192.168.1.0/24 network. For this example, we use only the trust-vr. Chapter 7 covers routing with virtual routers. Adding routes can be accomplished from either the WebUI or the CLI. When you add a route, there are several pieces of information you need to know beforehand:

- **Remote network** Identify the remote network route that you want to add. In our example we will be using 192.168.1.0/24. You can also add routes for single hosts such as 192.168.1.20/32.
- **Interface or virtual router** The interface is whichever physical interface on which the gateway is located.
- **Next hop gateway** You need to know which system can take your packets to the specified remote network. This device must be capable of connecting to the remote network, or if not, it must know where the remote network can be located.
- **Metric** The metric is a preference number, with the lowest number having the highest priority. All directly connected networks have a metric of zero. All static routes have a default metric of one. There may be cases in which you need to add the same route twice, the preferred route with the lower metric and the less preferred route with the higher metric. If the first route is unavailable, the firewall will use the next route.

Our first example of adding a static route in the WebUI.

1. Access **Network | Routing | Routing Entries**. A screen similar to the one shown in Figure 3.25 will be displayed.

Figure 3.25 Routing Entries

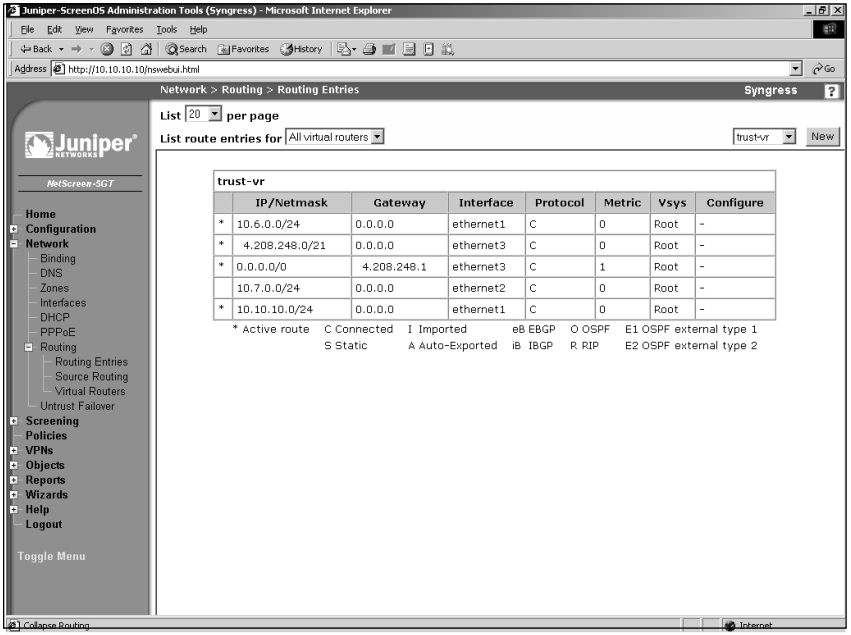
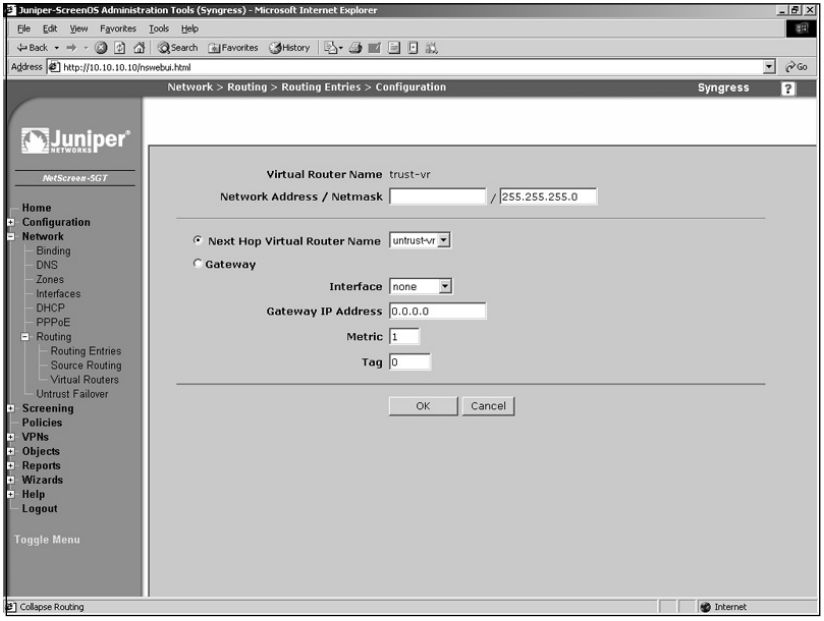


Figure 3.26 Configure a Routing Entry



2. Use the drop-down list in the upper right-hand corner to select the virtual router and select **New**. In our example, we will select **trust-vr**. A screen similar to the one shown in Figure 3.26 will be displayed.
3. Enter the **Network Address/Netmask**.
4. Select the **Gateway** option.
5. Use the **Interface** drop-down list to select the interface (gateway) that is the next hop and use the **Gateway IP Address** field to enter the gateway's IP address.
6. Press **OK**.

To add a static route using the CLI, type the command **set route *ipaddress/netmask* interface *interfacename* gateway *gatewayip***, where *ipaddress* is the virtual router's IP address, *netmask* is the virtual router's netmask, *interfacename* is the next hop gateway, and *gatewayip* is the IP address of the next hop gateway.

To remove a static route via the WebUI, access **Network** | **Routing** | **Routing Entries** and select the **Remove** link of the route you wish to delete. Press **OK** to confirm.

The most important and most used route on a firewall is the default route, or route of last resort. This route is used when no other route matches the traffic. Typically this route will point to your Internet router. If you are running either DHCP or PPPoE, your default route will likely come from that source. However, there may be times when you need to add your own default route. This can be done from either the WebU or the CLI. It is much like adding a static route.

From the WebUI:

1. Access **Network** | **Routing** | **Routing Entries**.
2. Select your virtual router from the drop-down list in the upper right-hand corner and select **New**.
3. Enter **0.0.0.0** in the **Network Address** field and type **0** in the **Netmask** field.
4. Select the **Gateway** option.
5. Use the **Interface** drop-down list to select the interface that acts as the next hop gateway and enter the **Gateway IP Address**.
6. Press **OK**.

To remove the static route using the CLI, type the command **set route 0.0.0.0/0 interface *interfacename* gateway *gatewayip***, where *interfacename* is the next hop gateway and *gatewayip* is the gateway's IP address.

Configuring System Services

On your Juniper firewall there are some other notable devices to configure. Configuring the time is very important for being able to correlate information in the logs to a specific time;

therefore, configuration of the local clock is critical. Also, the firewall executes specific events at given times. If the time is configured improperly, this can prevent events from occurring at the correct time.

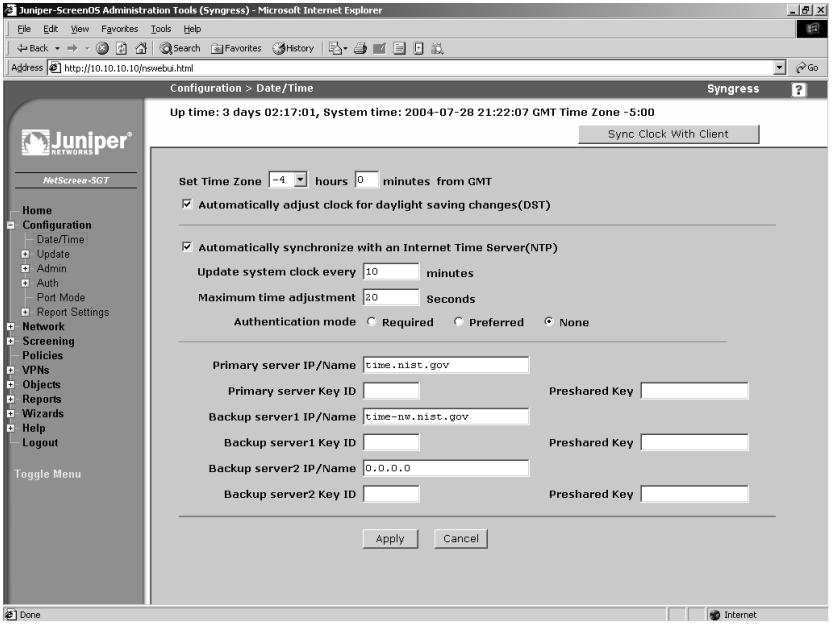
Most Juniper firewalls contain a built-in DHCP server. Typically, you can have a server on each interface. This allows you to manage your internal IP addressing from a single location. All Juniper firewalls are able to query DNS servers. This allows them to resolve hostnames to IP addresses. It is important to have working DNS servers configured on your firewall in case you want to use the network to synchronize time to an NTP server.

There is a great deal of information generated by your firewall in the form of logs. Because all Juniper firewalls have limited space for storing logs, you may want to be able to send this logging information to a remote system. We will look at how to configure, and use, two separate remote log repositories. Finally, we will look at license keys. These keys unlock the features of your firewall device. We will investigate how license keys work and how to update your license key.

Setting the Time

Every Juniper device contains an internal clock that runs continually while the device is powered on. You can manually set the clock from either the WebUI or the CLI. Ideally, you would configure your firewall to contact a timeserver using the Network Time Protocol (NTP) To ensure that the clock is set to the correct time, the steps shown in Figure 3.27 replicate the time configuration page from the WebUI, and it is used to manually set the time on your firewall,.

Figure 3.27 Date/Time Configuration



From the WebUI:

1. Access **Configuration** | **Date/Time**.
2. Use the **Set Time Zone** fields to specify the difference between your time zone and GMT (Greenwich Mean Time).
3. Enable the **Automatically adjust clock for daylight saving changes (DST)** option.
4. Press **Apply**.
5. Press the **Sync Clock with Client** button and select **Yes** to confirm.

```
To set the timezone and date/time using CLI, type the following commands:

Syngress-> set clock timezone vv
Syngress-> set clock MM/DD/YYYY hh:mm:ss
```

Where vv in the first command is the difference between local time and Greenwich Mean Time (GMT) (expressed as + or -, for example, +3 or –5), and where MM/DD/YYYY is the month, date, and year, and e hh:mm:ss is the hour, minute, and second.

Setting up timeservers to sync with the NTP protocol allows up to subsecond accuracy for time synchronization. NTP is a free service, and every system should use it. The only time you should not use it is when you want the firewall to generate no traffic. NTP can be configured from either the CLI, or the WebUI. However, you can force NTP synchronization only from the CLI. Figure 3.26 shows the time screen that contains the NTP settings.

From the WebUI:

1. Access **Configuration** | **Date/Time**.
2. Enable the **Automatically synchronize with an Internet Time Server (NTP)** option.
3. Enter **time.nist.gov** in the **Primary server IP/Name** field.
4. Enter **time-nw.nist.gov** in the **Backup server1 IP/Name** field.
5. Press **Apply**.

To synchronize the time via the CLI, type the following commands:

```
Syngress-> set ntp timezone -5
Syngress-> set ntp server time.nist.gov
Syngress-> set ntp server backup1 time-nw.nist.gov
Syngress-> set clock ntp
Syngress-> exec ntp update
```

When asked if you want to update the system clock, press **y** for yes.

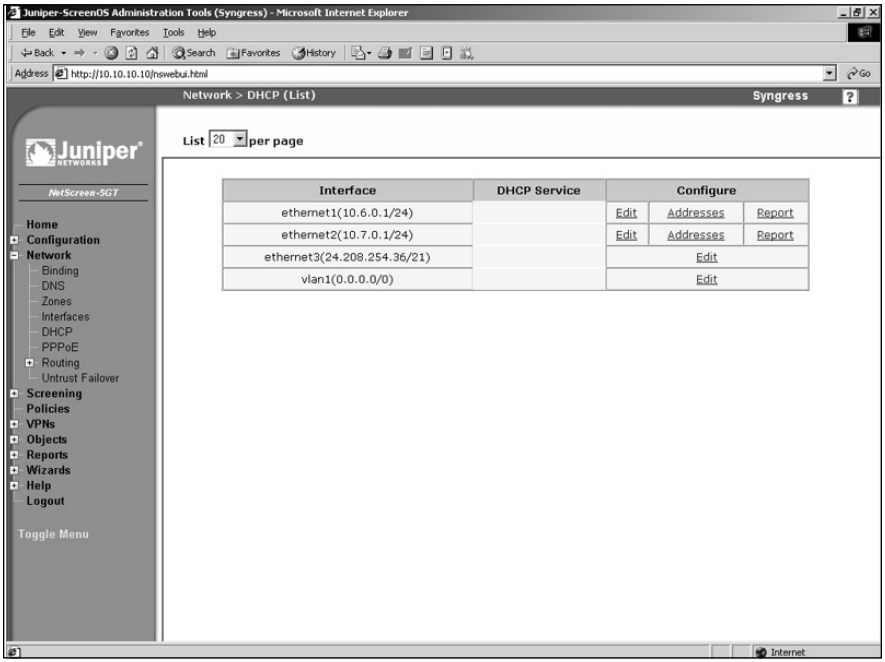
Finally, you can use Secure Network TimeProtocol (SNTP). This provides MD5-based authentication of each packet to ensure that the packet is from the specified server. To use

authentication, you must assign a key ID, and a preshared key for every timeserver you configure. Additionally, you must configure whether authenticaton is required, or simply preferred.

DHCP Server

Juniper firewall devices can act as a DHCP server to allow the firewall to control IP address allocation on the network. Any Juniper device is capable of hosting up to eight DHCP servers. The server can assign IP addresses from a pool, or from a reserved list based on MAC address. Another feature of the DHCP server on the Juniper firewall is that it can determine whether another DHCP server is running on the network. This can prevent a conflict between two servers concurrently handing out IP addresses. In our example, we will set up a DHCP server on the Eth2 interface of a Juniper-5GT. We will assign a pool of IP addresses as shown in Figure 3.28, and create one reservation based upon MAC address. DHCP servers can be configured from either the WebUI or the CLI.

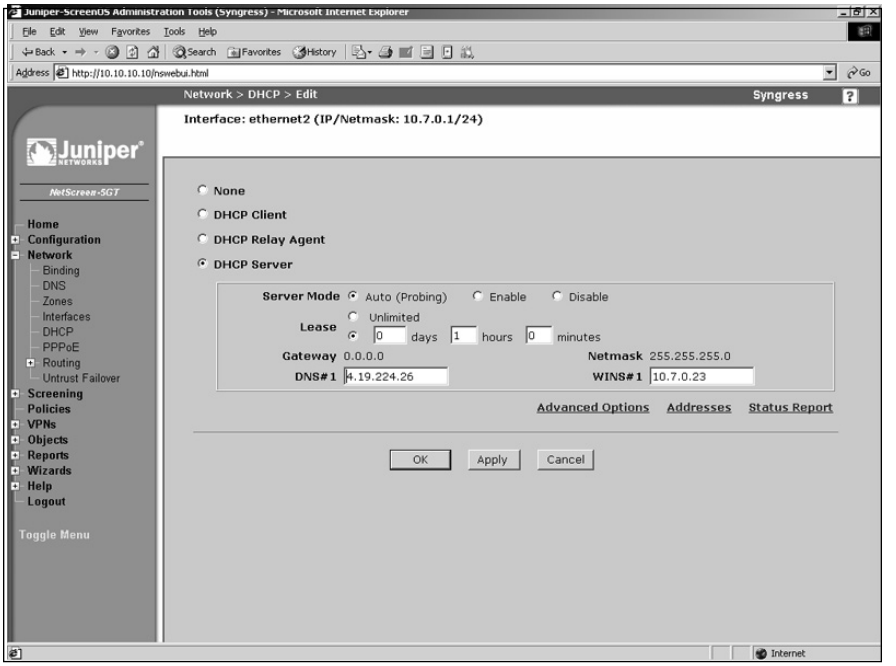
Figure 3.28 DHCP List



From the WebUI:

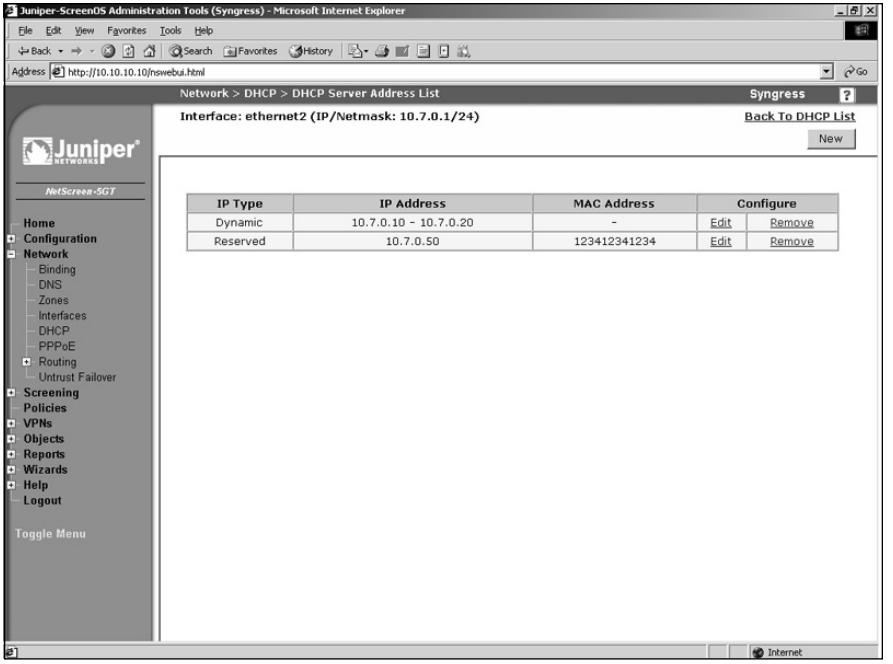
1. Access **Network** | **DHCP**.
2. Locate the Ethernet2 interface and select its **Edit** link. A screen similar to the one shown in Figure 3.29 will be displayed.

Figure 3.29 Edit a DHCP Entry



3. Enable the **DHCP Server** option.
4. For **Server Mode**, enable the **Auto (Probing)** option.
5. In the **Lease** section, select the option button that allows you to enter a specific time, and then enter the desired **days**, **hours**, and **minutes**.
6. Use the **DNS#1** field to enter the IP address of the primary DNS server.
7. Use the **WINS#1** field to enter the IP address of the primary WINS server.
8. Press **OK**. The DHCP list will be displayed.
9. Locate the ethernet2 interface in the list and select its **Addresses** link. A screen similar to the one shown in Figure 3.30 will be displayed.
10. Press **New**.
11. Ensure the **Dynamic** option is selected.
12. Use the **IP Address Start** field to enter the first IP address in the address pool.
13. Use the **IP Address End** field to enter the last IP address in the address pool.
14. Press **OK**. The DHCP Server Address List screen will be displayed.
15. Press **New**.
16. Select the **Reserved** option.

Figure 3.30 DHCP Server Address List



17. Use the **IP Address** field to enter the IP address that you wish to reserve.
18. Use the **Ethernet Address** field to enter the MAC address of the device for which you wish to reserve the specified IP address.
19. Press **OK**.

Use the following commands to configure the DHCP server via the CLI:

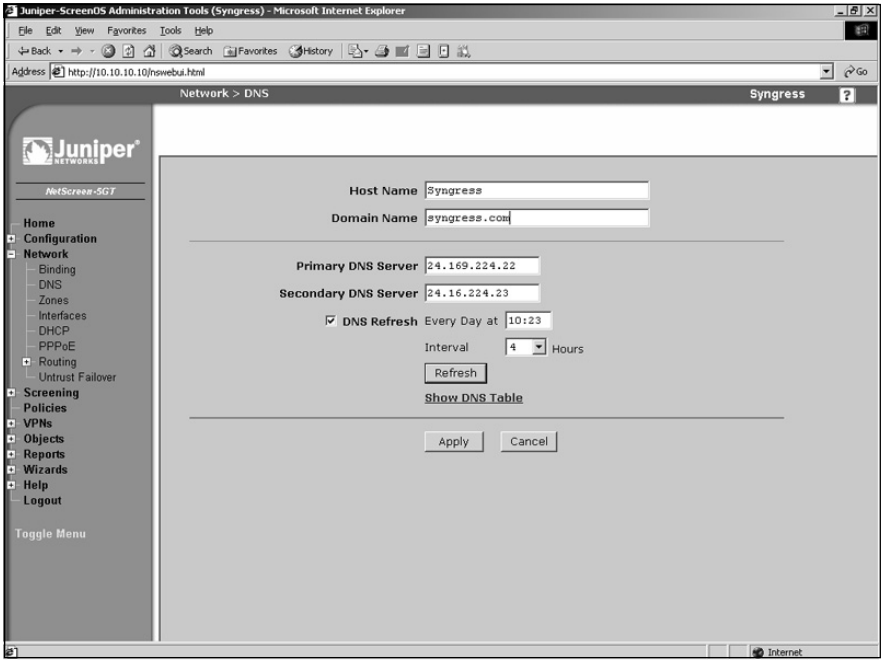
```
Syngress-> set interface ethernet2 dhcp server auto
Syngress-> set interface ethernet2 dhcp server enable
Syngress-> set interface ethernet2 dhcp server option lease 60
Syngress-> set interface ethernet2 dhcp server option dns1 10.7.0.23
Syngress-> set interface ethernet2 dhcp server option wins1 10.7.0.23
Syngress-> set interface ethernet2 dhcp server option netmask 255.255.255.0
Syngress-> set interface ethernet2 dhcp server ip 10.7.0.10 to 10.7.0.20
Syngress-> set interface ethernet2 dhcp server ip 10.7.0.50 mac 123412341234
```

DNS

Setting up your Juniper firewall as a DNS client is relatively simple. The firewall keeps a local cache of DNS entries, and you must decide when you want the cache to be cleared. DNS

can be configured from either the WebUI or the CLI. Figure 3.31 shows the WebUI screen for configuring DNS. The hostname and domain name are also set from this page. If you are using a DHCP, or PPPoE, client on your firewall, the DNS server settings and domain name may be passed down and configured for you.

Figure 3.31 DNS Configuration



From the WebUI:

1. Access **Network** | **DNS**.
2. Enter a **Host Name** and a **Domain Name**.
3. Enter the IP address of the **Primary DNS Server** and the **Secondary DNS Server**.
4. Enable the **DNS Refresh** option and enter the refresh time and frequency.
5. Press **Apply**.

Enter the following commands to configure the DNS server via the CLI:

```
Syngress->set hostname Syngress
Syngress-> set domain syngress.com
Syngress-> set dns host dns1 2.32.23.23
Syngress-> set dns host dns2 2.32.23.24
Syngress-> set dns host schedule 10:23 interval 4
```

SNMP

Simple Network Management Protocol (SNMP) allows remote administrators to view data statistics on a Juniper device. It also allows a Juniper device to send information to a central server. Juniper firewalls support SNMPv1 and SNMPv2c. It also supports the Management Information Base two (MIB II), or standard groups. The SNMP agent supports sending the following traps.

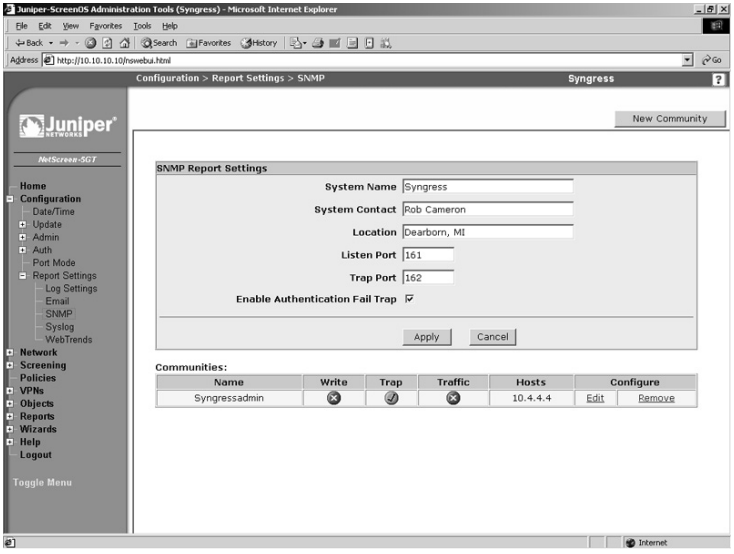
- Cold Start Trap
- Trap for SNMP Authentication Failure
- Traps for System Alarms
- Traps for Traffic Alarms

By default, the SNMP manager requires no configuration. This prevents unauthorized viewing of the system based upon default parameters. To configure your Juniper device for SNMP, you must configure community strings, SNMP host addresses, and permissions. In our configuration example, we will first set up the basic system information, and then we will create a new community. This can be done from either the WebUI or the CLI. You can create up to three communities, with up to eight IP ranges in each. An IP range can consist of a single host, or a network. If you configure a network, those defined IP addresses can poll only the device.

Use the following steps to configure SNMP via the WebUI:

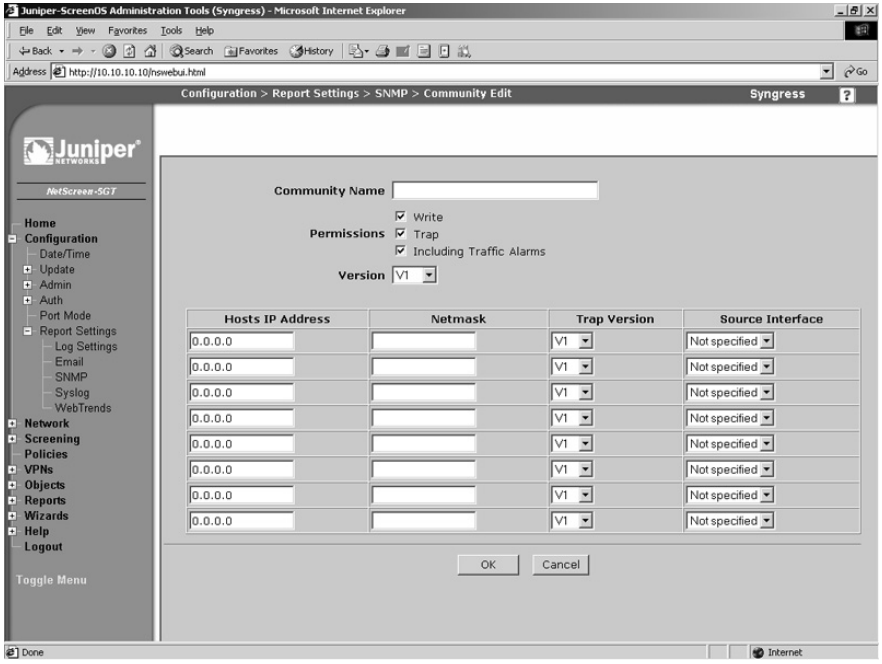
1. Access **Configuration** | **Report Settings** | **SNMP**. A screen similar to the one shown in Figure 3.32 will be displayed.

Figure 3.32 SNMP Report Settings



2. Enter the desired **System Name**, **System Contact**, and **Location**.
3. Enter the port numbers of the **Listen Port** and the **Trap Port**.
4. Ensure that the **Enable Authentication Fail Trap** option is enabled.
5. Press **Apply**.
6. Press **New Community**. A screen similar to the one shown in Figure 3.33 will be displayed.

Figure 3.33 New Community



7. Enter a **Community Name**.
8. Enable the **Write** option if you want to allow the remote SNMP user to modify this configuration.
9. Enable the **Trap** option to allow the SNMP agent to send traps to the defined hosts.
10. Enable the **Including Traffic Alarms** option if you wish to force the local SNMP agent to send traffic alarms to the defined hosts.
11. Use the **Version** drop-down list to select the SNMP version that this community will support. The **Any** option will cause the community to support both the v1 and v2c versions.

12. You must define at lease one host, or network, in the lower portion of the screen. To do so, enter the **Host's IP Address** and **Netmask**. Next, select the **Trap Version,** and use the **Source Interface** drop-down list to select the SNMP interface.
13. Press **OK**.

To remove a community, locate it in the community list and select its **Remove** link. Press **OK** to confirm.

To configure SNMP via the CLI, type the following commands:

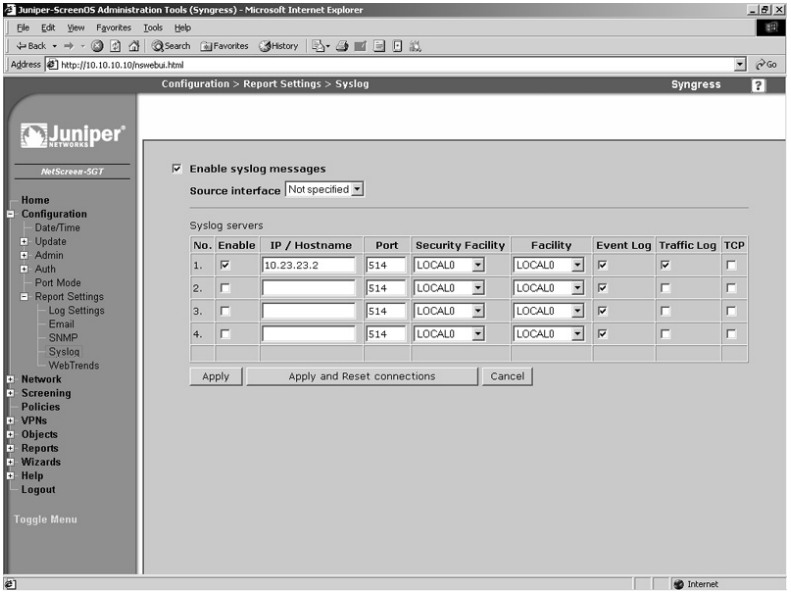
```
Syngress-> set snmp name Syngress
Syngress-> set snmp location "Dearborn, MI"
Syngress-> set snmp community Syngressadmin Read-Only version v2c
Syngress-> set snmp host Syngressadmin 10.4.4.4
```

Syslog

Juniper firewalls generate a great deal of logging. Logged information is contained on the local flash file system using the first-in, first-out (FIFO) method. The first log in will be the first log removed when logging space fills to the limit. If you want to keep your logs for an extended period of time, you must archive them to an external log server. A Juniper firewall can concurrently send information to up to four syslog hosts. Syslog can be configured from either the WebUI or the CLI. Logging is discussed in depth in the next chapter.

Use the following steps to configure the syslog server via the WebUI:

Figure 3.34 Syslog Configuration



1. Access **Configuration** | **Report Settings** | **Syslog**. A screen similar to the one shown in Figure 3.34 will be displayed.
2. Enable the **Enable syslog messages** option.
3. Use the **Source interface** drop-down list to specify the interface from which messages will be sent. If you do not specify an interface here, messages will be sent from the interface closest to the syslog host.
4. In the row labeled **No. 1**, enable the **Enable** checkbox, and type the **IP/Hostname** and **Port** of the remote syslog server.
5. Use the **Security Facility** drop-down list to select the syslog facility to which emergency and critical messages will be sent.
6. Use the **Facility** drop-down list to select the syslog facility to which all other messages will be sent.
7. Enable the **Event Log**, **Traffic Log**, and **TCP** options.
8. Press **Apply**. If you are updating an existing syslog configuration, select **Apply and Reset connections**.

Enter the following commands to configure syslog via the CLI:

```
Syngress-> set syslog config 10.23.23.2 facilities local0 local0
Syngress-> set syslog config 10.23.23.2 port 514
Syngress-> set syslog config 10.23.23.2 log all
Syngress-> set syslog enable
```

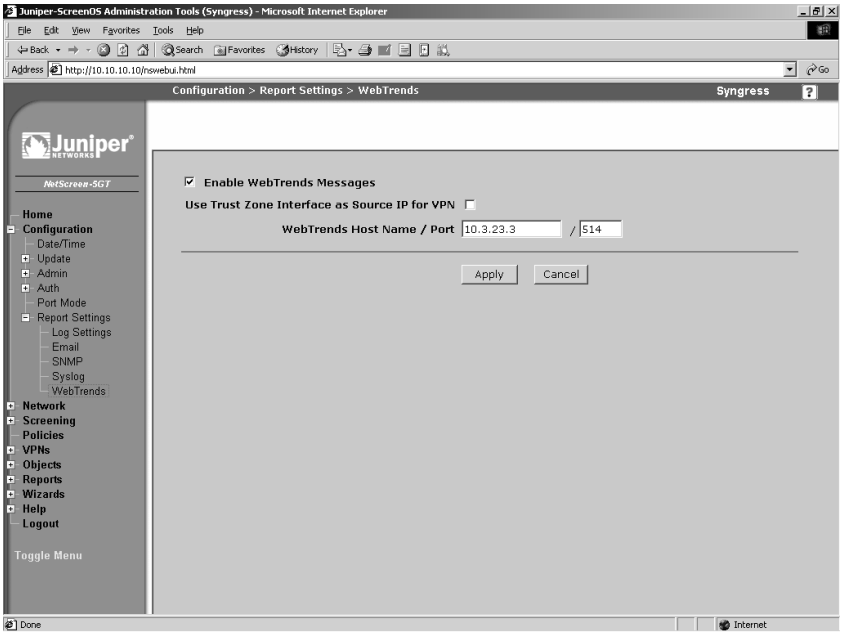
Web Trends

WebTrends firewall suite is a product from the company NetIQ. It is a syslog server that collects all logs, and allows also you to generate graphical reports from the logs. A remote WebTrends server can be configured either from the CLI or the WebUI.

Use the following steps to configure WebTrends via the WebUI:

1. Access **Configuration** | **Report Settings** | **WebTrends**. A screen similar to the one shown in Figure 3.35 will be displayed.
2. Enable the **Enable WebTrends Messages** option.
3. Enter the IP address and port number in the **WebTrends Host Name / Port** fields.
4. Press **Apply**.

Figure 3.35 Web Trends Configuration



Enter the following commands to configure WebTrends via the CLI:

```
Syngress-> set webtrends host-name 10.3.23.3
Syngress-> set webtrends port 514
Syngress-> set webtrends enable
```

Resources

Windows SSH client *PuTTY*: www.chiark.greenend.org.uk/~sgtatham/PuTTY/
Windows TFTP server Pumpkin: http://kin.klever.net/pumpkin/binaries
Windows Serial/telnet Client Tera Term: http://hp.vector.co.jp/authors/VA002416/teraterm.html

Summary

In this chapter we covered a great deal of information. The purpose of this chapter was to familiarize you with the initial configuration of a Juniper firewall. Before using your firewall, you must understand how to manage it. We explored various methods to manage your firewall. It is important to understand each option available to you. Each option is a separate tool that can be used to control your firewall.

There are two core types of remote management, the WebUI and the CLI. If you are using the serial console, Telnet, or secure shell, you are using the CLI. It is important to be proficient in both management tools. The WebUI is initially easier to use. However, in later chapters you will see that advanced troubleshooting techniques can be carried out only from the command line interface. These techniques are invaluable for more advanced configurations. We also mentioned a third type of management, NetScreen SecurityManager. The NetScreen SecurityManager product is an external source of management, covered in a later chapter.

We also discussed configuring your Juniper firewall to run on the network. Zones are a core part of the Juniper security infrastructure. The security zone is the most commonly used zone, and it is used on every interface, and in every policy. Each interface must be bound to a zone. In the next chapter we focus on basic policy creation and policy theory. We looked at the various types of interface that the firewall supports. The physical interface will be used on each type of Juniper device to interact with the network. The firewall can operate in two modes, layer two and layer three. In this chapter, we focused on layer three configuration of the device. In a later chapter, we focus on the layer two mode, transparent mode.

In the last section of the chapter, we discussed configuration of various system components. Configuring the time on your device is critical, because time is the central reference point used to correlate all events on the firewall. If someone were to break into your network, and your logs were off by several hours, or days, this could mislead your investigation of the break-in. Configuring your logs to be sent to a separate location is also important if you intend to keep your logs on a long-term basis. The syslog server and WebTrends server are powerful options. If you use NetScreen SecurityManager, it also can be used as a central log repository.

Solutions Fast Track

Managing the Juniper Firewall

☑ There are two methods to manage your firewall, the WebUI and the CLI.

- ☑ Configuration rollback is an important tool for saving a working configuration before you implement changes on your firewall that could potentially disrupt your firewall.
- ☑ If you use the WebUI the configuration still ends up in the CLI. It is a good idea to memorize the CLI commands because they are at the root of the configuration.

Configuring Your Firewall for the First Time

- ☑ Security zones are used to identify a logical area of your network.
- ☑ Physical interfaces can host multiple IP addresses on each interface.
- ☑ Loopback interfaces are always up when they are configured, and they must be bound to a zone the same as physical interfaces.

Configuring System Services

- ☑ Setting the system clock is crucial because it is your central point of reference for events that occur on the firewall.
- ☑ If you need to resolve hostnames to IP addresses, you must configure DNS servers.
- ☑ A Juniper firewall can hold only so many logs locally, so you must configure an external log server if you want to archive your logs.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: Why does Juniper use zones on interfaces? I have used this type of configuration on other devices, and I did not find it to be very effective.

A: Zones are designed to segment areas of the network from each other. On a Juniper firewall, using security zones during policy creation allows, or disallows, traffic from one zone to another. This simplifies policy creation by specifying which zone traffic can travel from and to. Furthermore, it removes the chance of an accidental configuration of access from one system to another. This can easily happen if you use a firewall that does not support zones.

Q: You cover securing the management interface extensively. Are all of those options really required?

A: Because the firewall is such a critical part of your network, you need to ensure its security well. Each option may be used in your network, or perhaps a combination of all of the options makes the most sense in your environment. By understanding all options, you will have the ability to pick and choose between them.

Q: I have looked at the command line interface, and I do not feel that it is very effective to use. Why should I use it when the WebUI is easier and quicker?

A: The WebUI is a very useful tool, and it should be used in conjunction with the CLI. Both have pros and cons. In later chapters, you will need to be proficient in using the command line interface and to be comfortable with its options. Even if you choose to use the WebUI most of the time, I encourage you to use the CLI from time to time so that you are comfortable when you have the need to access it.

Q: You have talked about several options like transparent mode and NetScreen Security Manager. Why did you give so few details on these?

A: These options are complex, and they each deserve separate discussion. There are dedicated chapters for each topic that examine each option in depth.

Chapter 4

Policy Configuration

Solutions in this chapter:

- Firewall Policies
- Policy Components
- Creating Policies

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

In the Juniper firewall, a policy is the core component of access control. In this section, we will explore the basic principles of a policy and how to create them. All firewall devices use some sort of statement that provides access control between two segments of a network. Each product implements access control differently. If you have experience with any firewall product, then Juniper policies should come easy for you. If you have never had the opportunity to create a network access control policy before, this section will help you understand the principles of access control as well as the methodology of creating a policy.

First, we will look at the definition of a policy and what creating one really means. We will also look into the theory of access control and specific methodologies behind allowing or denying access to network resources. In the second part of this section, we will review what makes up a policy on a Juniper firewall. Every policy must have several basic components defined before it can be created. We will look at each component and how to create them on your firewall.

Much like building a house, Juniper firewalls use different components to build policies. Several components are required for a policy. In this chapter, we will look at these components and how to create them for use in a policy. Components can be created via the Web user interface (WebUI) or the command-line interface (CLI). Each method generates the same result, but the process is different. As discussed throughout this book, becoming familiar with both methods will help you better understand the Juniper firewall platform.

In the final section of this chapter, we will take the components we created and use them to form policies. For the first time in this book, we will look at the WebUI and CLI separately, because the methods differ enough that each requires separate attention.

Firewall Policies

A policy permits, denies, or tunnels specified types of traffic between two points. That is the official definition of a firewall policy according to Juniper Networks. Let's look deeper into that definition. A policy is a single statement defining whether a resource can be accessed, and by whom. On a Cisco PIX or router, a policy is the equivalent to a conduit or access list. On a Check Point firewall, a policy is the equivalent of a firewall rule.

A policy does not reference a complete list of rules or the entire embodiment of the access control statements. Nor is a policy referenced as any sort of written statement in this case. A policy is a single access control statement. Every policy has the following five basic elements:

- **Direction** The first element is direction. The direction is based upon security zones. You must define two security zones in each policy. The first security zone must be the source of the traffic you want to access a specific resource. The second zone is the destination zone. The destination zone is where the destination system or network is located.

- **Source Address** The next component of a policy is the source addresses. This component defines the source Internet Protocol (IP) address of the source hosts. These hosts must be in the source zone as well. These source IP addresses can access the destination hosts in the destination zone. At a maximum, you may use *Any* as the source; this specifies any IP address in the source zone.
- **Destination Address** The destination hosts are the hosts that the source addresses will attempt to access. The destination hosts must be in the destination zone. Destination addresses must have a minimum of a single host. At a maximum, you may use *Any* as the destination; this specifies any IP address in the destination zone.
- **Service** When you define a service, you define which application you want the source address to access. Defining this is based upon both port and protocol. You can allow ICMP (Internet Control Message Protocol), TCP (Transmission Control Protocol), IP, and UDP (User Datagram Protocol) protocols. Each predefined service has these protocols specified. Much like the source and destination address, you can also specify *Any* for the service; this will allow any protocol using any port from the source address to access the destination address through the firewall.
- **Action** The specified action is what you want to happen to the traffic that matches the specified policy. Four actions can be imposed on traffic that matches the policy. The first action is *permit*. When specifying permit as your action, you are allowing the matching traffic to pass through the firewall. The second action you can select is *deny*. This action denies and drops the traffic if it matches the defined policy. The third action is *reject*. The reject action allows you to send a network message back to the originating host. If the connection is a TCP connection, then the firewall sends back a TCP reset. If the connection is ICMP or UDP, it sends an ICMP port unreachable message. The last action you can specify is *tunnel*. Tunnel first inherently permits the traffic that is specified by the policy. However, this traffic is only permitted to pass through the specified VPN (virtual private network) tunnel. If you use the action of tunnel, you must specify a VPN tunnel as well.

Additional items can also be defined for each policy. These additional items include logging, Network Address Translation (NAT), traffic shaping, counting, traffic alarms, antivirus scanning, scheduling, URL (Uniform Resource Locator) filtering, and user authentication.

- **Logging** Logging is an essential tool for both troubleshooting and recording who has accessed your network. We will look at logging in more detail later in this chapter.
- **NAT** NAT allows you to hide your internal IP addresses. It is used in almost every internal network. NAT has many intricacies on the Juniper platform, so it's given its own chapter in this book.

- **Traffic Shaping** Traffic shaping allows you to control the amount of bandwidth certain traffic can consume. (Traffic shaping is covered in more detail in Chapter 5.)
- **Counting** When you turn on counting for a policy, the Juniper firewall creates graphs for the traffic that has passed through the policy. These graphs are displayed in bytes and are useful in determining how much traffic has passed through an interface. (Configuring counting is covered in more detail in Chapter 5.)
- **Traffic Alarms** Traffic alarms allow you to generate an alert when a specific number of bytes per second or bytes per minute are exceeded. To use traffic alarms, you must enable counting. (Configuring traffic alarms is covered in more detail in the Chapter 5.)
- **Antivirus Scanning** Using antivirus scanning on your firewall allows you to scan traffic for viruses as the traffic passes through your firewall. (Configuring antivirus scanning is covered in Chapter 10.) Juniper currently uses the Kaspersky Anti-Virus engine in its products. Included in the Kaspersky engine is the capability to scan for Spyware and Adware as well.
- **Scheduling** Configuring scheduling for a policy allows you to create a policy that is in effect only at specific times. This permits you to create a policy that allows your users to browse the Internet only during specified hours. Scheduling is a powerful tool that keeps you from having to enable and disable access at specific times. (Scheduling is explained in full detail in Chapter 5.)
- **URL Filtering** There are times you may want to allow a user to access the Internet, but require some method of limiting access to appropriate Web sites. URL filtering allows you to allow or deny access to Web sites based upon their content. Juniper has teamed up with Websense and Surf Control for URL filtering in the current Screen OS release. However, you can use an embedded version of Surf Control. This lets you filter the Web browsing of your users without having a central server. (Using URL filtering is covered in Chapter 5.)
- **Anti-Spam** On some firewall products, you can configure the device to provide spam filtering. This functions when SMTP traffic is traveling inbound only, and is powered by Symantec's Bright Mail engine.
- **User Authentication** User authentication allows you to require authentication to the Juniper firewall before accessing specified resources in a policy. User authentication and using authentication servers are large subjects, requiring their own chapter. (Chapter 6 covers this subject in its entirety.)

Theory of Access Control

The theory of access control is quite simple: allow access to the required resources and deny everything else. On a Juniper firewall, everything is denied by default unless specifically

allowed. This makes creating your access control policies a straightforward process. If you want a resource to be accessed by another system, create a policy to allow access to it. If you do not want access allowed to a system, do not create a policy.

Now that you understand the beginnings of access control on a Juniper firewall, there are a couple different ideas to add into this mix when creating a policy. When traffic passes through the firewall, policies are checked in a top-down order, so the policy at the top will be checked first and then the second policy in order will be checked and so on. The best thing you can do is to create more specific policies at the top of your policy list and less specific policies as you go down the list.

Let's look at an example. Figure 4.1 shows an example of policy ordering. This is a screenshot of a Juniper policy, showing three policies. In the first policy, you see the source is very specific with only one host (WebMaster) connecting to a single destination (WebServer). This is the most specific policy in this example. The first policy only allows one single system to connect to another single system. In the second policy, any host can connect to the destination WebServer with only HTTP (Hypertext Transfer Protocol). This is a less specific policy as it allows literally any host to connect to WebServer, as long as it uses the proper protocol. In the last policy, any host can connect to the destination "FTP Servers" with the File Transfer Protocol (FTP). This is the least specific policy as it allows any host to connect to the group of FTP servers.

Figure 4.1 Policy Ordering

From Untrust To work, total policy: 3

ID	Source	Destination	Service	Action	Options	Configure			Enable	Move
7	Webmaster	WebServer	FTP HTTP HTTPS SSH	✓		Edit	Clone	Remove	☑	
6	Any	WebServer	HTTP	✓		Edit	Clone	Remove	☑	
8	Any	FTP Servers	FTP	✓		Edit	Clone	Remove	☑	

Why does the idea of most specific to least specific matter so much? Let's switch around policy number 6 and number 7 in the example in Figure 4.1. The top-most policy is now 6, while 7 is the second policy down. If we were to do this, all connections from WebMaster with the HTTP would be logged to policy number 6. This could create havoc when attempting to troubleshoot, or for long-term purposes of logging. The ID or identification for a policy is automatically generated when you create a policy from the WebUI. When creating a policy from the CLI, you get the option of setting the ID if you want or allowing the firewall to choose the next available number.

The last component of access control we need to look at is zones. Zones identify the direction a policy works in. Every policy requires a source zone and a destination zone. The source zone is the location from which the source traffic is originating. The destination zone is where the destination traffic is going. When creating a policy, you must choose both a

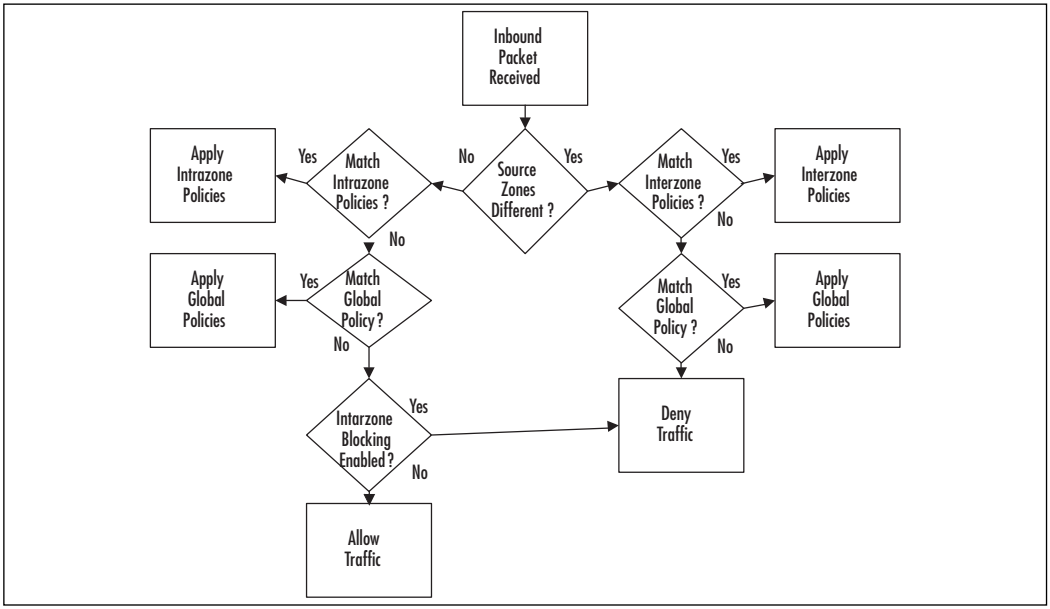
source and destination zone. The "Creating Policies" section later in this chapter discusses how to determine which components you need to create a policy.

Types of Juniper Policies

On a Juniper firewall, there are three different types of policies, each policy containing the same five core components. The only difference is the zones that the policy contains. A policy is classified by which source and destination zones are used in the policy. If you look at Figure 4.2, you will see a diagram representing the policy checking order. Before studying the diagram, let's briefly define the three types of policies:

- **Intrazone policies** An intrazone policy is a policy in which the source and destination zones are the same.
- **Interzone policies** An interzone policy is a policy in which the source and destination zones are different.
- **Global policies** A global policy is a policy in which the source and destination zones are both in the global zone.

Figure 4.2 Policy Checking Order



Intrazone Policies

There will be times on your Juniper firewall when you have multiple interfaces bound to the same zone. By default, traffic within the same zone is not blocked. You have the option of blocking intrazone traffic just as if the traffic was interzone traffic. If you do not enable intrazone blocking, all intrazone traffic is allowed. Use the following command to determine the current zone blocking state, where *zonename* is the name of the zone:

```
get zone zonename
```

The status will be listed on the "Intra-zone block" line. For example, the following output indicates that intrazone blocking is *not* enabled:

```
Intra-zone block: Off, attrib: Non-shared, flag:0x0008
```

To enable intrazone blocking, use the following command, where *zonename* is the name of the desired zone:

```
set zone zonename block
```

To disable intrazone blocking, use the following command, where *zonename* is the name of the desired zone:

```
unset zone home block
```

Interzone Policies

An interzone policy (in which the source and destination are in different zones) is the most common type of policy you are going to encounter. No configuration changes can be made to change the behavior of interzone policies.

Global Policies

A global policy is a policy in which the source and destination zone are in the global zone. The determination to use the global policy occurs in one of two situations. The first case is in which traffic has already gone through your interzone or intrazone policy list. So if the source zone is trust, and the destination zone is untrust, the flow will be checked against all the policies in the global zone that match that zone pairing. If the source IP, destination IP, and service also don't match, then the flow is checked against the global policy list. In Figure 4.2, you can see where global policies fall in the policy checking order. Global policies are very useful when you want to allow or deny a specific type of traffic regardless of the type of zone. For example, if you want to allow all zones to be able to get out and browse the Internet with HTTP traffic, but you only want to make one policy, you can do so using a global policy.

Default Policy

Juniper firewalls have a default out-of-the-box policy that makes them drop any traffic that doesn't match any other policies. This default policy is a hidden global policy. Juniper offers this as a security feature to ensure that any traffic you don't want to allow through is automatically dropped. This mitigates the risk of the firewall on the network by dropping any unmatched traffic. It is possible to change the behavior of this traffic from the CLI.

To override the default behavior (and therefore allow all traffic), enter the following command:

```
set policy default-permit-all
```

To change the firewall to deny all traffic by default, enter the following command:

```
unset policy default-permit-all
```

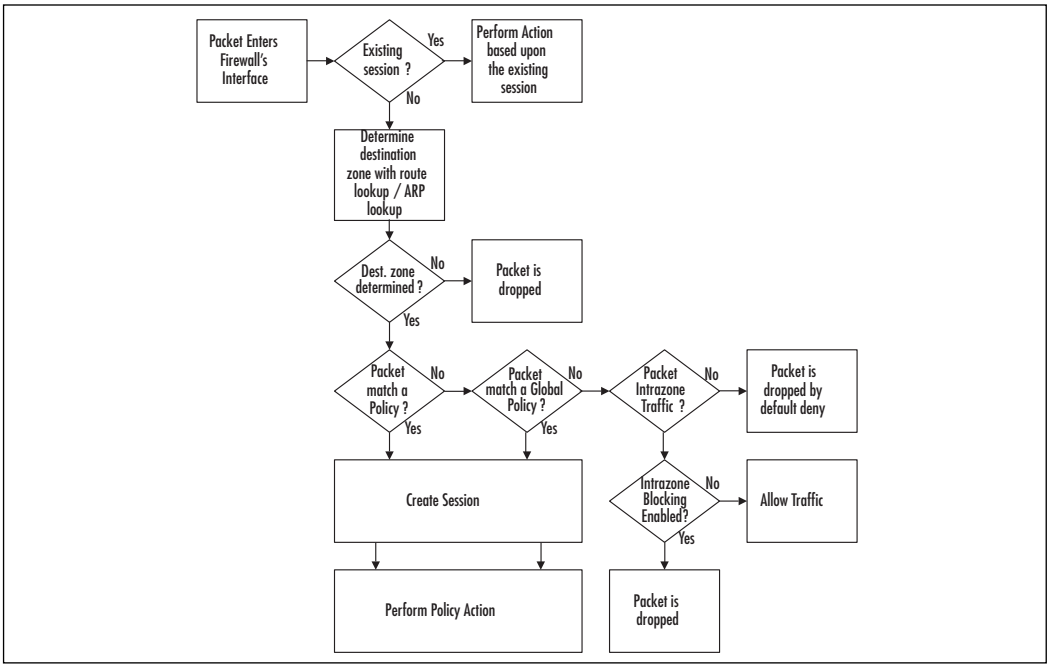
Policy Checking

When a connection is attempted, the Juniper firewall will receive the source packet on an interface in the source zone. To determine the destination zone, the Juniper firewall will perform either a route lookup or an Address Resolution Protocol (ARP) lookup to determine where the destination zone is. The decision for whether to use a route lookup or ARP lookup depends on what mode the firewall is in. If the firewall is in route mode, it will use a route lookup; if it is in transparent mode (or the interfaces are in transparent mode), it will perform an ARP lookup. Once the destination zone is determined, the firewall will perform a check against the list of policies that match that zone configuration.

For example, if the source zone is determined as the trust zone, and the destination zone is determined as the untrust zone, the firewall will check that list of policies. It will check the matching policy list starting from the first policy at the top of the list down to the bottom policy in that list. The first policy that matches the source IP address, destination IP address, and the service will be applied to that connection. The action of the first matching policy is then applied to that connection. If the connection is permitted, the connection creates a session in the firewall's internal session table. The allowed session is thus granted access to pass through the firewall. If the action of the connection is to deny the connection, then the connection is dropped. Finally, if the action on the connection is to tunnel, then the connection is permitted, and a session is created and passed into the applicable VPN connection.

The session table is a table that is stored in memory on the Juniper firewall. It contains a list of all of the allowed connections that have already passed through the policy and have been allowed. Before a connection is compared against the policy, it is compared against the session table to see if an active session has already been started. If the firewall sees that an existing session matches a session in the session table, that traffic is allowed through the firewall. For example, if you open your browser and access a Web site, that entire connection will be stored in one session in the session table. Figure 4.3 shows a condensed version of how a Juniper firewall determines what to do with network traffic as it passes though the firewall. This topic is fully discussed in Chapter 13.

Figure 4.3 Juniper Packet Logic



Tools & Traps…

The Five-Tuple

Many firewall or packet filtering products use a common filtering strategy. The matching mechanism is typically: Source IP, Destination IP, and Service or Port. It is the most basic way to identify a traffic flow. On a Juniper firewall, we increase this to five matching components. We not only use the previously mentioned components, but we also increase this to include the Source Zone and Destination Zone.

These two additional components make the match a five component matching mechanism. This leads to a more discriminate match and speeds up policy lookups. The term that Juniper engineers use for this match is the five-tuple, or 5-tuple.

Getting Ready to Make a Policy

Creating policies is actually a very easy process once you have all of your components in place. Much like building a house, you cannot build if you do not have all of the materials. Let's review our steps in creating a policy.

- **Have you determined your source and destination zones?** The source zone is going to be where the originating traffic for your connection is coming from. This is tied into your interfaces. If you can determine which interface the traffic is going to come in through, you can determine which zone the traffic will come in as the interface is tied to the zone. The destination zone is the zone that the traffic will use to exit the Juniper firewall. You can determine this by identifying which interface the traffic will come out from with routing. Look at the configuration of your network and see where the packet will route as it exits your firewall. The interface it exits is bound to a zone, and that zone is your destination zone.
- **Have you determined your source and destination IP addresses?** The source IP address can be a single IP address, multiple IP addresses, or every available IP address. In the interest of security, you should limit the IP addresses to as few as possible. If you are unsure which IP address you want to use, open the source up to a larger pool of IP addresses, then log the traffic as it goes through the firewall. Over time you can specify a smaller group of IP addresses for the source IP address. When determining your destination, you can use the same procedure by using logging to determine which IP addresses you can limit your traffic to. Ensure that the source and destination addresses have been created as address book entries. If the address book entries have not been created, now is the time to create them.
- **Have you determined which services you are going to allow in your policy?** Determining the services you want to use in your policy is a key factor in creating your policies. It is very important that you limit the amount of allowed services to the bare minimum, even if this means you will have 500 services permitted in your policy. This amount of services is much better than allowing all 65535 possible ports. Even policies that are outbound from your internal network should be limited. The more ports you have open, the more risk there is. An example of this would be if a virus was to infect a desktop and then that desktop began sending SMTP (Simple Mail Transfer Protocol) mail out directly to the Internet. Should all desktops be allowed to access the Internet directly with SMTP? These are the questions you should ask yourself as you create your policy.
- **Which action do you want to perform on matching traffic?** Now that you have narrowed down your traffic to exactly what you want to match, it's time to determine what you want to do with this traffic. You have three options: permit, deny, or tunnel. When you select to permit traffic, you are allowing the traffic to

pass between the two security zones on the firewall. The second option is to deny, or drop, the traffic before it passes through your firewall. By default, the firewall blocks all traffic as it attempts to pass through, so creating a policy to deny traffic allows you to apply special properties to the traffic such as logging. The last option is to tunnel the traffic. When you choose to tunnel the traffic, you are first explicitly permitting the traffic, but only to pass into a VPN tunnel. Choosing the tunnel option also forces you to choose a VPN that the traffic must pass into. (Configuring VPNs is discussed in Chapter 11.)

- **Where are you going to position your policy?** The position of your policy determines when your policy will take effect. Policies are checked in a top-down order based upon your source and destination zones. Once the source and destination zones are determined, the list of policies that matched the source and destination zone is checked, starting from the top policy and going to the bottom of the policy list. Once all of the policies in the matching source and destination zone are checked, the global policies are applied to your traffic.
- **Are there any additional options you want to apply to the traffic?** As we mentioned earlier in the chapter, there are many different options you can apply to your policy beyond the required components. In this chapter, we will look at the logging option only. When you turn on logging for a policy, each connection that passes through the firewall is written into the traffic log. (In the next chapter, we will look at all of the available options for policy creation.)

Policy Components

Juniper uses what is called the five-tuple or 5-tuple for matching when creating a firewall policy. This breaks down that you are defining five separate components:

- Source address
- Destination address
- Service
- Direction
- Action

Zones

When creating a policy, you must first determine the source and destination zones. The source zone will be where the source traffic is going to come from. The destination zone is the location where the destination traffic is going. Because zones are bound to interfaces,

you are also inherently choosing which interface the traffic will be using. This may help you when creating a policy since the concept of zones is different from many other firewall products.

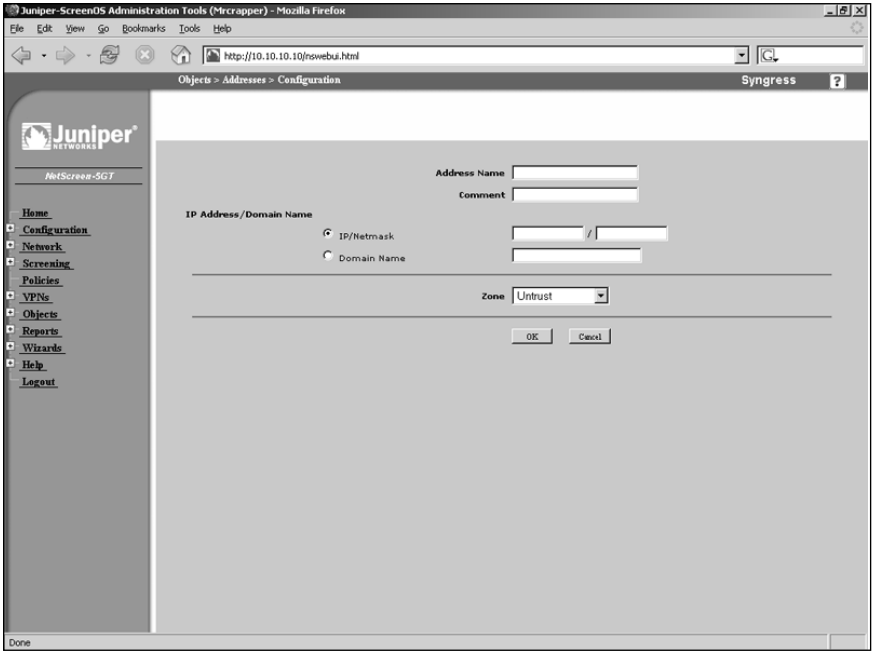
Address Book Entries

The next component you need to determine when creating a policy is which source systems should be able to access which destination systems—essentially, the source IP addresses and the destination IP addresses for your policy. This is a common firewall concept that you may have come across before. When using the command-line interface, you must create all of your address book entries before you make your policies. However, when using the WebUI to create policies, you can create new address book entries as you create the policy. If you choose this latter method of creating address book entries while creating a policy in the WebUI, you can only specify the IP address and netmask for the entry. You will have to go back at a later time and edit the address book entry if you want to associate a name with the address book entry. This idea will become clearer later in the chapter as we look at some examples of address book entry creation.

Creating Address Book Entries

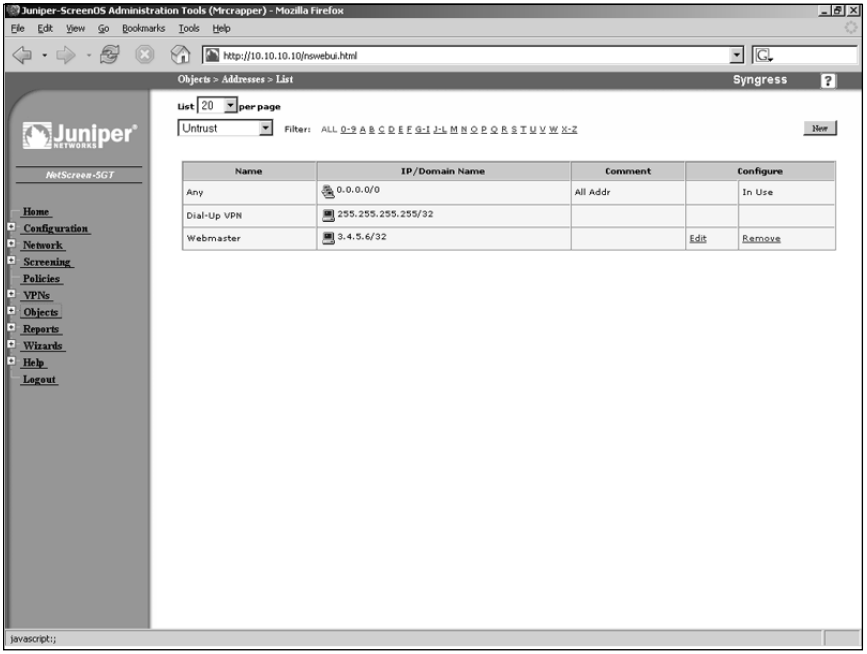
Figure 4.4 shows the WebUI address entry creation screen. Use the steps that follow to create an address book entry via the WebUI.

Figure 4.4 Address Book Entry Creation (WebUI)



1. Select **Objects** | **Addresses** | **List**.
2. Click the **New** button in the upper right-hand corner of the page.
3. Enter the **Address Name**. Refer to the "Naming Convention Errors" sidebar in this chapter to avoid naming errors.
4. If desired, use the **Comment** field to enter additional information about the address book entry.
5. If you wish to identify the address book entry by IP address, select and use the **IP/Netmask** fields to enter the desired IP address or IP subnet.
6. If you wish to identify the entry by domain name rather than by IP address, select the Domain Name option and enter the DNS-resolvable name. Note that your system must have DNS enabled for this feature to work properly.
7. Use the **Zone** drop-down list to specify the zone with which the entry will belong.
8. Click **OK**. The new entry will be displayed in the address book list (Figure 4.5).

Figure 4.5 Address Book List (WebUI)



Tools & Traps…

Naming Convention Errors

When creating objects, you can cause big problems for yourself. Juniper firewalls have no understanding of what you name your objects. This can create confusion for you in the future. When naming objects, no matter if they are address objects, address groups, custom services, or service groups, you should decide upon a naming convention for your organization, otherwise, you may get confused by what the object really does for you. Even if you add a comment to each object, you are unable to see this comment when adding an object to your policy.

For example, you can create an address group with the name "WebServer." When scrolling through address objects to add to your policy, you will see "WebServer" with no other information indicating that this object is a group. You or your colleagues may add this object to a policy without realizing what it actually is. When creating groups, it's a good idea to add the suffix "Grp" so that address groups and service groups can easily be identified.

It is also suggested that you avoid creating objects using the actual policy creation screen because of the long-term confusion it can cause for you when having address objects with no names associated with them. Many people find it to be helpful to use only IP addresses for naming their address objects. If this works best for you, go ahead and create your objects with that naming convention. There is one important caveat in doing so, however. From the object creation screen, you can create an object whose name is "10.10.10.10/32" and whose actual IP address is "10.10.10.0/24." So when you attempt to add this to your policy, all the objects that are listed for you to add to your policy are listed by the name and not the actual IP address. So if your name represents an IP address, but that name does not match the actually defined IP address, you could have unexpected results in your policy.

You can also create an address book entry via the CLI. To do so, enter the following command:

```
set address zone name IPaddress "comment"
```

In the preceding command, *zone* is the zone to which this entry will belong, *name* is the name of the entry, *IPaddress* is the IP address/subnet that specifies the range, and *"comment"* is a text string (in quotes) that serves as an optional comment about the entry. For example, the following command specifies that the WebServer entry (at 10.2.2.2/32) belongs to the untrust zone and includes the comment, "This is Darren's Web Server":

```
set address untrust WebServer 10.2.2.2/32 "This is Darren's Web Server"
```

Modifying and Deleting Address Book Entries

You can update existing address book entries via the WebUI. You may wish to do so as servers change IP addresses, or you may want to update the comments about an address object. You can modify everything about an address book entry except its zone. Note that you cannot modify an address object from the CLI; if you wish to change an address object's properties via the CLI, you must first delete it and then re-create it.

Use the following steps to modify an existing address book entry via the WebUI:

1. Access **Objects** | **Addresses** | **List**.
2. Click the **Edit** link of the address entry you wish to modify.
3. Update the desired fields and click **OK**.

Use the following steps to delete an existing address book entry via the WebUI:

1. Access **Objects** | **Addresses** | **List**.
2. Click the **Remove** link of the address entry you wish to delete.
3. Click **OK** to confirm.

Use the following commands to delete and re-create an address book entry (in lieu of being able to modify it directly).

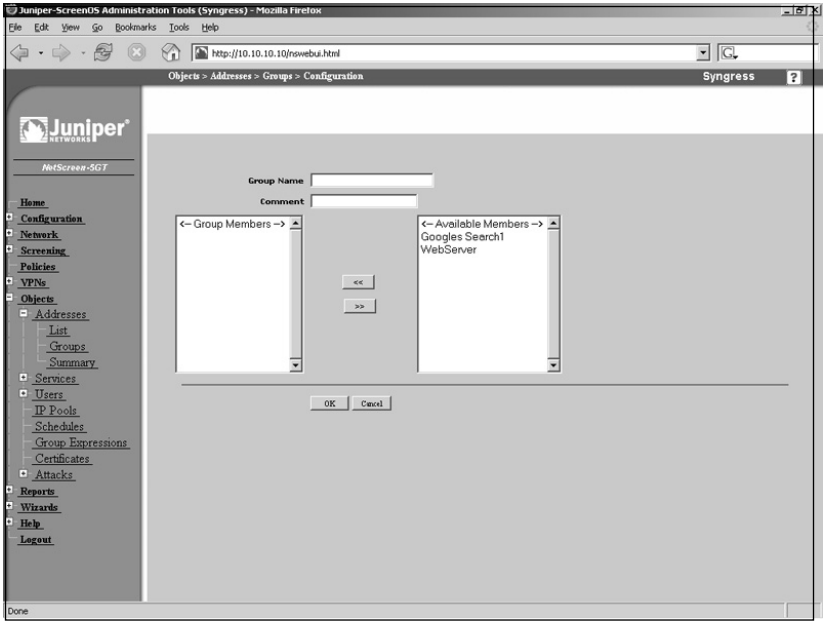
```
unset address domain "name"
set address domain name IPaddress "comment"
```

Address Groups

As you begin to amass many address objects, you will want a method to bring all of these address objects together into logical containers. This is accomplished with the use of address groups. An address group is a logical container that literally groups together address objects. Address groups are very handy when creating policies. Use the following steps to create an address group via the WebUI:

1. Access **Objects** | **Addresses** | **Groups**.
2. Click **New**. A screen similar to the one shown in Figure 4.6 will be displayed.
3. Enter the **Group Name** and, if desired, a **Comment**.
4. To place hosts in this group, select them from the list of **Available Members** on the right and click the **<<** button. The host(s) will be placed in the **Group Members** list. To remove a member from the group, click it and click the **>>** button. Continue this process until the **Group Members** list contains all of the desired hosts.
5. Click **OK**.

Figure 4.6 Address Group Creation (WebUI)



To modify a group, access the group list and click its **Edit** button. To delete a group, access the group list and click its **Remove** button, then click **OK** to confirm.

To create an address group using the CLI, enter the following command:

```
Set group address zone groupname comment "commenttext"
```

In the preceding command, *zone* is the zone to which the group will be placed, *groupname* is the name you wish to give the new group, and *commenttext* is the text you wish to place in the comment field (must be in quotes).

Use the following command to add an address to the group:

```
Set group address zone groupname add addressname
```

In the previous command, *zone* is the zone that contains the desired address and group, *groupname* is the name of the group, and *addressname* is the name of the address you wish to place in the specified group.

Services

The next component in creating your policy is using *services*. Services are the protocols that you would use to access a system over the network. Services on a Juniper firewall are represented by service objects. A service object is used to specify which applications can be used in a policy. Every Juniper firewall comes with a predefined set of services. The set of services that comes on your firewall varies per version of Screen OS you are running on your firewall.

Currently, Screen OS contains about 80 predefined services. These services are some of the more commonly used services you will use when defining your policies. Some protocols are also predefined because they function in a nonstandard way. One example is the FTP protocol. Because FTP sends special port redirects during its communication, Juniper has created a special mechanism to read inside the FTP connection to determine which ports to open up during the communication. Even though the predefined service only allows TCP port 21, the firewall is still able to dynamically allow ports based upon the FTP communication.

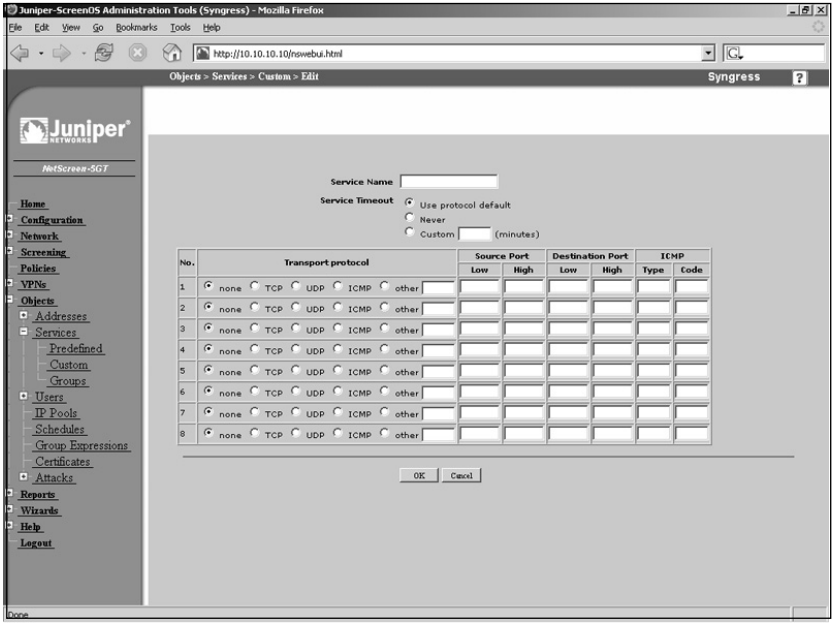
It would be impractical for Juniper to create every service that exists. Juniper allows you to create your own custom service objects. These custom service objects can be used just like a predefined service object in your policy. When Juniper firewalls match in a policy, they match based on the destination port in the packet. This is how a service matching is performed on a Juniper firewall.

Creating Custom Services

A service object has several defining properties that tell the firewall how to identify traffic. These properties are specified when defining a new service object. The options you use when creating a new policy depend on the type of protocol you are creating. Use the following steps to create a custom service via the WebUI:

1. Access **Objects | Services | Custom**.
2. Click **New**. A screen similar to the one shown in Figure 4.7 will be displayed.

Figure 4.7 Service Object Configuration (WebUI)



3. Enter the **Service Name**.
4. Use the Service Timeout options to specify how long the service session should stay open. The protocol default is 30 minutes for TCP and one minute for UDP. Select **Never** if you do not want to impose a timeout value. To specify your own timeout value, select the **Custom** option and enter the desired number of minutes (up to a maximum of 40 minutes).
5. You can define up to eight protocols for this service object. This can be useful in creating a service that uses multiple ports. To define a protocol, select its type from the **Transport Protocol** field. Next, enter the **Low** and **High Source Ports** and the **Low** and **High Destination Ports**. To specify a single port, enter the same number in the **High** and **Low** fields.
6. Click **OK**.

To create a custom service via the CLI, enter the following command:

```
set service servicename protocol protocol src-port srclow-high dst-port dstlow-high
```

In the preceding command, *servicename* is the name of the service, *protocol* is the protocol type (TCP, UDP, or ICMP), *srclow-high* is the low and high source port range, and *dstlow-high* is the low and high destination port range.

Tools & Traps…

What Exactly Is "Any"?

When creating policies on a Juniper firewall, you will see the option "any" available for the source, destination, and service. This is available here on Juniper firewalls and on other firewall products. The question always comes down to, "what does *any* actually mean?"

On a Juniper firewall, *any* literally means any address in the zone (when used as a source or destination address) and any service (when used as a service). This is something important to note, as other firewall products do not always mean "any," even when they say "any."

Modifying and Deleting Services

After creating your service, there may be times when you will want to modify that service or perhaps delete it. Modifying a service is much like creating it. The only difference is that

when you come to the editing screen, the portions of the service you have created are already defined for you. From the CLI, if you want to add additional protocols to a service, you can. However, if you need to edit existing parts of the service, you must delete the service and then re-create it.

Use the following steps to modify an existing service via the WebUI:

1. Access **Objects** | **Services** | **Custom**.
2. Click the **Edit** link of the service you wish to edit.
3. Make the desired changes to the **Service Name** and/or **Service Timeout** fields.
4. Modify the values for any of the existing protocols. You can add new protocols to this service simply by entering the appropriate data, and you can remove protocols by selecting the **none** option of the protocol you wish to remove.
5. Click **OK**.

To delete a service via the WebUI, access the services list and click the **Remove** link of the service you wish to delete. Click **OK** to confirm. To delete a service via the CLI, enter the following command:

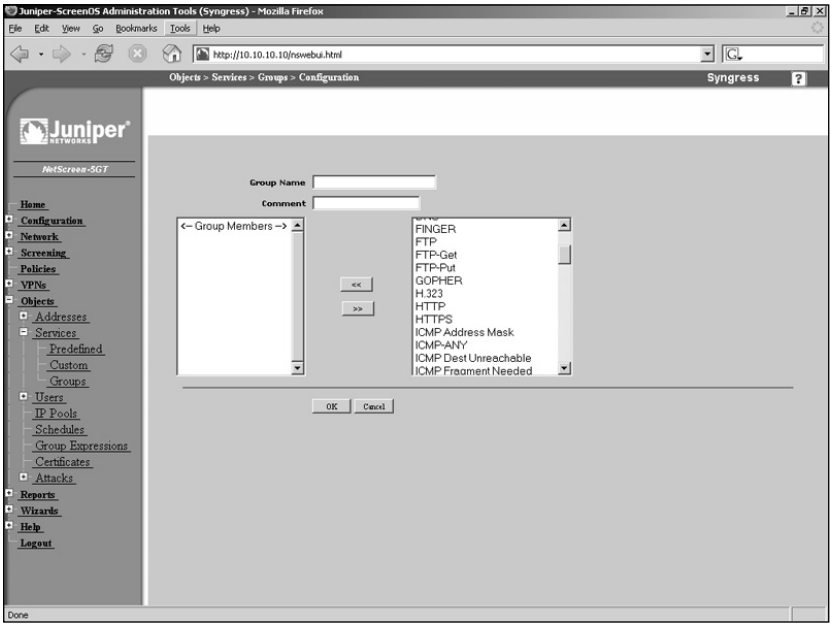
```
unset service "servicename"
```

Service Groups

Even though each individual service can contain up to eight service definitions, you will still want to group services together into logical containers. You can do this through the use of service groups. A service group functions just like an address group, and its creation is nearly identical. Use the following steps to create a service group via the WebUI:

1. Access **Objects** | **Services** | **Groups**.
2. Click **New**. A screen similar to the one shown in Figure 4.8 will be displayed.
3. Enter the **Group Name** and, if desired, enter a **Comment**.
4. To place hosts in this group, select them from the list of **Available Members** on the right and click the << button. The host(s) will be placed in the **Group Members** list. To remove a member from the group, click it and click the >> button. Continue this process until the **Group Members** list contains all of the desired hosts.
5. Click **OK**.

Figure 4.8 Service Group Creation (WebUI)



To modify an existing service group, access the service group list and click the **Edit** link of the group you wish to modify. To delete a service group, access the service group list and click the **Remove** link of the group you wish to delete. Click **OK** to confirm.

Use the following command to create a service group via the CLI, where *groupname* is the name of the new group:

```
set group service "groupname"
```

To add items to the group, enter the following command:

```
set group service "groupname" add item
```

In the previous command, *groupname* is the name of the service group and *item* is the name of the service that will be added to the specified group. To delete a service group via the CLI, enter the following command:

```
unset group service "groupname"
```

Creating Policies

Now that you are familiar with the components of creating policies, you can begin actually creating them. Policies are the main reason why you are implementing your firewall in the first place: to control network traffic. In this section, we will begin looking at how to put policy components together to form a policy.

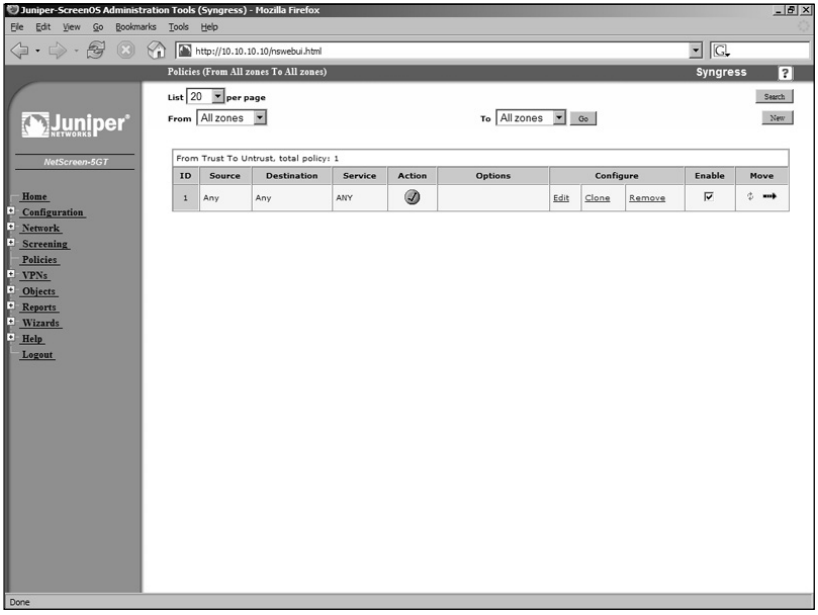
Creating a Policy

In this section, we will begin to work with policies. In all of the previous sections of the book we have worked with both the CLI and the WebUI in the same section. However, in this section we will look at the WebUI and the CLI in separate sections. This will bring better clarity to the two different methods of creating a policy. Even though the CLI is not as easy to use as the WebUI, knowing how to use the CLI is crucial. The configuration is always stored as CLI commands, so knowing what each command does will empower your use of the platform.

Creating a Policy via the WebUI

The WebUI is easier to interpret, it allows for easier modification of the policy, and at times can be faster to use. When you start to have over 20 policies on your firewall, the CLI will seem as if all the policies run together, whereas on the WebUI, the icons and coloration of the policies will seem to flow. This is all a matter of preference, but I suggest using whatever tool makes the most sense to you. There is no reason to make the administration of the Juniper firewall harder on yourself than it has to be. In Figure 4.9, you will see what the main policy page looks like. This page is the root of all policy creation in the WebUI.

Figure 4.9 The Root of Policy Creation



From here, we can do everything we need to do with policies. We can create, remove, reorder, search, enable, disable, and clone policies. To access this screen, simply select the **Policies** link from the menu on the left side of the screen. As you can see in Figure 4.9,

currently we only have one policy. This policy allows any source to go to any destination via any protocol. The action (indicated by the checkmark in the green circle) is permit. Table 4.1 lists the different icons that may be displayed on this screen, as well as their descriptions.

Table 4.1 Policy Action Icons

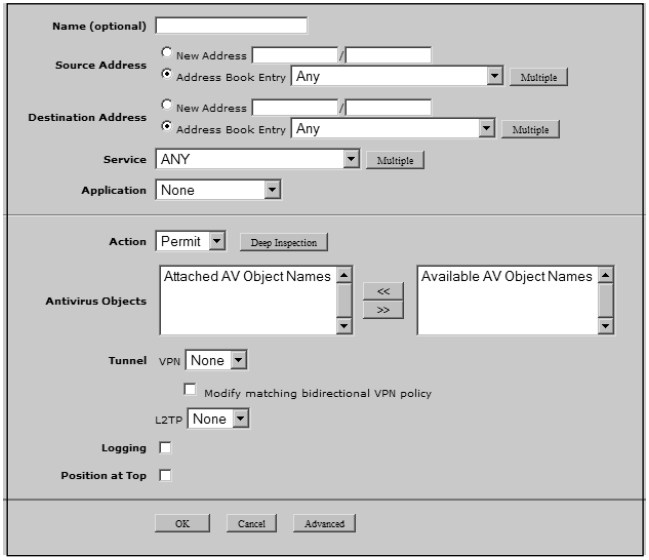
Action	Icon	Description
Permit		The permits the traffic specified in the policy.
Deny		This denies the traffic specified in the policy.
Tunnel		The policy permits and then tunnels the matching traffic.
Bi-Directional Tunnel		The policy permits and then tunnels the matching traffic. It also has a matching policy that has the source and destination reversed.
Policy Based NAT		This policy permits the traffic matching the policy but it also performs NAT on the traffic.

These various policy icons are very informative and simple to understand. When defining a new policy from the WebUI, you begin by selecting the source and the destination zones. Once you select the zones and create the new policy, there is no way to change the source and destination zones. If you wish to change the source and destination zones, you must delete the undesired policy and then create a new one with the correct zones.

Use the following steps to create a policy via the WebUI:

1. Access the Juniper screen administration tools page and click **Policies** in the menu.
2. Click **New**. A screen similar to the one shown in Figure 4.10 will be displayed.
3. Enter the policy **Name**. This should be a descriptive name that will allow you to identify what the policy does.
4. Use the **Source Address** options to specify the source address for the policy. If it is a new address, select the **New Address** option and enter the IP address range. If the address already exists in the address book, select the **Address Book Entry** option and enter the name of the entry. You can select multiple address book entries by clicking the **Multiple** button.

Figure 4.10 Policy Definition Screen



5. Use the **Destination Address** options to specify the source address for the policy. If it is a new address, select the **New Address** option and enter the IP address range. If the address already exists in the address book, select the **Address Book Entry** option and enter the name of the entry. You can select multiple address book entries by clicking the **Multiple** button.
6. Use the **Service** drop-down list to specify the services you want to use in this policy. Select a single service or group of services, or select **ANY**, or click **Multiple** if you wish to specify multiple (but not all) services.
7. Use the **Application** drop-down list to map a custom-defined service to a specific application layer.
8. Use the **Action** drop-down list to specify whether matching traffic will be permitted, denied, or tunneled. If you select **Tunnel**, you must also select an option from the **Tunnel** drop-down list. To apply deep inspection groups to the policy, click the **Deep Inspection** button. (Deep inspection is explained in more detail in Chapter 10.)
9. The **Antivirus Objects** section allows you to specify which antivirus scanners will be applied to the policy. To select an antivirus object, select it from the **Available AV Object Names** list on the right, and then click the << button to place it in the **Attached AV Object Names** list on the left.
10. If you selected **Tunnel** in the **Action** drop-down list, use the **Tunnel VPN** drop-down list to specify the appropriate VPN tunnel. (VPN configuration is discussed in greater detail in Chapter 11.)

11. If you wish to turn on logging for this policy, enable the **Logging** checkbox.
12. If you wish to place this policy at the top of the list of policies with matching source and destination zones, enable the **Position at Top** checkbox.
13. Click **OK**.

Reordering Policies in the WebUI

Once you have all of your policies created in the WebUI, you may find you need to reorder them. Every newly created policy is placed at the bottom of the policies that have the same source and destination zones unless you enabled the **Position at Top** option when creating the policy. Once the policy is created, you can modify the policy placement on the Policies list page. Table 4.2 shows the different icons you can use to reorder policies.

Table 4.2 Policy Action Icons

Icon	Description
	Selecting this option allows you to choose the placement of your policy, by policy number. A pop-up window will be displayed, asking you where you want to place your policy based upon the number of your policy. (See Figure 4.11 for an example.)
	This option allows you to specify where you want to place your policy based upon a selection screen. At the selection screen, you can click on a similar arrow to choose where you want to place your policy. (See Figure 4.12 for an example.)

Figure 4.11 Order Policies by Number

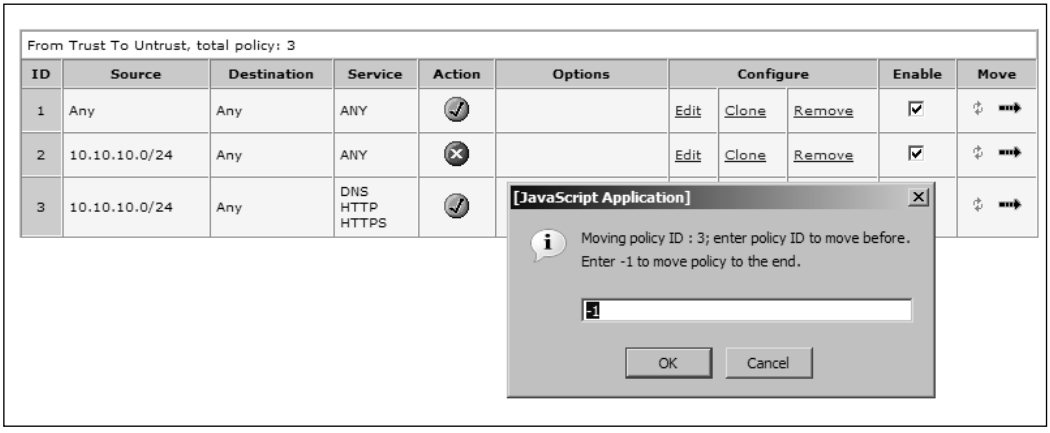
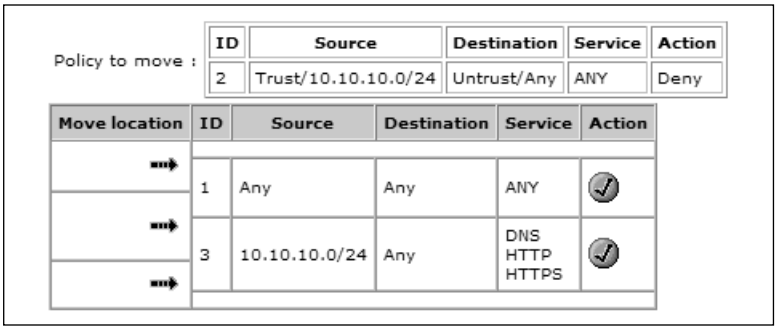


Figure 4.12 Choose Policy Placement



Tools & Traps…

Negation

When creating policies and working with address book entries, you can enable an option called *negate*. This concept is used on several firewall products and can be quite useful depending on what you are attempting to accomplish. The negate option is available for the source and destination addresses. The option is turned on for either source or destination addresses, and can be turned on separately for each policy.

Turning on the negate option will apply the following logic: everything except the selected objects. For example, suppose you created a policy with the following configuration: Source: 10.10.10.0/24 Negated; Destination: Any; Service: FTP; Action: Permit. You are effectively saying, "Allow any source address to FTP, except for 10.10.10.0/24." This can save you time instead of making a policy to deny the 10.10.10.0/24 network to access FTP and then a second policy to allow access to FTP to any.

The negate option can be used in both the WebUI and the CLI. To use this option in the WebUI when you are creating a policy, click the **Multiple** button for the source or destination address. Once you have selected what you want to negate in the pop-up window, enable the **Negate the Following** option, which can be found in the upper left-hand corner of the window. To use this from the command line, you must first create the policy, then go into the sub-shell for the policy and negate the source address and destination address. See the following command for an example:

```
Syngress-> set policy id 3
Syngress(policy:3)-> set src-address negate
Syngress(policy:3)-> exit
```

Continued

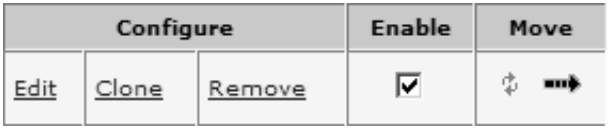
```
Syngress-> get policy id 3
name:"none" (id 3), zone Trust -> Untrust,action Permit, status "enabled"
1 source (negated): "10.10.10.0/24"
1 destination: "Any"
3 services: "DNS", "HTTP", "HTTPS"
Policies on this vpn tunnel: 0
nat off, url filtering OFF
vpn unknown vpn, policy flag 0000, session backup: on
traffic shapping off, scheduler n/a, serv flag 00
log no, log count 0, alert no, counter no(0) byte rate(sec/min) 0/0
total octets 0, counter(session/packet/octet) 0/0/0
priority 7, diffserv marking Off
tadapter: state off, gbw/mbw 0/-1
No Authentication
No User, User Group or Group expression set
Syngress->
```

Other Policy Options in the WebUI

Some additional WebUI options may be helpful as you begin to create policies. These options are available from the root policies page in the WebUI (see Figure 4.13).

- **Edit** Use a policy's **Edit** link to modify its configuration.
- **Clone** Use this option to create a copy of the policy. The policy's original information will be displayed, but can be edited for your needs. This can save time when creating multiple policies that have only slight differences.
- **Remove** Click a policy's **Remove** link to delete it. The policy will immediately be removed from the firewall.
- **Enable** Use this option to enable or disable the policy.

Figure 4.13 Additional Policy Options



Creating a Policy via the CLI

Even though the point-and-click nature of the WebUI may make policy management easier, the CLI provides the fastest methods of policy management. Using the CLI requires more memorization of the commands and the order in which you use them. Once you get a grasp of CLI policy management, it will become an effective management tool. Three basic commands can be used to manage policies. The first command is *set policy*, which is the root of all policy creation. All commands that involve creating and manipulating policies begin here. The second command is *get policy*, which displays information about all, or specified, policies. Finally, the *unset policy* command is used for removing policies.

```
Syngress-> set policy ?
before               insert a policy
default-permit-all   permit if no policy match
from                 from zone
global               set global policy
id                   specify policy id
move                 move a policy
name                 specify policy name
top                  put this policy as the first one in the list
Syngress-> get policy ?
>                    redirect output
|                    match output
<return>
id                   show one policy
all                  show all policies(including global policy)
from                 from zone
global               show global policies
Syngress-> unset policy ?
<number>             policy id
default-permit-all   permit if no policy match
id                   policy id
Syngress->
```

To view a list of all existing policies, enter the command *get policy*. You can also list policies by specifying the source and destination zones. This is done with the command *get policy from <Src-Zone> to <Dst-Zone>*. A list of all policies matching the specified parameters will be displayed. Use the command *get policy global* to view all of the global policies. Finally, use the command *get policy all* to view all of the policies, including the global policies. The *get policy* command supports the following parameters:

- **ID** This is the ID number of the policy. It is a unique number that is used to identify the policy.
- **From** The source zone.

- **To** The destination zone.
- **Src-address** The source address objects.
- **Dst-address** The destination address objects.
- **Service** The service specified for the policy.
- **Action** The action to apply to the traffic that matches the policy.
- **State** Whether the policy is enabled or disabled.
- **ASTLCB** This represents which special properties are turned on in the policy. A = Authentication, S = Scheduling, T = Traffic Shaping, L = Logging, C = Counting, B = HA Backup.

```
Syngress-> get policy
Total regular policies 4, Default deny.
ID From   To      Src-address Dst-address Service Action State   ASTLCB
1  Trust  Untrust Any         Any         ANY     Permit enabled -----X
2  Trust  Untrust 10.10.10.0/ Any         ANY     Deny   enabled -----X
3  Trust  Untrust 10.10.10.0/ Any         DNS     Permit enabled -----X
                                          HTTP
                                          HTTPS
4  Trust  Untrust Any         Any         ANY     Permit enabled -----X
Syngress->
```

You can even look at the configuration of a policy by using the *get policy id <number>* command, where *<number>* is the policy ID.

```
Syngress-> get policy id 1
name:"none" (id 1), zone Trust -> Untrust,action Permit, status "enabled"
src "Any", dst "Any", serv "ANY"
Policies on this vpn tunnel: 0
nat off, url filtering OFF
vpn unknown vpn, policy flag 0000, session backup: on
traffic shapping off, scheduler n/a, serv flag 00
log no, log count 0, alert no, counter no(0) byte rate(sec/min) 0/0
total octets 1301676800, counter(session/packet/octet) 0/0/0
priority 7, diffserv marking Off
tadapter: state off, gbw/mbw 0/-1
No Authentication
No User, User Group or Group expression set
```

Creating a policy via the CLI requires the same components as if you were using the WebUI. The full command for creating a policy via the CLI is

```
set policy from <Src-Zone> to <Dst-Zone> <Src-Address> <Dst-Address> <Service>
<Action>
```

Five areas in the preceding example command must be filled in to complete the policy. The *<Src-Zone>* or source zone, *<Dst-Zone>* or destination zone, *<Src-Address>* or source address book entry, *<DstAddress>* or destination address book entry, *service*, and *action*. These are the same five minimum options you would use when creating a policy from the WebUI. Once you create the policy, it is given a policy ID or unique identifier. This identifier is used to reference the policy throughout the system. The firewall will return *policy ID = <Identifier>* once the policy has been created.

Notice that this command only allows you to specify one source address, one destination address, and one service. You can add more once the policy has been created by using the *set policy id <ID Number>* to enter the sub-shell that allows you to modify the policy. The sub-shell for policies is the only sub-command shell in the entire firewall.

Once in the policy sub-shell, you have the same options as in the regular shell: set, get, and unset. Using the *set* command, you can add additional source addresses, destination addresses, and services, as well as other policy options. The *unset* command is used to remove parts from the policy, and the *get* command is used to obtain information about the policy. When creating policies from the CLI, you can place a policy in a specific position as it is created by entering the following command:

```
set policy before <ID> from <Src-Zone> to <Dst-Zone> <Src-Address> <Dst-Address>
<Service> <Action>
```

Specify the <ID> as the ID number of the policy you want to place the policy before. If you want to create a policy and place it at the top of the list of policies with the same source and destination zone, you would use the following command:

```
set policy top <Src-Zone> to <Dst-Zone> <Src-Address> <Dst-Address> <Service>
<Action>
```

The following is a snippet of code that shows an example of creating a policy and manipulating it in the sub-shell.

```
Syngress-> set policy from trust to untrust 10.10.10.0/24 any FTP permit
policy id = 6
Syngress-> get policy id 6
name:"none" (id 6), zone Trust -> Untrust,action Permit, status "enabled"
src "10.10.10.0/24", dst "Any", serv "FTP"
Policies on this vpn tunnel: 0
nat off, url filtering OFF
vpn unknown vpn, policy flag 0000, session backup: on
traffic shapping off, scheduler n/a, serv flag 00
log no, log count 0, alert no, counter no(0) byte rate(sec/min) 0/0
total octets 0, counter(session/packet/octet) 0/0/0
priority 7, diffserv marking Off
tadapter: state off, gbw/mbw 0/-1
No Authentication
No User, User Group or Group expression set
```

```
Syngress-> set policy id 6
Syngress(policy:6)-> set service DNS
Syngress(policy:6)-> set src-address 10.10.9.0/24
Syngress(policy:6)-> set name "Allow FTP"
Syngress(policy:6)-> set log
Syngress(policy:6)-> exit
Syngress-> get policy id 6
name:"Allow FTP" (id 6), zone Trust -> Untrust,action Permit, status "enabled"
2 sources: "10.10.10.0/24", "10.10.9.0/24"
1 destination: "Any"
2 services: "DNS", "FTP"
Policies on this vpn tunnel: 0
nat off, url filtering OFF
vpn unknown vpn, policy flag 0000, session backup: on
traffic shapping off, scheduler n/a, serv flag 00
log yes, log count 0, alert no, counter no(0) byte rate(sec/min) 0/0
total octets 0, counter(session/packet/octet) 0/0/0
priority 7, diffserv marking Off
tadapter: state off, gbw/mbw 0/-1
No Authentication
No User, User Group or Group expression set
Syngress->
```

Other Policy Options Available in the CLI

Once you have all of your policies defined, you can use the CLI to reorder the policies. To move an existing policy above another, use the following command:

```
set policy move <ID1> before <ID2>
```

Specify the policy you want to move with its policy ID as <ID1> and the policy you want to move it before as its policy ID as <ID2>. To move an existing policy after another, use the following command:

```
set policy move <ID1> after <ID2>
```

Specify the policy you want to move with its policy ID as <ID1> and the policy you want to move it after as its policy ID as <ID2>. This may seem like an insignificant option, but if you have ever used a Cisco IOS or Cisco PIX access list, you will appreciate this option. Neither Cisco OS allows you to manipulate the policies or access lists this way. Instead, you must first remove all of the applied policies and then add them all back to the firewall. Finally, you can delete policies via the CLI. To delete a policy from the CLI, you must know the policy ID of the policy you want to remove, and then use either the *unset policy id <ID>* or *unset policy <ID>* command.

Summary

In this chapter, we focused on the basics of policy creation. The basics that we looked at are the foundation for much more to come in the way of policies. We looked at policies in this chapter as a primary tool of access control. In the next chapter, we will expand on this by looking at various other options you can apply to policies. When creating a policy on a Juniper firewall, you must have a minimum of five components. This idea is continually stressed, as it will help you ease into policy creation on a Juniper firewall.

The first section of the chapter Juniper Policies took us through the main ideas of policies on a Juniper firewall. When creating your list of policies you must create policies from least specific to most specific. This will apply the specific policies first to your traffic since the least specific policies may unintentionally match your traffic. Also in the first section, we looked at the three types of policies, as well as how and where they take effect. All three policies are very similar, but they are classified based upon the combination of zones in the policy.

When creating policies on a Juniper firewall, you build them out of components. These components must be created before you make a policy. Each one of the components for a Juniper firewall is treated as an object. The components that we looked at in this chapter are the main components for a policy. Address objects represent hosts or subnets of IP addresses. Service objects can be a strange concept. Many competitive firewall products create services as a single protocol. If you want to create several services and represent them as a compilation, you must make a group. On a Juniper firewall, a service object can contain up to eight protocols. This allows you to take an entire suite of protocols and make them into one logical object.

Policy creation is a common task for an administrator of a Juniper firewall. In this chapter, we looked at the two methods of policy creation: the WebUI and the CLI. Each has its own merits. The WebUI may be easier to use for looking at policies, while the CLI may be faster for creating policies. The choice is yours, but never limit yourself to a single option. It always pays to be familiar with both options because in the end all policies are stored as CLI commands. If you want to use the CLI to do something, but are unsure of the command, you can probably do what you need to do from the WebUI. Then look at the configuration from the CLI to see what the commands are to use the CLI in the future.

Solutions Fast Track

Firewall Policies

- ☑ A policy on a Juniper firewall is what other firewall products consider a single rule.
- ☑ When creating a policy, the policies at the top should start with the most specific access and descend to the least specific access.

- ☑ Three types of policies are available on a Juniper firewall: intrazone, interzone, and global.

Policy Components

- ☑ Five components are required to create a policy: source zone, destination zone, source address, destination address, and services.
- ☑ When naming address objects or service objects, it is best to decide on a naming convention to ease long-term administration.
- ☑ Service objects can contain up to eight individual protocols.

Creating Policies

- ☑ The WebUI and the CLI can both be used for creating policies. However, it may be easier for people to use the WebUI because of its GUI nature.
- ☑ If you want to keep a policy, but not have it stay active, you can disable the policy.
- ☑ When creating policies via the CLI, you have more choices over where the policy will initially be placed.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: Can you explain the least restrictive to most restrictive policy ordering again?

A: The list of policies with the same matching source and destination zone are checked from the top down. If you were to place less restrictive policies, such as policies allowing entire networks as your source, before individual hosts, the policies with the networks would apply to your traffic first. This may lead to unexpected results if more specific policies are not taking effect first. The Juniper firewall has no automatic way to determine if the list of policies are listed in the right order. You can, however, use the CLI command *exec policy verify* to see if you have policies overshadowing each other. This still would require you to manually make changes to fix the policy ordering.

Q: Is it possible to use IP address ranges as address objects?

A: When creating address book objects, you can only create objects based on subnetting. Even when you make a single host object, you are making it with a 32-bit mask only allowing for a single host. If you require a range of hosts, see if you can fit it into a subnet. If you can't, you will be required to create each host individually, and then place them into a group.

Q: I clearly can make address objects while making policies. I do not understand why you are against this.

A: I firmly believe in creating each part of your policy in order. I think it is best to create all of your objects before you attempt to use them inside of policies. If you need to create a quick address object in a policy, then go right ahead. Rename the object to something that makes sense to you. Everyone has his or her own style of management, so use whatever best suits you.

Q: I am familiar with using other firewall software, and I am confused about why you would bind address objects to zones.

A: Having address objects inside of zones just furthers the zone concept. It is essentially binding that object into the logical location of a zone. Because most other firewall software does not use zones, they essentially have no need to organize address objects in any way other than by type.

Q: What are the differences between service objects and service groups?

A: A service object contains the protocol definition of a service. Each service object allows you to define up to eight protocols in a single service. A service group only contains service objects.

Chapter 5

Advanced Policy Configuration

Solutions in this chapter:

- **Traffic-Shaping Fundamentals**
- **Deploying Traffic Shaping on Juniper Firewalls**
- **Advanced Policy Options**

- ☑ **Summary**
- ☑ **Solutions Fast Track**
- ☑ **Frequently Asked Questions**

Introduction

As you may have already noticed, there is quite a lot you can do with firewall policies. In this chapter, we will cover some other functionalities that can be configured within policies and interfaces. One of the great strengths of the Juniper firewalls over other vendors is that you can configure many of the Juniper options on a very granular basis. This allows you to make specialized decisions regarding the firewall's behavior for specific conditions, as opposed to being forced to make broad decisions across the whole platform. We will specifically focus on traffic shaping, counting, and policy scheduling in this chapter. Other chapters will cover additional policy topics such as Network Address Translation, user authentication, and attack and prevention.

We will begin this chapter with a discussion of traffic shaping, which is also referred to as *traffic management* and *quality of service* (*QoS*). This allows you to prioritize the traffic on your network, based upon specific criteria you define. Traffic shaping can be a very complex topic, which entire books have been dedicated to covering, but we will work to demystify this powerful tool, as well as give real-world examples to help you on your way to deploying it on your firewalls. We will then follow our traffic-shaping discussion with other advanced policy topics such as counting and scheduling. These tools can be invaluable in helping you configure top-notch functionality on your firewalls.

Traffic-Shaping Fundamentals

A common theme in modern networking is how to balance the needs of certain users and applications with the resources available. This has become increasingly important in recent years with the increasing popularity of real-time interactive applications. This trend is likely to continue since more and more applications are deploying these time-sensitive protocols and functionality.

In this section, we will provide an introduction to traffic shaping and QoS in the real world. This should serve to help you in configuring traffic shaping on your networking equipment, but is not meant to be the definitive traffic-shaping guide. Even if you are already familiar with traffic shaping, this chapter should simply act as a refresher given the many traffic-shaping concepts.

The Need for Traffic Shaping

Modern networking equipment has become faster, smaller, and more robust. Today, networks are capable of supporting much more bandwidth-intensive applications, along with newer interactive real-time protocols. Even consumer-grade Internet lines have greatly improved by bringing broadband and even fiber optics into homes. All of these advancements have been welcomed, as well as feared, by many network administrators. The problem is that not all traffic is created equal—and your users will quickly recognize a network that does not work around this fact.

Bits on a wire are just bits on a wire, right? An electrical engineer's answer to this question might be "yes," while a network administrator's would probably be "it depends." Although many aspects of network equipment technology have improved, applications have become even more demanding on your network resources. What's more, many new applications have become increasingly sensitive to both bandwidth and latency. These bandwidth- and latency-sensitive applications are mostly focused around real-time applications such as voice and video.

Why is there such a difference between interactive and other client-server model traffic such as http, e-mail, and file transfer? The answer is actually pretty logical. While you probably won't notice much of a performance impact if your Web page took another second to load or your e-mail took 10 more seconds to get to the recipient, you would most certainly notice a gap in voice communication with a friend over the phone, or a lapse in video feed during a videoconference. Since interactive applications often have a steady stream of bandwidth, constant disruptions could easily make the communication useless, not to mention very annoying. The main issue with this type of communication is that it needs to be delivered in a timely manner. If it is not delivered quickly, the data won't be useful. That's why most real-time applications use UDP as the transport layer protocol, as opposed to TCP. This is because most real-time traffic can handle some packet loss without any noticeable impact. TCP, instead, would introduce overhead with acknowledgements and retransmissions that would not be worth the data that's being sent. Of course, application developers have also done a great deal to improve their applications so they can handle some delay, jitter, and loss, but there is only so much these applications can tolerate.

On the other hand, you have traffic such as HTTP, FTP, and SMTP, which tends to be *bursty* (sends a bunch of traffic all at once, and then becomes silent). This traffic is not as sensitive to loss as the real-time applications. You certainly wouldn't want chunks of your e-mail messages going missing, or your Web-blog leaving out the clinching details of your weekend. That's why these protocols employ TCP, which ensures that the data are transferred and accurate, even if it takes many times and more resources. As we mentioned before, performance on these protocols can definitely be noticeable, but the impact is often not debilitating (unless there are some really significant issues).

One common misconception that administrators make is that they can simply throw more bandwidth at the problem. Of course, bandwidth is now more attractively priced than it used to be, so this seems like a logical solution to the problem. Administrators are often disappointed when they add a significant amount of bandwidth and the performance issues either don't go away or return at a later point. The best explanation that I have for this phenomenon is from personal experiences. Back when I was in college, peer-to-peer applications were really becoming popular and were used by virtually everyone. This caused significant issues on the university network. Mission-critical applications were in contention with very demanding leisure traffic generated by students. What would happen if administrators simply doubled their bandwidth? The students would just start downloading twice as much data, while the overall percentage of load on the network would remain unchanged.

Mission-critical applications would still face the same issues, even though the network would be double the original capacity.

While pesky user traffic can put a big dent in your network performance, there is still quite a lot that can be said about legitimate user traffic causing issues. You might be able to restrict users from running unauthorized applications, but there are still a lot of issues that can occur with authorized network applications. Different network protocols might not always cooperate with each other, especially in times of network resource contention. A common scenario is a user opening an FTP session which will try to grab all the bandwidth it can, while effectively choking off VoIP traffic across a WAN link. This isn't the only problem that can occur during heavy loads on a network. While TCP has some mechanisms for congestion control, and loss recovery, this can sometimes have a bad effect on your traffic. For instance, when TCP traffic is lost, TCP will be able to recognize this and resend the traffic to the other host. This means you have just had to transmit the same data twice. Although retransmissions are common in networking, when networks are really under contention, every session may have to retransmit much of its data, thus wasting a ton of bandwidth that could otherwise be used for stable connections. This effect will quickly add up as the available resources dwindle.

Another solution that is commonly employed is to simply compress traffic over WAN links. Although this solution can aid in reducing the load on the network that connections make, you are still not addressing the problem that certain traffic that is being sent over the network requires more priority than others, that some traffic shouldn't be transmitted, and that some traffic does not compress well. The latter is often traffic that has either already been compressed or is encrypted. This traffic can actually grow in size if the compression algorithms are run on it, so most compression engines will bypass this traffic. Compression engines are certainly valuable, but they seem to be best purposed when they are compressing traffic whose sessions have been optimized.

Different Traffic Types

Every network can utilize traffic shaping to enhance network service delivery. There are many applications that are used on your network. Each type of application generates different types of traffic. These various traffic types can affect your network in different ways. Several types of common traffic include

- **Interactive Applications** Telnet and SSH applications are classified as interactive applications. This is essentially any application where you input information and you gain immediate output. Thus, the application interacts with you and what you input. The application should respond to your input immediately.
- **Latency Sensitive** VoIP and Streaming Video applications require that the information be delivered in order and in a timely fashion. These applications can be rendered useless in cases where effective delivery is not possible.

- **Bursty** HTTP (Hypertext Transfer Protocol) and SMTP (Simple Mail Transfer Protocol) applications operate by sending bursts of data instead of streams of data. Bursty is Juniper's classification of this traffic.
- **Novelty Traffic** Used by streaming media and peer-to-peer applications, novelty traffic is typically not required on a company's network and it can over-utilize your network resources.

Each type of traffic has different requirements. However, on a typical network this comes down to a first-come-first-serve type of usage, with transport layer protocols struggling to get the most out of the available bandwidth. If you could decide which traffic takes priority in using the available resources, you could make your network more efficient. This is where traffic shaping comes into play. It allows you to determine which traffic gets priority for bandwidth usage.

For protocols that are more susceptible to latency, such as VoIP, you can do two things to ensure that it will be given the proper networking environment to succeed. First, you can set the traffic to a high priority. This will ensure that the traffic is passed in a timely manner. Second, you can guarantee that bandwidth will be available for this protocol. These two powerful network tools will ensure your applications get the bandwidth they need.

How Traffic Shaping Works

Now that you understand why traffic shaping is important to implement in most networks, we will now discuss how traffic shaping interacts with traffic. At a high level, traffic shaping allows you to allocate bandwidth and delivery priority to different types of data. When configuring traffic shaping, you specify how the firewall should classify traffic, and then how that classified traffic should be processed. Three main types of traffic-shaping techniques can be deployed on the Juniper firewall: interface-based traffic shaping, bandwidth-allocated traffic shaping in policies, and priority-based traffic shaping in policies. These types can be deployed at the same time on the firewall, but it is important to have a good understanding of the firewall's traffic shaping behavior so you do not create any unwanted effects.

Bandwidth-Based Traffic Shaping

The ability to control traffic by either providing or restricting the amount of bandwidth that is available to traffic is called bandwidth-based traffic shaping. This style of traffic shaping usually allows you to define either a guaranteed amount of bandwidth, a maximum amount of bandwidth, or both. With these controls, you might say that you want traffic to have at least X bits per second (bps) of bandwidth, but no more than Y bps. That would be an example of providing a guaranteed amount of bandwidth, as well as a maximum. Alternatively, you could specify that certain traffic should get at least X bps, with no limit, which would be considered simply guaranteed bandwidth. Lastly, you might say that you would like to limit the maximum amount of traffic to a certain ceiling, which would be known as maximum bandwidth. Different situations would merit different solutions.

Guaranteed Bandwidth

Sometimes you need to ensure that certain traffic can use a certain amount of bandwidth, regardless of other conditions. For instance, you might have a service level agreement to provide a client with so much bandwidth to your services. Traffic shaping would be important because if you do not implement such a control, you cannot ensure that other traffic would not overwhelm the network resources and create performance issues.

Maximum Bandwidth

Let's say you want to make sure traffic doesn't consume a certain level of bandwidth, or perhaps you want to limit web browsing bandwidth at your organization so it doesn't use more than a specific amount of traffic. If so, you could ensure that certain non–mission-critical traffic isn't allowed to interfere with the performance of business-critical traffic.

Priority-Based Traffic Shaping

The ability to classify traffic into different queues which have a certain level of service is called priority-based traffic shaping. Priority queuing allows you to create separate queues which act as buffers for the traffic that is placed in them. Each queue is serviced based upon a specific value that is set for each queue. By defining which queue should be serviced in what order, you can ensure that certain traffic gets processed before other traffic. This calculation isn't based upon bandwidth, but rather it's based upon what queue the traffic is put into.

Choosing the Traffic-Shaping Type

Now that you understand the different facilities you have at your disposal for performing traffic shaping, you are probably wondering when you would want to use one traffic-shaping mechanism over another. Of course, much of this decision depends on what you need to accomplish, the resources you have available, and what type of traffic you are dealing with. A couple of guidelines are useful for many common situations.

First, you must evaluate the type of traffic you are dealing with. Is the traffic interactive or not? Is the traffic sensitive to latency? Is the traffic mission-critical? Does it need to meet a service level agreement? The answers to these questions will provide you with some direction for what type of traffic shaping you ought to perform.

Generally speaking, using bandwidth-based traffic shaping is a good choice for non-interactive traffic. This traffic is often bursty, but can tolerate some degree of delay. If you need to make sure it has many resources available, you would want to use the Guaranteed Bandwidth option, whereas if you needed to cap its bandwidth usage, you could assign it a maximum amount of bandwidth. You could also guarantee that it gets a certain amount of bandwidth, but does not pass a limit set. You can accomplish that scenario by using both guaranteed bandwidth and maximum bandwidth. By determining your organizational need for the different types of traffic, you will be able to evaluate what bandwidth levels you should set for the various traffic on your network.

Interactive bandwidth can be much more difficult to deal with. The reason why interactive traffic is more volatile is because the data must be delivered in a tight time frame, without much loss, in order for it to be useful. With interactive traffic, you aren't usually as concerned with the amount of bandwidth as you are with the speed with which it's delivered. That is why priority-based traffic shaping is a better option for interactive traffic. This allows you to specify what traffic should be placed in the highest-priority queue, while other traffic that is not as latency-sensitive will be placed in a lower-priority queue. The firewall will service the higher-priority queue before it will service lower-priority queues, thus the interactive traffic will get the quick service it needs.

Deploying Traffic Shaping on Juniper Firewalls

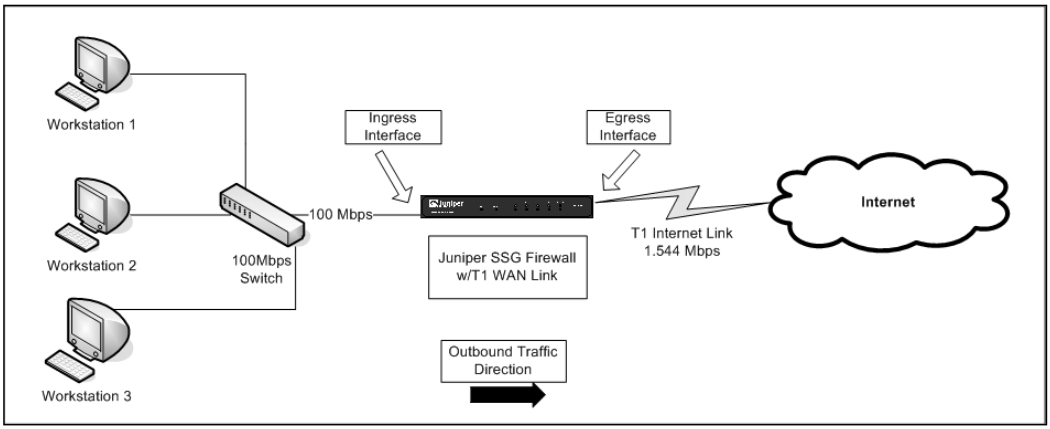
Now that we have discussed some traffic-shaping fundamentals, let's cover how Juniper implements traffic shaping on their firewalls. We will begin this section with an overview of the properties and terminology that Juniper implements in their firewalls. We will then cover several examples to help tie academic and practical knowledge together.

Methods to Enforce Traffic Shaping

The Juniper firewalls primarily use firewall policies to match traffic and enforce traffic shaping. This is a very powerful feature because it allows you to granularly configure traffic shaping specifically on traffic that you would like to match. Of course, you can also define other actions to be taken for the policy rule such as VPN tunneling, deep inspection, logging, NATing, anti-virus, and so on. We will thoroughly explore configuring traffic shaping in firewall policies later in this chapter.

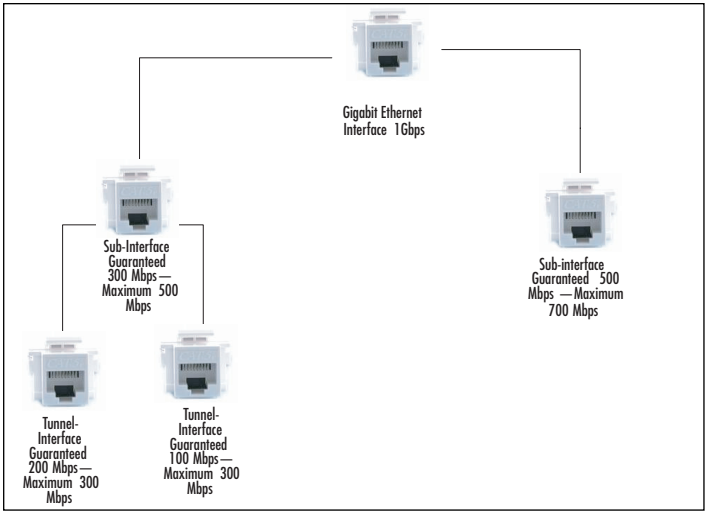
Although firewall policies are the primary mechanism to configure traffic shaping, there is another place where it is important to configure shaping. Each interface can be configured for a specific amount of ingress and egress bandwidth. The reason why it is important to configure these values is because they allow you to restrict traffic from traversing the firewall too far through the firewall's packet processing flow before getting dropped. For instance, you can define that traffic going upstream should only get 1.544 Mbps of traffic because you have a T1 interface upstream, so there isn't much sense in passing traffic much faster than that. By limiting this close to the source, traffic that exceeds this limit will be dropped before requiring that the firewall has to fully process the packet and waste resources. When this action takes place on the interface that the traffic was received, this is known as *Ingress policing*. As you can see in Figure 5.1, since the outbound path to the Internet only supports 1.544 Mbps of bandwidth, by limiting the bandwidth on the Ingress interface to 1.544 you can restrict the traffic earlier in the traffic flow. This is a more efficient use of network resources. You can also configure this action to occur as traffic leaves an interface, which is known as *Egress policing*.

Figure 5.1 Interface-Based Traffic Shaping



You can also configure interface bandwidth for traffic shaping on virtual interfaces such as sub-interfaces and tunnel interfaces. Of course, you cannot exceed the bandwidth of the physical interfaces that the virtual interfaces are applied to. Essentially, you must take the sum of the bandwidth of your virtual interface, plus the other virtual interfaces which are applied to this interface and verify that they do not exceed the bandwidth of the physical interface. If you do not configure the bandwidth on your interface, the value will be taken from the physical interface from which it is attached. Also, you can only define the maximum amount of bandwidth a physical interface can use, but on virtual interfaces you can specify a guaranteed bandwidth since that virtual interface may be in contention with other virtual interfaces on the same physical interface. Figure 5.2 shows the interface bandwidth hierarchy that applies to interface and virtual interfaces on the Juniper firewalls.

Figure 5.2 Interface Bandwidth Hierarchy



NOTE

Traffic shaping can only be configured on sub-interfaces and tunnel interfaces. It cannot be configured on aggregate interfaces, redundant interfaces, or virtual security interfaces. You can create tunnel interfaces which are bound to sub-interfaces. In those scenarios, you can actually apply interface bandwidth settings to both the *Parent* physical interface, the *Child* sub-interface, and the *Grand-Child* tunnel interface.

Interface Bandwidth Properties

When you configure the bandwidth on interfaces, you are allowing the firewall to determine how much bandwidth can effectively pass through the interface of this device. This is important because there is no sense in configuring more bandwidth than the weakest link of the path to the destination. Configuring this can help save resources and optimize the traffic patterns on your network. Listed in the following are the interface bandwidth properties that can be configured for both physical interfaces and virtual interfaces.

- **Ingress Maximum Bandwidth** This value is the maximum bandwidth setting for this interface. Any traffic that exceeds this value will be dropped, as it is over the threshold. This applies to traffic arriving on the physical interface. This value is in kilobits per second (Kbps).
- **Egress Maximum Bandwidth** When traffic leaves a physical interface to be put onto a physical link, this is known as Egress traffic. This field defines the maximum bandwidth speed that can be put onto the link as it exits the firewall. This can be useful so that you do not over-saturate a link with excess bandwidth, but rather choke the bandwidth back before it gets put on the link. This value is in Kbps.

Virtual Interface Bandwidth Properties

In addition to being able to configure the bandwidth settings for the physical interfaces, you can also do so on the virtual interfaces. The following properties can be configured on the virtual interface.

- **Ingress Maximum Bandwidth** Just like the physical interfaces, you can configure how much bandwidth can be taken in on the virtual interface to help prevent traffic from being processed that is over the traffic threshold capabilities. This is measured in Kbps.
- **Egress Maximum Bandwidth** You can also configure what the maximum bandwidth should be set to for virtual interfaces to help prevent sending more traffic than can be handled on a downstream link. This is measured in Kbps.

- **Egress Guaranteed Bandwidth** To ensure your interface receives the appropriate amount of bandwidth when multiple virtual interfaces are configured on a single physical interface, you can additionally specify the Guaranteed Bandwidth. Of course, you cannot specify more bandwidth than the interface is capable of supporting, and you must be mindful of the guaranteed bandwidth settings of other virtual interfaces on the same physical interface.

Policy-Based Properties

As mentioned at the beginning of this section, the primary method of performing traffic shaping in the firewall is to configure policies to match traffic, and then enforce traffic shaping on them. The following lists the various properties you can configure on traffic-shaping policies.

- **Traffic Shaping** This option enables traffic shaping on the policy. If this is not selected, traffic shaping will not be active for the policy.
- **Policing Bandwidth** On a policy-by-policy basis, you can define the bandwidth in Kbps for policing. What this does is restrict the bandwidth used by this policy on the ingress interface side, so that when traffic matches this policy that exceeds the bandwidth amount, that traffic will be dropped. Just like entering the maximum bandwidth on the interface, this causes traffic to only be allowed through the device if the bandwidth level doesn't surpass the threshold. This is an effective way to help reduce firewall resources because the traffic is dropped earlier in the packet flow.
- **Guaranteed Bandwidth** The amount of bandwidth that you guarantee will be available for this policy to transmit is considered the *Guaranteed Bandwidth* and can be configured on a policy-by-policy basis in Kbps. You must be careful you don't guarantee more bandwidth than what's available with other policies on the firewall. This is useful to help ensure that certain traffic will get at least a specific amount of bandwidth. If this is configured by itself, the traffic may additionally burst to any bandwidth level the firewall can support.
- **Maximum Bandwidth** Each policy can define a maximum amount of bandwidth that the traffic matching the policy can transmit in Kbps. You configure that value in this field.
- **Traffic Priority** You can set eight different traffic priority classes on the firewall. These classes determine the order which traffic will get serviced on the firewall. You can configure a specific traffic class that the policy should map the traffic to on a per-policy basis. The traffic can only be mapped to one class.
- **DiffServ Marking** Differentiated Services (DiffServ) allows you to tag packets according to their priorities. This allows you to mark individual packets in the

Type of Service (ToS) byte in the IP (Internet Protocol) header. This conforms to Request For Comment (RFC) 2474 and RFC 1349. Table 5.1 shows a mapping of the DiffServ codes to the traffic priorities configured on a Juniper firewall.

DiffServ Properties

You can configure DiffServ on your firewall systemwide and also mark packets that travel through a policy configured to do so. DiffServ on the system has the following properties:

- **DSCP Class Selector** Enabling this option will allow DSCP Class selection on your system.
- **IP Precedence** You can set the eight different DiffServ values in the ToS bits of the IP packet. The firewall will map them to the appropriate queues that you define in the IP precedence fields. If you leave it at default, priority 7 (bits 111) will be mapped to queue 0 (highest priority) all the way down to priority 7 (bits 000), which is the lowest value (shown in Table 5.1).
- **Mode** You can define three different modes for DiffServ traffic shaping on your firewall:
  - **Auto** This turns on DiffServ mapping if there is a policy that uses ingress or egress traffic shaping.
  - **On** This turns on DiffServ mapping regardless of whether there is a policy or not.
  - **Off** This turns it off regardless of other configurations.

Table 5.1 DiffServ Mapping to Juniper Priority Codes

Web	CLI	DiffServ
High	0	111
2nd	1	110
3rd	2	101
4th	3	100
5th	4	011
6th	5	010
7th	6	001
Low	7	000

It is entirely possible to grind your network to a halt with a bad traffic-shaping configuration, so it is important to consider all aspects of it before implementing traffic shaping. In the next section, we will look at and describe the various rules of traffic shaping.

NOTE

Any bandwidth that is left over from the guaranteed bandwidth may be used by other traffic. Of course, the maximum bandwidth will put a cap on how much traffic can be sent for that interface or policy.

Traffic-Shaping Mechanics

Before we jump right into a traffic-shaping configuration, let's look at the rules of how traffic shaping works on the Juniper firewalls. These very specific rules will help you understand the consequences of traffic shaping. Priority queuing contains eight different queue levels ranked from highest to lowest. The higher the queue ranking, the more precedence it gets over the other queues. If there are three policies and each policy was configured with a different priority level, the highest priority traffic would get processed first before the lower priority traffic. It's important to remember that traffic is matched on the firewall rule just like any other firewall rule, but when the rule uses traffic shaping as an advanced option, it may not transmit in a first-in first-out fashion. Depending on how you have your policies set to enforce traffic shaping, some traffic may be queued in times of contention, or dropped altogether.

If you are considering using traffic shaping, most likely you are looking for two things: a way to guarantee bandwidth to specific traffic, and a way to cap how much bandwidth specific traffic will use. Setting the guaranteed bandwidth and maximum bandwidth settings accomplishes this. These are configured on a per-policy basis, directly in the policy. When you configure guaranteed bandwidth, you are saying that the defined amount of bandwidth will be available for the traffic. There is no restriction on configuring this. So if you only have a T1 with 1.544 Mbps available, but you guarantee 10 Mbps, you will have a serious problem. The firewall can over-allocate bandwidth to this traffic. This will leave no bandwidth available for other traffic. When bandwidth is allocated for traffic, it is done so in a bidirectional manner. So, if 256 Kbps is guaranteed outbound for a policy, the inbound return traffic will have the same 256 Kbps bandwidth guarantee.

Maximum bandwidth specifies the total amount of bandwidth that can be allocated to the traffic specified in a given policy. This is the absolute ceiling for the traffic and cannot be exceeded. It's also very useful to specify bandwidth restrictions for protocols such as FTP, streaming media, or HTTP from specific hosts. This allows you to have hosts still use these bandwidth-intensive protocols, but restrict how much bandwidth they can use. The decision about which traffic gets how much bandwidth is based on these three concepts (priority queuing, guaranteed bandwidth, and maximum bandwidth).

An important factor in bandwidth allocation is what happens to all of the other traffic that matches policies that do not have traffic shaping configured on them. These factors

effect what happens to the unmatched traffic relative to the traffic matched to traffic shaping policies. All traffic that does not match an existing traffic-shaping policy will use the following configuration:

- **Guaranteed Bandwidth** No guaranteed bandwidth.
- **Maximum Bandwidth** Unlimited maximum bandwidth.
- **Priority** Lowest priority (priority 7).

NOTE

Traffic that is not already guaranteed by another policy is then serviced by priority queuing if it is enabled. Traffic with the highest priority will be serviced before any traffic with lower priority. If you have multiple policies with traffic at the same priority, that traffic will be serviced in a round robin fashion. Of course, queuing only happens when there is bandwidth contention. If you want a strict priority queuing environment, do not configure any guaranteed bandwidth.

Traffic shaping is a very complex process. Many factors come into play in regards to creating an effective traffic-shaping design. Not only must you consider the effects of guaranteed bandwidth, maximum bandwidth, and priority, you must also consider the protocol you are trying to shape. When shaping a protocol, understand how that protocol works first. If possible, do a study to determine bandwidth usage for each protocol. You may be surprised by how each protocol performs.

One particular protocol is HTTP. I have seen many organizations configure a poor traffic shaping policy around this protocol. Typically, the HTTP protocol does not use a great deal of bandwidth. If the Web site you are trying to traffic shape is mainly a text-based site with light images, the bandwidth consumption will be relatively low compared to a site delivering many images or multimedia content. When you access a Web site, you send a small amount of data that requests the content on the page. Afterward, the Web server delivers the requested information.

The user will review the information by reading the page or looking at the pictures, and can usually then click a second link on the site to access more content. For this type of Web site, there is no consistent passing of data. All the data is passed in bursts. Planning for this type of application can be tough because of its inconsistent distribution of data. This is where a study to determine exactly how much bandwidth is used would largely benefit you.

Damage & Defense…

Default Traffic Handling

It is possible for you to determine how the Juniper firewall handles the rest of the traffic that matches policies without traffic shaping enabled. You can tell your firewall to handle traffic using three modes. The default mode is *auto*. In auto mode, traffic shaping is turned on for all traffic the first time you configure a policy with traffic shaping. In the auto mode, if no traffic shaping is turned on, no traffic shaping is applied.

The second option you can configure is for traffic shaping to be on all the time. This enforces traffic shaping to all traffic, regardless of whether or not you have configured a policy with traffic shaping. This will apply the default traffic shaping options (no guaranteed bandwidth, unlimited maximum bandwidth, and lowest priority) to all traffic.

The last option is to disable traffic shaping on all traffic that is not part of a traffic-shaping policy. This mode is how the firewall acts when there are no traffic-shaping policies configured. To configure these options, you must use the CLI since these configuration changes cannot be made from the WebUI. In summary:

- **Traffic Shaping Auto** Sets traffic-shaping mode to auto.
- **Traffic Shaping always on** Sets traffic-shaping mode to on.
- **Traffic Shaping always off** Sets traffic-shaping mode to off.

```
Syngress-> get traffic-shaping mode
traffic shaping is set to auto by user
traffic shaping is currently turned off by the system
Syngress-> set traffic-shaping mode on
Syngress-> get traffic-shaping mode
traffic shaping is set to on by user
traffic shaping is currently turned off by the system
Syngress-> set traffic-shaping mode off
Syngress-> get traffic-shaping mode
traffic shaping is set to off by user
traffic shaping is currently turned off by the system
Syngress-> set traffic-shaping mode ?
auto                    automatically turn on/off traffic shaping
off                     turn off traffic shaping
```

Continued

```
on                    turn on traffic shaping
Syngress-> set traffic-shaping mode auto
Syngress-> get traffic-shaping mode
traffic shaping is set to auto by user
traffic shaping is currently turned off by the system
Syngress->save
```

NOTE

If you turn traffic shaping off for one policy while it remains on for another, the policy without traffic shaping will get the default policy. You can turn this off with the *set traffic-shaping mode off* command. This command is also useful if you want to set up traffic shaping but not enable it on the system.

Traffic-Shaping Examples

The Juniper firewall provides you with a wide range of traffic shaping capabilities to help manage traffic on your network. The best way to begin to understand traffic shaping is by example. In this section, we will cover a couple of traffic-shaping examples. These examples will help you better understand the application and use of traffic shaping.

Traffic-Shaping Example 1

In this example, we have a simple network setup: one firewall with a single trust and untrust interface. The company also has a single T1 with a 1.544-Mbps bandwidth. On the trust side of the network, we have a single IP block with two subnets. The Marketing department uses the 10.1.1.0/25 network, while the Research Services department uses the 10.1.1.128/25 network. Each department has different types of requirements. The Sales department has very little use for the Internet besides using e-mail. The Research Services department, on the other hand, has to perform research, most of which comes from using the Internet.

We have a possible contention of resources since the Marketing department has lately been using the Internet for streaming media, given it is inspirational for their work. This has slowed the production of the Research Services department and lowered their important productivity. You have decided to implement a traffic-shaping policy to ensure that the Research Services department is getting access to the resources they need. Table 5.2 shows our pseudo policy and what it does for us:

Table 5.2 Example 1: Pseudo Policy

Source	Destination	Service	Guaranteed Bandwidth	Maximum Bandwidth	Traffic Priority
Research Services					
10.1.1.128/25	Any	HTTP, HTTPS	512 Kbps	Unlimited	High
Marketing					
10.1.1.0/25	Any	Streaming Media	256 Kbps	512 Kbps	2nd
Entire Company					
10.1.1.0/24	Any	Any	512 Kbps	Unlimited	Low

We have set up three policies for the company. The first policy allows the Research Services department to access the Internet with the HTTP and HTTPS (HTTP Secure) protocols. This allows the department to access Web sites to acquire the information they need. We are guaranteeing 512 Kbps, or about one-third of the T1, because of the importance of this action. This traffic is given a high-priority tag to ensure it gets as much bandwidth as possible.

The second policy allows for the Marketing department to access streaming media. We guarantee that they will have 256 Kbps for streaming media protocol. However, in this policy, we also cap the total bandwidth they use to 512 Kbps. This traffic is given the second highest priority because upper management wants to ensure they have access to the streaming media.

The final policy covers the entire company for access to the Internet. On this policy, we use the entire network, 10.1.10/24, which encompasses both 10.1.1.0/25 (Marketing) and 10.1.1.128/25 (Research Services). We guarantee 512 Kbps for this traffic with no cap on how much bandwidth they can use. This traffic has the lowest possible priority. Let's look at the numbers and how the traffic breaks down for availability.

- **Guaranteed Bandwidth** Total guaranteed bandwidth 512 Kbps + 512 Kbps + 256 Kbps = 1280 Kbps. The available floating bandwidth, 264 Kbps, is left from the T1 in cases where all of the policies are using the maximum bandwidth.
- **Maximum Bandwidth** Only one policy is configured with maximum bandwidth. This is to ensure that the marketing department does not consume the entire T1.
- **Traffic Priority** The first policy will always get priority over the rest of the policies for any bandwidth remaining after all of the guaranteed bandwidth is used. The other policies will always get their guaranteed bandwidth regardless of the priority. If the first policy does not use the remaining bandwidth, it will first be available to the second policy, followed by the third policy.

Traffic-Shaping Example 2

Because of our excellent use of traffic shaping, the Research Services department has become extremely productive. This has allowed the company to grow and add new departments and, of course, new requirements:

- **Research Services** Lead researchers Darren, Rich, and Charlie have come up with the new requirements. The Research Services team needs access to FTP as well to utilize the usual HTTP and HTTPS protocols. Using the FTP protocol will not be as important as the HTTP and HTTPS protocols. Because the Research Services team has doubled its staff, they now are using the entire 10.1.1.0/24 network for their own department. The team has found the Internet responsive during its use.
- **Marketing Department** Patty has scolded the Marketing team for their lack of productivity and has denied them access to the streaming media services. However, the Marketing department now requires use of the Internet, much like Research Services. They need HTTP and HTTPS to identify new ways to be successful. The Marketing department uses the 10.1.2.0/25 network.
- **Human Resources** Nancy heads up the new Human Resources department. This department was created to work with all of the new employees. Nancy's only major requirement is to have her Human Resources application download new recruit information over FTP throughout the day. The files they need are small, but must be consistently delivered. The Human Resources department uses the 10.1.2.128/25 network.

We must determine an effective policy to maximize the T1 for this up-and-coming company. Table 5.3 shows the new policy. We have expanded on our original policy to include more policies.

Table 5.3 Example 2: Policy

Source	Destination	Service	Guaranteed Bandwidth	Maximum Bandwidth	Traffic Priority
Research Services					
10.1.1.0/24	Any	HTTP, HTTPS	512 Kbps	Unlimited	High
10.1.1.0/24	Any	FTP	128 Kbps	128 Kbps	High
Marketing					
10.1.2.0/25	Any	HTTP, HTTPS	256 Kbps	Unlimited	3rd

Continued

Table 5.3 Example 2: Policy

Source	Destination	Service	Guaranteed Bandwidth	Maximum Bandwidth	Traffic Priority
Human Resources					
10.1.2.128/25	Any	FTP	128 Kbps	128 Kbps	2nd
Entire Company					
10.1.1.0/24, 10.1.2.0/24	Any	Any	256 Kbps	Unlimited	Low

We continued with the theme of our original policies. The first policy still allows Research Services to access the Internet with the 512-Kbps guarantee. The second policy allows for Research Services to FTP to the Internet with a 128-Kbps guarantee. Because it is not as important as HTTP and HTTPS, we give FTP less bandwidth. Both of these policies have their traffic labeled as high priority.

The third policy allows the Marketing department to access the HTTPS and HTTP protocols. They have fewer people in their department, so they require less bandwidth. We have guaranteed the department 256 Kbps of bandwidth. The traffic from the Marketing department is not rated as important as either the Research Services or Human Resources traffic, but it is deemed more important than all of the other traffic coming from the company. Consequently, we have given this policy third priority.

The fourth policy is used to address the Human Resources department's requirement for FTP. The files for Human Resources are small and require very little bandwidth. We have guaranteed Human Resources 128 Kbps and have specified a maximum bandwidth of 128 Kbps. This will ensure that they get the available bandwidth, but does not allow them to capitalize on the rest of the available bandwidth. The last policy allows the rest of the company to access the Internet, guaranteeing them 256 Kbps. This traffic is not required for the company to function and has been given a low priority.

- **Guaranteed Bandwidth** Total guaranteed bandwidth is 512 Kbps + 128 Kbps + 256 Kbps + 128 Kbps + 256 Kbps = 1280 Kbps. The available floating bandwidth (264 Kbps) is left out of the T1 in cases where all the policies are using the maximum bandwidth.
- **Maximum Bandwidth** We have two separate policies with maximum bandwidth. These policies are used with maximum bandwidth to ensure they do not use up all of the available floating bandwidth.
- **Traffic Priority** The first two policies will always get priority over the rest of the priorities for any bandwidth remaining after all of the guaranteed bandwidth is used. The Human Resources FTP policy will get second priority to bandwidth.

Since this policy is already guaranteed bandwidth and the maximum bandwidth it can use is the same, the guarantee configuring the priority does not change much because it will already get the bandwidth guaranteed to it. The Marketing policy will be able to use any bandwidth left over that the research services team does not use. The rest of the company can use the guaranteed bandwidth of 256 Kbps, as well as any other bandwidth left over.

So far, we have reviewed the theory of traffic shaping. We will now look at the practical ways to configure the components of traffic shaping. You can configure policy-shaping in two places. First, bandwidth must be configured on the interfaces you intend to use traffic shaping on, and second, you must configure traffic shaping on each policy.

Interface Bandwidth

Configuring bandwidth for each interface is a simple process. You must first determine how much bandwidth you have for each connection. Traffic shaping is typically employed for the Internet, but it can be used anywhere in the network. If you do not configure the interface bandwidth manually, the firewall will assume the interface link as its bandwidth.

The following steps will guide you through the WebUI configuration:

1. Access **Network** | **Interfaces**.
2. Click the **Edit** link of the interface you wish to configure.
3. Use the **Traffic Bandwidth** field to enter the speed of the interface (in Kilobytes per second, or kBps).
4. Click **OK**.

Use the following commands to configure bandwidth via the CLI:

```
Syngress-> get interface untrust
Interface untrust:
  number 1, if_info 88, if_index 0, mode route
  link up, phy-link up/full-duplex
  vsys Root, zone Untrust, vr trust-vr
  dhcp client disabled
  PPPoE disabled
  *ip 214.208.253.9/24   mac 0010.db61.0e01
  *manage ip 214.208.253.9, mac 0010.db61.0e01
  route-deny disable
  ping disabled, telnet disabled, SSH disabled, SNMP disabled
  web disabled, ident-reset disabled, SSL disabled
  webauth disabled, webauth-ip 0.0.0.0
  OSPF disabled  BGP disabled  RIP disabled
  bandwidth: physical 100000kbps, configured 0kbps, current 0kbps
```

```
              total configured gbw 0kbps, total allocated gbw 0kbps
  DHCP-Relay disabled
  DHCP-server disabled
Syngress-> set interface untrust bandwidth 1544
Syngress-> get interface untrust
Interface untrust:
  number 1, if_info 88, if_index 0, mode route
  link up, phy-link up/full-duplex
  vsys Root, zone Untrust, vr trust-vr
  dhcp client disabled
  PPPoE disabled
  *ip 214.208.253.9/24   mac 0010.db61.0e01
  *manage ip 214.208.253.9, mac 0010.db61.0e01
  route-deny disable
  ping disabled, telnet disabled, SSH disabled, SNMP disabled
  web disabled, ident-reset disabled, SSL disabled
  webauth disabled, webauth-ip 0.0.0.0
  OSPF disabled  BGP disabled  RIP disabled
  bandwidth: physical 100000kbps, configured 1544kbps, current 0kbps
              total configured gbw 0kbps, total allocated gbw 0kbps
  DHCP-Relay disabled
  DHCP-server disabled
Syngress->save
```

Policy Configuration

Configuring traffic shaping on a policy is a simple process. The hard part is determining the configuration for each policy. In this example, we will configure traffic shaping on a policy that already exists.

Use the following steps to create a policy configuration via the WebUI:

1. Go to **Policies**.
2. Click the **Edit** link of the policy you want to modify.
3. Click the **Advanced** button at the bottom of the page (note that you can access the traffic shaping configuration by clicking the **Advanced** button when creating a new policy).
4. Enable the **Traffic Shaping** option.
5. Enter the desired **Guaranteed Bandwidth** (in kbps). A value of **0** indicates there is no guaranteed bandwidth configured.
6. Enter the desired **Maximum Bandwidth** (in kbps). A value of **0** indicates there is no maximum bandwidth configured.

7. Use the **Traffic Priority** drop-down list to select the desired priority. If you want to mark packets with DiffServ Codepoint Marking, enable the **DiffServ Codepoint Marking** option.
8. Click **OK**.

NOTE

You can only set traffic shaping on a policy when you create the policy. If you want to modify an existing policy, you must first delete it and then re-create it

The following commands are used for policy configuration via the CLI:

```
Syngress-> set policy from trust to untrust any any HTTP permit traffic gbw 100
priority 0 mbw 200 dscp enable
policy id = 2
Syngress-> get policy id 2
name:"none" (id 2), zone Trust -> Untrust,action Permit, status "enabled"
src "Any", dst "Any", serv "HTTP"
Policies on this vpn tunnel: 0
nat off, url filtering OFF
vpn unknown vpn, policy flag 4000, session backup: on
traffic shapping on, scheduler n/a, serv flag 00
log no, log count 0, alert no, counter no(0) byte rate(sec/min) 0/0
total octets 0, counter(session/packet/octet) 0/0/0
priority 0, diffserv marking On
tadapter: state on, gbw/mbw 100/200
No Authentication
No User, User Group or Group expression set
Syngress-> set policy from trust to untrust any any FTP permit traffic gbw 0
priority 0 mbw 200 dscp enable
policy id = 3
Syngress-> get policy id 3
name:"none" (id 3), zone Trust -> Untrust,action Permit, status "enabled"
src "Any", dst "Any", serv "FTP"
Policies on this vpn tunnel: 0
nat off, url filtering OFF
vpn unknown vpn, policy flag 4000, session backup: on
traffic shapping on, scheduler n/a, serv flag 00
log no, log count 0, alert no, counter no(0) byte rate(sec/min) 0/0
total octets 0, counter(session/packet/octet) 0/0/0
```

```
priority 0, diffserv marking On
tadapter: state on, gbw/mbw 0/200
No Authentication
No User, User Group or Group expression set
Syngress-> save
```

Configuring Traffic Shaping on a VPN Policy

To ensure you shape traffic within a tunnel properly, you must do so on the policy that is acting as the policy for the VPN. In this example, we will show how a policy-based VPN can be used to shape traffic that is traversing it. This example assumes you already have a VPN configured called TigersVPN.

To configure this example through the Juniper WebUI:

1. Go to **Policies** and select the appropriate **From Zone** and **To Zone** for the policy and click **New**.
2. Define the **Source** and **Destination Addresses**, along with the **Services** for the VPN policy.
3. Select **Tunnel** for the action, and choose the appropriate **VPN** from the drop-down list.
4. Define additional options for this policy (such as logging) and click the **Advanced** button.
5. Enable the **Traffic Shaping**, and define the appropriate parameters for it. This can include **Policing Bandwidth**, **Guaranteed Bandwidth**, **Maximum Bandwidth**, and **DSCP Marking**.
6. Click **OK**.

For this example, we will use the following parameters:

Policy	
From Zone	**Trust**
To Zone	**Untrust**
Source Address	**10.1.1.0/24**
Destination Address	**192.168.1.0/24**
Service	**FTP**
Action	**Tunnel**
VPN	**TigersVPN**
Logging	**Enabled**
Traffic Shaping	**Enabled**
Guaranteed Bandwidth	**200Kbps**
Maximum Bandwidth	**300Kbps**

To configure this example through the Juniper CLI:

set policy from trust to untrust 10.1.1.0/24 192.168.1.0/24 ftp tunnel vpn TigersVPN log traffic gbw 200 mbw 100

save

Configure Traffic Shaping on a Route-Based VPN

In this example, we will configure traffic shaping on a VPN which uses routes with tunnel interfaces instead of policies. Note that there are many different ways to perform similar traffic shaping functionality in a route-based VPN depending on where the tunnel interface is in respect to the ingress interface. In this example, we will assume that the tunnel1.1 interface is bound to the Trust interface, therefore it does not require a policy for traffic in the trust zone (since we are not using intrazone blocking for the Trust Zone). We will assume you have configured the VPN, bound it to the Tunnel1.1 Interface, and created routing for this example. We will enable traffic shaping on the ingress interfaces to ensure the VPN gets appropriate connectivity.

To configure this example through the Juniper WebUI:

1. Go to **Network | Interfaces** and click the **Edit** button next to the physical interface which the tunnel interface is bound to.
2. Define the **Ingress Maximum Bandwidth**.
3. Click **OK**.
4. Go to the tunnel interface by choosing **Network | Interface** and clicking **Edit** next to the tunnel interface you would like to configure traffic shaping on.
5. Define the **Ingress Maximum Bandwidth** for this interface which will be useful in making sure you don't send traffic faster than the WAN link, which supports what the VPN can handle.
6. Click **OK**.

In this example, we will use the following configuration:

Interface	**Trust**
Ingress Maximum Bandwidth	**100000Kbps**
Interface	**Tunnel1.1 (bound to Trust interface)**
Ingress Maximum Bandwidth	**1544Kbps**

To configure this example in the Juniper CLI:

set interface Trust bandwidth ingress mbw 100000

set interface tunnel1.1 bandwidth ingress mbw 1544

save

WARNING

You must be careful when applying traffic shaping to encapsulated traffic such as an IPSec VPN, or other tunneling protocols (GRE, PPTP, L2TP). Since the firewall either cannot (in the case of IPSec VPNs) or does not inspect the traffic within the tunnel, enabling traffic shaping on the traffic will apply it to the tunnel, but not the contents within it. This can particularly be an issue if you have latency-sensitive traffic through a VPN. Since the tunnel will be shaped rather than the traffic within it, you can have significant quality and performance degradation.

Enabling DSCP Class Mapping on the Firewall

In this example, we will enable DSCP Class Selection on the firewall. This can be enabled to ensure that traffic set with appropriate ToS bits by other devices in the network gets the appropriate service.

To configure this example in the Juniper WebUI:

1. Go to **Configuration | Advanced | Traffic Shaping**.
2. Select **DSCP Class Selection**.
3. Set the **IP Precedence** values according to your needs.
4. Select the appropriate **Mode**.
5. Click **OK**.

In this example, we will configure the following settings:

DSCP Class Selection	**Enabled**
IP Precedence	**Set as Defaults**
Mode	**Auto**

To configure this example in the Juniper CLI:

```
set traffic-shaping dscp-class-selector
save
```

Configuring DSCP Marking in a Policy

In this example, we will configure DSCP marking, which will be used to set appropriate ToS bits in the traffic matching that policy.

To configure this example in the Juniper WebUI:

1. Go to **Policies** and select the **From** and **To** zones for the policy and then click **New**.
2. Select the appropriate **Source** and **Destination** addresses.
3. Define the **Services** for this policy.
4. Specify the **Action** as well as any additional options for the policy, and then click **Advanced**.
5. Enable **Traffic Shaping**, and then enable the **DiffServ Code Marking** option.
6. Specify the **DSCP Value** that will be set on the traffic which passes through the policy.
7. Click **OK**.

In this example, we will use the following settings:

Policy	
From Zone	**Trust**
To Zone	**Untrust**
Source Address	**10.5.0.0/16**
Destination Address	**192.168.2.0/24**
Service	**SIP**
Action	**Permit**
Logging	**Enabled**
Traffic Shaping	**Enabled**
Diffserv Code Marking	**Enabled**
DSCP Value	**7**

To configure this example in the CLI:

set policy from trust to untrust 10.5.0.0/16 192.168.2.0/24 sip permit log traffic dscp enable value 7

save

Advanced Policy Options

Several options on a Juniper firewall are considered *advanced* options. They are not necessarily more complex, they are more like miscellaneous options that don't fit into a particular category. All of these options are invoked directly on the policy much like how we configured traffic shaping earlier.

In this section, we look at counting and scheduling. Counting provides the option to track bandwidth that is used on a per-policy basis. This can be helpful in determining an effective traffic shaping policy. Scheduling allows you to set times at which a policy is active. Typically, once a policy is created, it is always in effect until you delete it or disable it. Scheduling allows you to specify the times at which a policy is active. This can be particu-

larly useful with traffic shaping. For instance, you can configure certain traffic-shaping policies to be active at certain times of day, while others are active during different hours.

In this section, we cover most of the available advanced options, while some of the other features require more in-depth coverage, and so we set aside their own chapters for such discussions. Because of the breadth of knowledge involved in the various options of user authentication, it has been given its own chapter. Other advanced options omitted from this chapter are NAT (covered in Chapter 8) as well as Anti-Virus, Deep Inspection, Anti-Spam, and URL Content Filtering (covered in Chapter 10).

Counting

The counting feature allows you to display a graphical view of traffic that passes through the policy. This can be useful in determining traffic usage for a specific policy. It also can assist you in determining effective traffic-shaping policies. Counting can be enabled on any policy. When using counting, you can also enable something called a *traffic alarm*, which is a threshold for the policy in Bytes per second, KB per Minute, KB per Hour, MB per Day, or MB per Month. If the threshold is exceeded, a traffic alarm will be generated and can be sent to you via e-mail. The traffic alarm is also logged.

In Figure 5.3, you can see an example of a graph that is generated by configuring counting. At the top of the page is a drop-down list labeled *Granularity*. This allows you to choose one of the following display units:

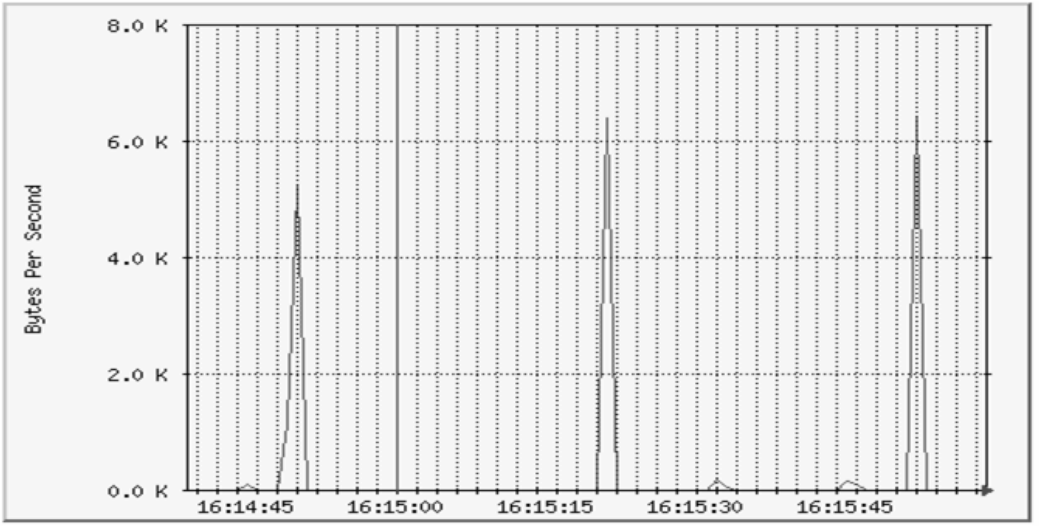
- Bytes Per Second
- Kilobytes Per Minute
- Kilobytes Per Hour
- Megabytes Per Day
- Megabytes Per Month

It is also possible to download the data in a text file. An example of the text file is listed next. You can use this text file to generate your own reports with the data.

```
========================================================================

Second Counters Log for Policy:

   (Src = "Any", Dst = "Any", Service = "ANY")

    Current system time is Sat, 16 Oct 2004 15:14:14
========================================================================

                Time Stamp   Counter ( Bytes )
```

```
Sat, 16 Oct 2004 15:14:14   0000036229
Sat, 16 Oct 2004 15:14:13   0000034407
Sat, 16 Oct 2004 15:14:12   0000023846
Sat, 16 Oct 2004 15:14:11   0000029281
```

Figure 5.3 A Counting Graph Example



You can look at traffic alarms from both the WebUI and the CLI. A report is generated for every time period in which the traffic exceeds the set threshold. Figure 5.4 shows an example of an alarm report from the WebUI.

Figure 5.4 A Traffic Alarm Report (WebUI)

Traffic log for policy :

ID	Source	Destination	Service	Action
1	Trust/Any	Untrust/Any	ANY	Permit

Date/Time	Event	Details
2004-10-16 21:42:52	Minute Rate Alarm	Rate = 3578 KBytes/Min. is over threshold.
2004-10-16 21:41:53	Minute Rate Alarm	Rate = 4279 KBytes/Min. is over threshold.
2004-10-16 21:40:54	Minute Rate Alarm	Rate = 4511 KBytes/Min. is over threshold.
2004-10-16 21:39:55	Minute Rate Alarm	Rate = 4200 KBytes/Min. is over threshold.
2004-10-16 21:38:56	Minute Rate Alarm	Rate = 3938 KBytes/Min. is over threshold.
2004-10-16 21:37:57	Minute Rate Alarm	Rate = 3874 KBytes/Min. is over threshold.
2004-10-16 21:36:58	Minute Rate Alarm	Rate = 4127 KBytes/Min. is over threshold.
2004-10-16 21:35:59	Minute Rate Alarm	Rate = 5130 KBytes/Min. is over threshold.
2004-10-16 21:34:29	Minute Rate Alarm	Rate = 2476 KBytes/Min. is over threshold.
2004-10-16 21:33:30	Minute Rate Alarm	Rate = 2387 KBytes/Min. is over threshold.
2004-10-16 21:32:31	Minute Rate Alarm	Rate = 2288 KBytes/Min. is over threshold.
2004-10-16 21:31:32	Minute Rate Alarm	Rate = 2184 KBytes/Min. is over threshold.
2004-10-16 21:30:33	Minute Rate Alarm	Rate = 2362 KBytes/Min. is over threshold.
2004-10-16 21:29:34	Minute Rate Alarm	Rate = 3221 KBytes/Min. is over threshold.

To view a policy's traffic alarm reports via the CLI, click the policy's red alert light, or select **Reports** | **Policies**.

The following traffic alarm information was accessed via the CLI:

```
Syngress-> get alarm traffic
Recent Alarm Time     PID Source               Destination          Service
2004-10-16 21:47:57     1 Any                  Any                  ANY
Total entries matched = 1
Syngress-> get alarm traffic detail
PID 1, src Any, dst Any, service ANY
Total alarm entries under this policy = 4095
Date       Time               Rate      Threshold   Unit
2004-10-16 21:47:57           2902              5   KBytes/Minute
2004-10-16 21:46:58           3442              5   KBytes/Minute
2004-10-16 21:45:59           4443              5   KBytes/Minute
2004-10-16 21:44:50           3164              5   KBytes/Minute
2004-10-16 21:43:51           3235              5   KBytes/Minute
2004-10-16 21:42:52           3578              5   KBytes/Minute
2004-10-16 21:41:53           4279              5   KBytes/Minute
2004-10-16 21:40:54           4511              5   KBytes/Minute
2004-10-16 21:39:55           4200              5   KBytes/Minute
2004-10-16 21:38:56           3938              5   KBytes/Minute
2004-10-16 21:37:57           3874              5   KBytes/Minute
2004-10-16 21:36:58           4127              5   KBytes/Minute
2004-10-16 21:35:59           5130              5   KBytes/Minute
2004-10-16 21:34:29           2476              5   KBytes/Minute
2004-10-16 21:33:30           2387              5   KBytes/Minute
2004-10-16 21:32:31           2288              5   KBytes/Minute
2004-10-16 21:31:32           2184              5   KBytes/Minute
2004-10-16 21:30:33           2362              5   KBytes/Minute
2004-10-16 21:29:34           3221              5   KBytes/Minute
Total entries matched = 19
```

Configuring Counting

Configuring counting is simple. Counting can be enabled or disabled at any time. When you configure counting, it is either turned on or off. In this example, we will enable counting on a policy that already exists.

Use the following steps to enable counting via the WebUI:

1. Go to Policies and click the desired policy's **Edit** link.
2. Click **Advanced** at the bottom of the page.

3. Enable the **Counting** option.
4. Click **OK**.

Use the following scripts to enable counting via the CLI:

```
Syngress-> set policy from trust to untrust any any HTTP permit count
policy id = 2
Syngress-> get policy id 2
name:"none" (id 2), zone Trust -> Untrust,action Permit, status "enabled"
src "Any", dst "Any", serv "HTTP"
Policies on this vpn tunnel: 0
nat off, url filtering OFF
vpn unknown vpn, policy flag 0000, session backup: on
traffic shapping off, scheduler n/a, serv flag 00
log no, log count 0, alert no, counter yes(2) byte rate(sec/min) 0/0
total octets 0, counter(session/packet/octet) 0/0/2
priority 7, diffserv marking Off
tadapter: state off, gbw/mbw 0/-1
No Authentication
No User, User Group or Group expression set
Syngress-> set policy id 2
Syngress(policy:2)-> unset count
Syngress(policy:2)-> exit
Syngress-> get policy id 2
name:"none" (id 2), zone Trust -> Untrust,action Permit, status "enabled"
src "Any", dst "Any", serv "HTTP"
Policies on this vpn tunnel: 0
nat off, url filtering OFF
vpn unknown vpn, policy flag 0000, session backup: on
traffic shapping off, scheduler n/a, serv flag 00
log no, log count 0, alert no, counter no(0) byte rate(sec/min) 0/0
total octets 0, counter(session/packet/octet) 0/0/0
priority 7, diffserv marking Off
tadapter: state off, gbw/mbw 0/-1
No Authentication
No User, User Group or Group expression set
Syngress-> set policy id 2
Syngress(policy:2)-> set count
Syngress(policy:2)-> exit
Syngress->
```

In Figure 5.5, you can see the icon that is added to your policy. You can click this icon to access the counting graph, which is represented here as an hourglass. You are only able to

view the graphs from the WebUI. If you are using the CLI, you can see the stored counter information in its raw form, but it is of very little help in actually determining the traffic usage.

Figure 5.5 Policy with Traffic Shaping Configured (WebUI)

The following is an example of raw counter information, obtained via the CLI:

```
Syngress-> get counter policy 1 second
PID: 1, Interval: Second, Unit: Byte/Sec, End Time: 16 Oct 2004 16:14:39
000-005: 0000039654 0000035190 0000034479 0000042527 0000029679 0000047886
006-011: 0000033058 0000034236 0000042506 0000032629 0000041460 0000042747
012-017: 0000045812 0000051081 0000067825 0000057319 0000055379 0000043726
018-023: 0000061160 0000072803 0000058361 0000066299 0000073356 0000072003
024-029: 0000076061 0000091056 0000084565 0000064143 0000047321 0000061755
030-035: 0000051065 0000062170 0000046592 0000060783 0000057485 0000079750
036-041: 0000053997 0000044322 0000045913 0000000000 0000056328 0000061494
042-047: 0000052587 0000041281 0000048066 0000055305 0000048326 0000045536
048-053: 0000043505 0000043834 0000047886 0000049541 0000050748 0000048746
054-059: 0000051015 0000067368 0000039355 0000041967 0000039633 0000047315
060-065: 0000066774 0000060505 0000054568 0000046993 0000051292 0000054856
066-071: 0000061414 0000044580 0000035620 0000035112 0000043073 0000041217
072-077: 0000046928 0000055871 0000050939 0000033101 0000035341 0000032518
078-083: 0000031710 0000035645 0000036502 0000042580 0000047418 0000031568
084-089: 0000045538 0000045069 0000048985 0000055465 0000036345 0000055489
090-095: 0000063875 0000049474 0000050028 0000037453 0000040042 0000036762
096-101: 0000028722 0000042958 0000040367 0000000000 0000052461 0000041931
102-107: 0000044813 0000038372 0000049706 0000050366 0000046635 0000036129
108-113: 0000041911 0000042353 0000038854 0000030692 0000037721 0000028314
114-118: 0000040465 0000025109 0000056224 0000040654 0000053751
Syngress->
```

Configuring Traffic Alarms

To configure traffic alarms, first determine what values you want to monitor. You can choose to use Bytes per second, Kilobytes per minute, or both. Use 0 for any option you do not wish to use. Traffic alarms can be configured from both the CLI and the WebUI. You can click this red icon to access the report for that policy. If you are using the CLI, you can con-

figure traffic alarms both during policy creation, or after the policy has been created. Note that you must have counting enabled in order to enable traffic alarms.

Use the following steps to configure traffic alarms via the WebUI:

1. Click the **Edit** link of the policy you want to modify.
2. Click **Advanced**.
3. In the **Alarm Threshold** section, use the **Bytes/Sec** field to enter the bytes per second you wish to monitor. If you do not wish to use this option, leave the field blank
4. Use the **KBytes/Min** field to enter the Kilobytes per minute you wish to monitor. If you do not wish to use this option, leave the field blank.
5. Click **OK**.

Use the following scripts to configure traffic alarms via the CLI:

```
Syngress-> set policy from trust to untrust any any FTP permit count alarm 0
256
policy id = 2
Syngress-> get policy id 2
name:"none" (id 2), zone Trust -> Untrust,action Permit, status "enabled"
src "Any", dst "Any", serv "FTP"
Policies on this vpn tunnel: 0
nat off, url filtering OFF
vpn unknown vpn, policy flag 0000, session backup: on
traffic shapping off, scheduler n/a, serv flag 00
log no, log count 0, alert no, counter yes(2) byte rate(sec/min) 0/256
total octets 0, counter(session/packet/octet) 0/0/2
priority 7, diffserv marking Off
tadapter: state off, gbw/mbw 0/-1
No Authentication
No User, User Group or Group expression set
Syngress-> set policy id 2
Syngress(policy:2)-> unset count
Syngress(policy:2)-> set count alarm 500 512
Syngress(policy:2)-> exit
Syngress-> get policy id 2
name:"none" (id 2), zone Trust -> Untrust,action Permit, status "enabled"
src "Any", dst "Any", serv "FTP"
Policies on this vpn tunnel: 0
nat off, url filtering OFF
vpn unknown vpn, policy flag 0000, session backup: on
traffic shapping off, scheduler n/a, serv flag 00
```

```
log no, log count 0, alert no, counter yes(2) byte rate(sec/min) 500/512
total octets 0, counter(session/packet/octet) 0/0/2
priority 7, diffserv marking Off
tadapter: state off, gbw/mbw 0/-1
No Authentication
No User, User Group or Group expression set
Syngress->save
```

Scheduling

When you create a policy on a Juniper firewall, you immediately activate that policy into the running configuration. If you do not want to use that policy, you can either disable it or remove it manually. Scheduling is a function that allows you to have a policy that is active only at specific times. You would create a schedule object based upon a single time, day, or reoccurring time.

For example, you may want to allow your users to browse the Internet after 5 P.M. By creating a schedule object, you can define a time or times that you want to allow this activity. A schedule object can be created to occur at a single time or on a recurring schedule. When configuring scheduling, the time is based upon the local time of the firewall.

Scheduling Properties

Scheduling objects allow you to configure several options which can be used to define when a policy is active on the firewall. The following properties can be configured within a schedule object on the firewall:

- **Schedule Name** This is the name for the schedule object which will be referenced elsewhere in the configuration.
- **Comment** You can define a comment to this schedule name for reference purposes.
- **Recurring** If this schedule object should recur on a weekly basis, you can define it here.
- **Week Day** You can specify two times per day per schedule object that the policy will be enabled and then disabled. You do this on a daily basis.
- **Once** If the event should only occur once at a scheduled time and then shut off, you can define it here. You define the start and end time in terms of *mm/dd/yyyy hh:mm*.

Configuring Scheduling

Configuring a policy to schedule is a two-step process. First, you must create a schedule object. Next, you must apply the schedule object to a policy. You can apply the scheduling object to an existing policy or to a policy as it is being created. You can create and apply scheduling objects from both the CLI and the WebUI. In this example, we will create a schedule object.

Creating a schedule object requires the use of a name and the definition of either a recurring or a one-time instance. If you configure a recurring time, you can configure two different periods per day. Any days that you do not want to apply a schedule to, leave those days blank. To configure a single occurrence, you must configure a start and stop time along with a start and stop date.

Use the following steps to add a schedule object via the WebUI:

1. Go to **Objects | Schedule** and click **New**.
2. Enter the name of the object in the **Schedule Name** field.
3. Enter a brief description in the **Comment** field.
4. Select either **Recurring** or **Once**.
5. Enter the start and end times for the schedule object.
6. Click **OK**.

Use the following steps to edit an existing schedule object:

1. Access **Objects** | **Schedules**.
2. Click the **Edit** link of the schedule you wish to edit.
3. Make the desired changes and click **OK**.

To remove a schedule object:

1. Access **Objects | Schedules**.
2. Click the **Remove** link of the schedule object you wish to delete.
3. Click **OK** to confirm. (Note that you cannot delete an object that is used in a policy.)

The following scripts are used for configuring scheduling via the CLI:

```
Syngress-> set scheduler "Upgrade Period" once start 08/02/2004 12:00 stop
11/14/2004 12:00 comment "The will allow for contractor access"
Syngress-> get scheduler
One-time Schedules:
Name                 Start Time          Stop Time           Comments
Upgrade Period       08/02/2004 12:00    11/14/2004 12:00    The will allow f
```

```
Syngress-> set scheduler "After Hours" recurrent monday start 17:00 stop 19:00
Syngress-> set scheduler "After Hours" recurrent tuesday start 17:00 stop 19:00
Syngress-> set scheduler "After Hours" recurrent wednesday start 17:00 stop
19:00
Syngress-> set scheduler "After Hours" recurrent thursday start 17:00 stop 19:00
Syngress-> set scheduler "After Hours" recurrent friday start 17:00 stop 19:00
Syngress-> get scheduler
One-time Schedules:
Name                 Start Time           Stop Time            Comments
Upgrade Period       08/02/2004 12:00     11/14/2004 12:00     The will allow f

Recurrent schedules:
Name           Weekday       Start1   Stop1   Start2   Stop2   Comments
After Hours    Monday        17:00    19:00   N/A      N/A
After Hours    Tuesday       17:00    19:00   N/A      N/A
After Hours    Wednesday     17:00    19:00   N/A      N/A
After Hours    Thursday      17:00    19:00   N/A      N/A
After Hours    Friday        17:00    19:00   N/A      N/A

Syngress->save
```

NOTE

Even though it seems as if there are multiple objects named *After Hours*, they all represent the same object.

Once you have created your service objects, you can now apply them to your policy. If a policy has a schedule, but the policy is currently active, there is no way to tell if the policy has scheduling configured from the main policies page. The only way is to drill down on the policy to the advanced configuration page.

Use the following steps to apply scheduling to a policy via the WebUI:

1. Go to **Policies** and click the **Edit** link of the policy you want to modify.
2. Click **Advanced**.
3. Use the **Schedule** drop-down list to select the schedule object you want to apply to the current policy.
4. Click **OK**.

The following scripts are used to configure a policy for scheduling via the CLI:

```
Syngress-> set policy from trust to untrust any any HTTP permit schedule "After
Hours"
policy id = 3
Syngress-> get policy id 3
name:"none" (id 3), zone Trust -> Untrust,action Permit, status "enabled"
src "Any", dst "Any", serv "HTTP"
Policies on this vpn tunnel: 0
nat off, url filtering OFF
vpn unknown vpn, policy flag 0000, session backup: on
traffic shapping off, scheduler After Hours(off), serv flag 00
log no, log count 0, alert no, counter no(0) byte rate(sec/min) 0/0
total octets 0, counter(session/packet/octet) 0/0/0
priority 7, diffserv marking Off
tadapter: state off, gbw/mbw 0/-1
No Authentication
No User, User Group or Group expression set
Syngress->save
```

WARNING

It is critical that you ensure the date and time on the firewall is correct in order for schedule objects to work properly. Since the firewall references the date and time in the firewall to ensure the schedule is enabled at the appropriate time, the time must be accurate. If it is not, the schedules will be applied at different times then what you are expecting!

Configuring Schedule and Traffic Shaping

One of the powerful uses of scheduling is to use it in conjunction with scheduling. For instance, you can configure one policy to be active during a part of the day when there are certain traffic patterns, and another policy to be active during another part of the day. That's exactly what we will be doing in this example.

To configure this example in the Juniper WebUI:

1. Create the schedule objects for the policy by going to **Objects | Schedules** and clicking **New**.
2. Specify the **Recurring** option and the appropriate days and times for the schedule. For this example we will create one schedule for Monday through Friday which will have times 9 A.M. to 5 P.M.

3. Click **OK**.
4. Create a policy which does not use scheduling by going to **Policies** and selecting the appropriate **From Zone** and **To Zone** for the policy and clicking **New**.
5. Define the **Source** and **Destination Addresses**, along with the **Services** for the VPN policy.
6. Select **Permit** for the action.
7. Define additional options for this policy (such as logging) and click the **Advanced** button.
8. Enable **Traffic Shaping**, and define the appropriate parameters for it. This can include **Policing Bandwidth**, **Guaranteed Bandwidth**, **Maximum Bandwidth**, and **DSCP Marking**.
9. Select the **Schedule** from the drop-down menu.
10. Click **OK**.
11. In the policy window, you can simply click the **Clone** hyperlink.
12. Go to the **Advanced** section and turn off **Scheduling** and set the **Traffic Shaping** you would like to have for the traffic outside the scheduled policy window.
13. Click **OK**.

For this example, we will want to make sure the policy with the schedule is above the policy without. This way, traffic will match the scheduled policy when it is enabled, but when it is not, it will match the traffic for the other policy, which is any other time.

In this example, we will use the following properties:

Schedule	**Schedule1**
Monday-Friday	**09:00-17:00**

Policy	
ID	**1**
From Zone	**Trust**
To Zone	**Untrust**
Source Address	**Any**
Destination Address	**Any**
Service	**HTTP**
Action	**Permit**
Logging	**Enabled**
Traffic Shaping	**Enabled**
Maximum Bandwidth	**100kbps**

Schedule	**Schedule1**
ID	**2**
From Zone	**Trust**
To Zone	**Untrust**
Source Address	**Any**
Destination Address	**Any**
Service	**HTTP**
Action	**Permit**
Logging	**Enabled**
Traffic Shaping	**Enabled**
Maximum Bandwidth	**1000kbps**

To configure this example in the Juniper CLI

set scheduler "Schedule1" recurrent monday start 9:0 stop 17:0

set scheduler "Schedule1" recurrent tuesday start 9:0 stop 17:0

set scheduler "Schedule1" recurrent wednesday start 9:0 stop 17:0

set scheduler "Schedule1" recurrent thursday start 9:0 stop 17:0

set scheduler "Schedule1" recurrent friday start 9:0 stop 17:0

set policy id 1 from "Trust" to "Untrust" "Any" "Any" "HTTP" permit schedule "Schedule1" log traffic mbw 100

set policy id 2 from "Trust" to "Untrust" "Any" "Any" "HTTP" permit log traffic mbw 1000

save

Summary

In this chapter, we looked at the various advanced policy options. As mentioned at the beginning of the chapter, these are more like miscellaneous options than options that are truly advanced or complex.

Traffic shaping is what should be considered an advanced feature. To successfully implement traffic shaping, you must research your exact requirements for each policy you implement. A poor configuration with traffic shaping can be as bad in some cases as not using traffic shaping at all. We looked at the use of guaranteed bandwidth, traffic prioritization, and maximum bandwidth restrictions. These three components of traffic management allow you to ensure that the available bandwidth is allocated to the policies that need it.

Counting, although simple to configure, can yield some powerful results. Using counting can help you identify traffic patterns on a per-policy basis. This can help you identify which policies require more bandwidth and which use less bandwidth. You can use this information to make more informed decisions when configuring traffic shaping.

Policies are typically in one of two states: enabled or disabled. When using scheduling, you can create schedule objects to specify which times the policies will be enforced and for how long policies are effective. This allows for access or denial of resources in your network based upon the time of day, thus allowing you to have a more dynamic set of firewall policies.

The advanced topics that we covered in this chapter can empower you to configure the firewall above and beyond a traditional stateful firewall device. They allow you to granularly and efficiently control the traffic on your network, even based upon the time of day. Since traffic patterns change frequently throughout the day, this can be a great asset for you to deploy. A good working knowledge of the state and traffic load of the firewall is also important. The Juniper firewall can provide you with plenty of details to ensure that your firewall is functioning as expected, and will make your job as an administrator much more pleasant.

Solutions Fast Track

Matt: should there be a section here titled "Traffic-Shaping Fundamentals" along with any associated bullet points? Also, is the "Network Traffic Management" heading below really supposed to read "Deploying Traffic Shaping on Juniper Firewalls", as cited at the beginning of the chapter? The three bullets at the beginning of the chapter should match those here at the end. Mike

Traffic-Shaping Fundamentals

- ☑ To create an effective traffic-shaping policy, it will take time and research in order to properly utilize the various features of traffic shaping.
- ☑ Guaranteed bandwidth is always allocated first, even on the lowest priority level.

- ☑ Bandwidth that is allocated is always allocated bidirectionally.
- ☑ Maximum bandwidth allows you to put a cap on how much total bandwidth a policy can use.
- ☑ By applying a priority to traffic matched to a policy, you are deciding for the firewall which traffic should be allocated more bandwidth and which traffic is nonessential.
- ☑ Priority-based queuing is good for interactive and streaming data such as voice. Guaranteed bandwidth is best for bursty traffic such as HTTP and SMTP.

Advanced Policy Options

- ☑ Counting can assist in creating an effective traffic-shaping policy.
- ☑ Using authentication in your policies allows you to help ensure that the person using the policies resources is authorized to use them.
- ☑ Scheduling allows you to configure a policy that is effective during specific times of the day.
- ☑ You can utilize both scheduling and traffic shaping to deploy a firewall which changes with the traffic patterns of your network at different times of the day.
- ☑ Using traffic alarms for counting can let you know when you are getting close to utilizing all of your bandwidth for your Internet link. Having a good working knowledge of your environment is definitely important to ensure that your network is functioning as you expect, and that there are no surprises.

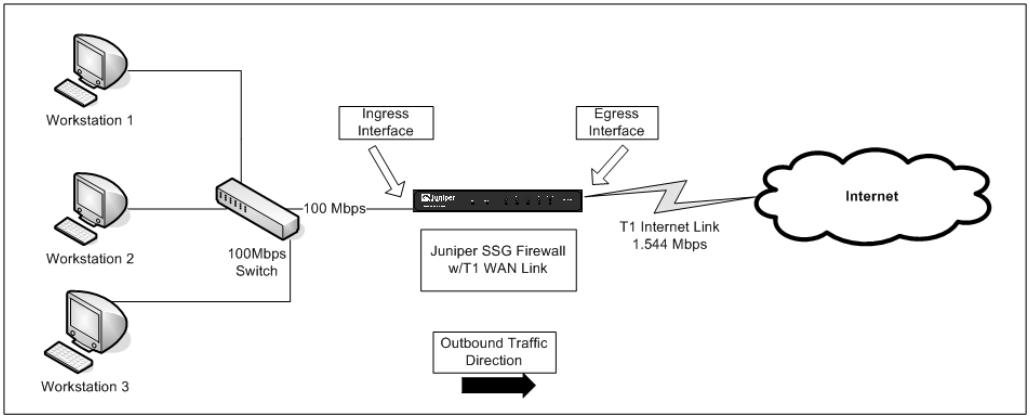
Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: What's a good way to remember the difference between Egress and Ingress traffic on an interface?

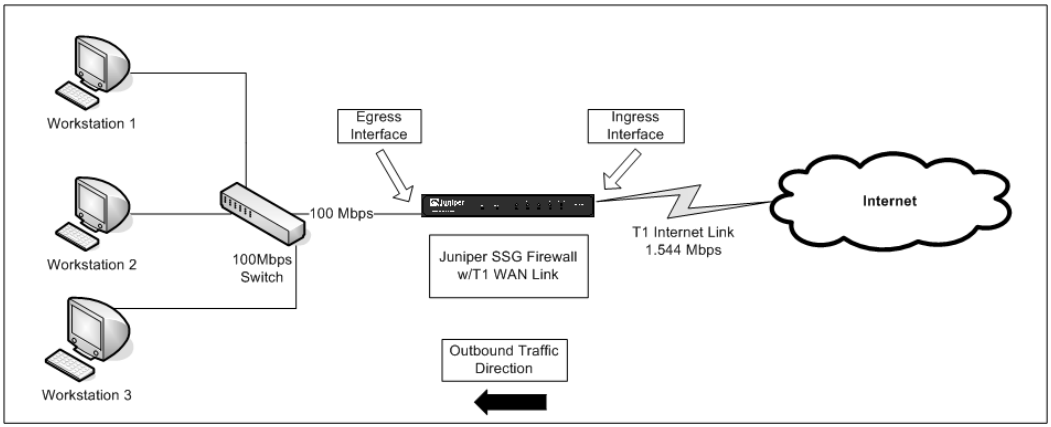
A: The definitions of Egress and Ingress interfaces can be a bit confusing since it really depends on which way traffic is flowing. When traffic is arriving on an interface from a physical link, this is known as the Ingress traffic. On the other hand, when traffic is leaving an interface to be sent onto a physical link, this is known as Egress traffic. So really, an interface supports both Egress and Ingress traffic, it just depends on which direction the traffic is flowing in order to determine how the traffic is limited by egress and ingress bandwidth settings on an interface. Figure 5.6 shows the traffic going outbound in one direction, with the Ingress and Egress interfaces labeled, while Figure 5.7 shows the traffic going in the opposite direction and the respective Ingress and Egress interfaces.

Figure 5.6 Ingress/Egress Interfaces with Respect to Traffic Direction



Q: When should I use Ingress Bandwidth settings versus Egress Bandwidth settings?

Figure 5.7 Ingress/Egress Interfaces with Respect to Traffic Direction



A: Ingress bandwidth settings should be configured as close to the source of the traffic as possible (meaning not on the outbound interface). This will help minimize the processing the firewall must do on the traffic since it will be dropped earlier in the packet flow if there is contention. Sometimes, you may have multiple interfaces which may contribute to the overall outbound packet flow. Let's say you have an internal Trust Interface for your LAN, and a DMZ interface which both send traffic through the Untrust interface that connects to a T1. In these situations, you might want to also limit the traffic that can be sent on the Egress settings of the Untrust interface so it doesn't pass more than 1.544 Mbps onto the T1. In such a situation, you wouldn't want to configure Ingress policing on the Trust or DMZ interfaces to 1.544 Mbps because it could still turn out to be more than the T1 link could handle if both interfaces were sending at even half capacity. Also, if you set ingress policing on those interfaces to a low value, they would not be able to send full bandwidth speeds between themselves (such as 100 Mbps for Fast Ethernet). So basically, in that situation, you would want to restrict the Untrust "Egress" Interface with Egress Bandwidth settings, and the Trust and DMZ interfaces with Ingress settings of 100 Mbps, assuming they are Fast Ethernet. See Figure 5.8 for a visual representation of this example.

Q: Is there a point to configuring traffic shaping with a reject or deny?

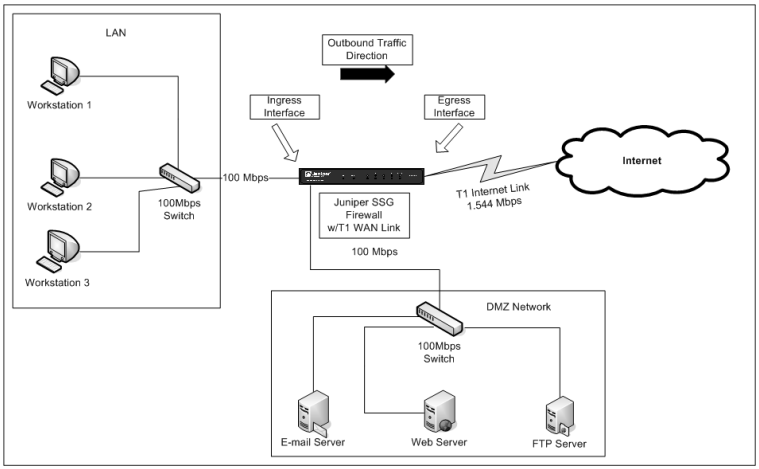
A: No, this would not be very effective since you are going to be dropping this traffic anyways. You should only configure traffic shaping on policies where you either permit or tunnel the traffic.

Q: Why is traffic shaping so difficult to use?

A: Traffic shaping requires some reasonable planning to use. When you configure traffic shaping, you are setting up rules that the firewall must follow when prioritizing traffic.

Because the firewall is unable to make cognitive decisions, you must determine all of the decisions that need to be made for traffic shaping up front. This can be difficult because it requires some planning, but once you have tuned an effective traffic shaping configuration on your firewall, the results will be well worth the effort.

Figure 5.8 LAN and DMZ Bandwidth Shaping



Q: Traffic shaping does not seem to be as fully featured as I think it should be. Why would Juniper even use it on their firewalls?

A: The traffic shaping option on the Juniper firewall is an excellent tool for traffic management. Because it is only one small part of what the firewall product can do, it is not the focus of the product. Many products exist solely to do traffic shaping, and those products excel at providing that type of capability. The traffic shaping option on a Juniper firewall provides the minimum required options to be able to support traffic shaping effectively.

Q: Does the use of counting affect the performance of your firewall?

A: If you enable counting on your firewall it will cause a slight performance decrease because of all the internal operations the firewall must perform to store the counting information. However, the impact is minimal. The impact would only be noticeable if you are already running your firewall at peak capacity. In most situations, counting can be enabled on your policies with little performance detriment.

Q: Scheduling looks like a great tool, but can you use it in a policy to deny traffic?

A: The scheduling option can be enabled on any policy, regardless of what that policy does. The policy can be a deny policy, or even contain VPNs. The action or content of the policy does not affect the ability to enable scheduling on a policy.

Chapter 6

User Authentication

Solutions in this chapter:

- User Account Types
- Local and External Authentication Servers
- Policy-Based User Authentication
- 802.1x Authentication
- Authentication Enhancements

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

User authentication is one of the core principles of security. Juniper firewalls have an extensive set of user authentication capabilities that provide you with many options to strengthen the security of your network. Traditionally, user authentication has been used as a tool to help identify and validate the identity of a particular user. Modern networks demand much more than simple user identification, however. With the security of your network on the line, you have a lot to lose by not verifying and restricting who is accessing your resources. Juniper has recognized and addressed the difficulties that administrators face trying to harden their networks, while balancing the needs of users and remaining flexible.

At first glance, some administrators may feel overwhelmed with the number of authentication options the Juniper firewall is capable of. Not only does Juniper provide you with its own internal database, you can also integrate other standardized and proprietary authentication systems. The wide range of capabilities that you have at your disposal will enable you to harden your perimeter, and also provide peace of mind that you can track and audit network activity.

In this chapter, we will begin by discussing the different types of user accounts. Second, we will cover the different authentication servers that can be used to authenticate users through different access methods. Third, we will delve into policy-based user authentication, which will enable you to control every aspect of user access to your network. We will follow our policy-based authentication discussion with an in-depth look at 802.1x authentication on the Juniper firewall. This is a relatively new technology that allows you to control user access right down to the Data-Link layer. Lastly, we will cover some of the extensions to the authentication facilities, which can provide you with a more powerful means to administer access. By the end of this chapter, you should have a thorough understanding of the different methods to authenticate users, and how to implement and manage each of these solutions.

User Account Types

Juniper firewalls support several different user account types. Each account type fulfills a specific functional role and has specific attributes. It really helps to understand the capabilities of each user type, especially when you have the option of choosing one type over another. The different user account types are outlined in this section. We will cover them in great depth, explaining the differences between them, and the attributes specific to each type. The different account types include the following:

- Admin user
- Auth user
- IKE user
- XAuth user
- L2TP user

- 802.1x

Perhaps the most important user in any firewall is the Administrator. This account has many unique properties which set it apart from other user accounts. Administrator accounts, or Admin accounts for short, are special accounts which have permission to view and/or modify the configuration of the firewall. They even have their own database when stored on the firewall. Admin accounts cannot be used for any other purpose (such as VPN authentication). There are technically five types of Administrator accounts, all of which are covered next. We will follow each explanation with an example to help put your knowledge to work.

Admin Account Types

- **Root Admin** The root admin account is the most privileged account. It can create and delete other admin accounts, virtual systems (VSYS), and modify certain aspects of virtual systems. This account can only be stored locally on the firewall, and there must be at least one root admin.
- **Root-Level Read/Write Admin** This administrator has all of the privileges of the Root admin except it cannot create and delete VSYS, along with modifying certain aspects of VSYS. The reason why it is called a Root-Level account is because it is not an administrator account created within a VSYS (discussed next). Root-Level accounts may be stored both locally and externally.
- **Root-Level Read-Only Admin** This administrator can only view the configuration of the firewall; it cannot modify any of it. Additionally, some debugging commands are restricted. This account may be stored both locally and externally.
- **VSYS Read/Write Admin** VSYS Admin accounts have read/write access only to the VSYS to which they have been assigned. They cannot view any other VSYS, nor can they view the root-level configuration. Certain functions are restricted, such as the physical interface assignment, and determining which firewall zones are shared. Those functions must be performed by the Root Admin. VSYS Admin accounts may be stored locally or externally.
- **VSYS Read-Only Admin** The VSYS Read-Only account can only view the configuration of its own VSYS. It cannot modify its VSYS, nor can it run certain debugging commands. This account may be stored both locally and externally.

Local Admin Authentication

Administrator accounts are stored locally when they are created on the firewall. Storing an Admin account locally gives you the most flexibility with the different firewall features. Of course, storing an account locally has some drawbacks, but it is definitely the most common practice for storing accounts. Setting up local authentication for Admin accounts is easy since

most of the work is already done for you right out of the box. The main tasks that you will need to perform are creating the Admin accounts, as well as enabling the necessary services for the Administrator to connect to the firewall with.

Configuring Admin Users with Local Authentication

In this example, we will set up two administrator accounts which will authenticate locally to the firewall. One account will be set for read-write access, while the other account will be read only. Both of these accounts will be configured at the Root-Level of the firewall.

To set up the Admin users under the Juniper WebUI:

1. Select **Configuration | Admin | Administrators**, then choose **New**.
2. Specify the **Administrator Name**, which will be the username the administrator will log in as.
3. Specify the password in the **New Password** and **Confirm Password** fields.
4. Select either **Read-Write** for access to both view and modify the configuration, or **Read-Only** for access that will only allow the administrator to view the configuration.
5. If the administrator should be allowed to use SSH to log in to the firewall, then select the **SSH Password Authentication** checkbox.
6. Press **OK** to save the changes, or click **Cancel** to delete them without applying them.
7. Under **Configuration | Admin | Administrators**, make sure that the Admin Auth Server is set to **Local**.

For our example, we will enter the following:

Administrator Name:	**superadmin**
New Password:	**75a*Lforty**
Confirm Password:	**75a*Lforty**
Privileges:	**Read-Write**
SSH Password Authentication	**Checked**

We will create a second Administrator with the following:

Administrator Name:	**readonlyadmin**
New Password:	**foxTr0T1**
Confirm Password:	**foxTr0T1**
Privileges:	**Read-Only**
SSH Password Authentication	**UnChecked**

We will configure the Admin Auth Server using

Admin Auth Server Local

To set up Administrator accounts using the Juniper CLI:

```
set admin user "superadmin" password "75a*Lforty" privilege "all"
set admin user "readonlyadmin" password "foxTr0T1" privilege "read-only"
set admin ssh password disable username readonlyadmin
set admin auth server Local
save
```

NOTE

If you enable SSH admin access, make sure the interface the administrator will connect to has SSH enabled. If you are connecting with SSH V2 instead of V1, you must also enable SSH V2 on the device. For instance, if the administrator will connect to the Trust interface, the Trust interface must have SSH enabled, or else you will get a connection timeout without even being prompted to authenticate. The same will be true if you are trying to connect with SSH V2 and it is not enabled on the device.

External Authentication for Admin Accounts

Admin accounts may be authenticated by external authentication servers not part of the local system. In order to authenticate users externally, you must first set up the actual external server performing the authentication. Next, you must configure the firewall to authenticate the appropriate accounts to this external authentication server. When external authentication is employed, the firewall actually acts as an authenticating client, which presents the credentials to the server. We will begin our discussion of external authentication with a few of the fundamentals of authenticating Admin accounts externally. We will follow this discussion with an example so you can see the external authentication in action.

External Authentication Properties

- **Authentication** You may use RADIUS, LDAP, and SecurID to authenticate Admin users externally. Admin credentials are stored on the external server, and the firewall will query the server with the credentials the authenticating user presents in order to determine whether the user may log in. You may set the level of access an externally authenticated admin gets. This can either be read/write, read-only, or get privileges from the RADIUS server.

- **Obtaining Privileges from RADIUS** Administrator privileges may be queried from RADIUS server. This is not supported on LDAP or SecurID. You must upload the Juniper **Dictionary File** to the RADIUS server for this to work.
- **Read/Write Access** Root-Level Admins can be assigned read-write access regardless of what type of authentication server is used. VSYS-Level Admins may only be authenticated via RADIUS, and the **dictionary file** must be loaded into the server; otherwise, the VSYS Admins will not be able to log in.
- **Read Only Access** Root-Level Admins can be assigned read-only access regardless of what type of authentication server is used. VSYS-Level Admins may only be authenticated via RADIUS, and the **dictionary file** must be loaded into the server; otherwise, the VSYS Admins will not be able to log in.
- **Administrator Login Process** Administrators may log in via HTTP, HTTPS, Telnet, or SSH. The firewall will first check its local admin database to see if the credentials match an entry there. If there is no match for authentication in the local database, the firewall will query the external database if it is set to do so.

Configuring Admin Users with External Authentication

In this example, we will be setting up an external authentication server to authenticate Admin accounts that are stored on the external server. We will be covering external authentication servers in great detail in the following section. For this example, we will simply use a RADIUS server.

To configure this example with the Juniper WebUI:

1. Under **Configuration | Auth | Auth Servers**, select **New**.
2. Specify the **Name**, **IP/Domain Name**, and **Backup Servers** (optional).
3. You may specify the timeout values for the length of the user session with the **Timeout** and **Forced Timeout** values.
4. You may only select the **Admin** checkbox for the server to authenticate Admin users. Admin authentication servers cannot support any other type.
5. Make sure that **RADIUS** is selected, and specify a **Shared Secret**. For this example, we will leave the other RADIUS settings at their default values.
6. Set the firewall to use the RADIUS server for administrator authentication. This is accomplished by choosing **Configuration | Admin | Administrators** and under the **Admin Auth Server** drop-down menu, selecting the Local/<*External Auth Server Name*>.
7. Optionally, you can specify that the firewall should **Get privileges from the RADIUS server**. You could also select **External admin has read-only privileges**, or **External admin has read-write privileges**.

For this example, we will be using the following settings:

Server Type	**RADIUS**
Server Name	**RADIUS**
IP/Domain Name	**10.19.1.2**
Timeout	**10**
Forced Timeout	**0**
Account Type	**Admin**
RADIUS Shared Key	**R847noM**
RADIUS Port	**1645**
Retry Times	**3**
Retry Timeout	**3**

To set up the Admin server under the Juniper CLI:

```
set auth-server "Radius" id 2
set auth-server "Radius" server-name "10.19.1.2"
set auth-server "Radius" account-type admin
set auth-server "Radius" radius secret "R847noM"
set admin auth server "Radius"
set admin privilege get-external
save
```

Authentication Users

While Admin users define administrator accounts that can log in to the firewalls to manage them, there are also several types of user-level accounts. These accounts are typically distributed to users for operational, rather than administrative, use. We will cover each of these types in the following section.

Auth User Type

Auth users are accounts that are general-purpose users, which can be used on the firewall for policy authentication and WebAuth, to name a few. We will first discuss the properties of Auth users, and then cover an example in which we implement them into a firewall.

Auth User Type Properties

- **Auth Users** These are the actual user accounts that are configured on the firewall. The only properties for the Auth User account are username and password, and whether the account is enabled. As mentioned earlier, Auth users can be configured for policy-based user authentication and for WebAuth.
- **Auth Groups** A group becomes an Auth group when an Auth user is added to the group.

- **Local Authentication** Auth user accounts may be stored locally as well as Auth groups.
- **RADIUS Integration** You can use a RADIUS server to authenticate your Auth users so that their configuration is placed in a central location. This can also determine group membership.
- **LDAP Integration** LDAP authentication can be used to authenticate credentials passed by a user.
- **SecurID Integration** You can use SecurID to provide user authentication for Auth users.

Configuring Auth Users and Groups

In this example, we will configure a locally authenticated Auth user, and place it in a group.

To create Auth users and groups under the Juniper WebUI:

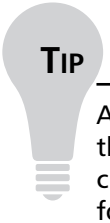
1. Select **Objects | Users | Local**, and then choose **New**.
2. Specify the **User Name**.
3. **Enable** the user account if this user should be able to log in. **Disable** will prevent the user from logging in, although the user will still be configured on the firewall.
4. At the very least, the **Authentication User** option must be checked for this user to be considered an Authentication User. A user may additionally be configured for other user types.
5. Enter the **User Password** and the **Confirm Password**, then press **OK**.
6. Create a group to reference multiple Auth users by. To create the group, select **Objects | Users | Local Groups**, and then choose **New**.
7. Specify a **Group Name**.
8. To add a user to the group, highlight the username you would like to add in the **Available Members** box by clicking it, and then click the **<<** button to add the user to the **Group Members**. To remove a user from the group, simply highlight the username you would like to remove, and then click the **>>** button.
9. Click **OK** to add the group.

We will configure the following information for our example:

User	**RMolly**
Status	**Enabled**
Auth User Type	**Checked**
Password	**Client9aM**
Group Name	**Local**
Members	**RMolly**

To create the Auth user and group under the Juniper CLI, do the following:

```
set user "RMolly" uid 1
set user "RMolly" type  auth
set user "RMolly" password "Client9aM"
set user "RMolly" "enable"
set user-group "Local" id 1
set user-group "Local" user "RMolly"
save
```



TIP

A user that is a member of a group may not be deleted without first removing the user from the group. You may alter certain attributes of the user, such as changing the password, without removing a user from the group. This is true for other group types such as IKE, XAuth, L2TP, and Admin.

The IKE User Type

Juniper has created a special user for VPN clients. This user has properties that are unique from other user types supported on the firewall. These properties are specifically used during client VPN negotiation.

IKE User Properties

- **IKE Users** These users are strictly used for authenticating VPN users. IKE user accounts may only be stored locally, but can be grouped together for easier management. These users authenticate to the firewall either by a username (IKE ID) or by X.509 certificates. With basic IKE users, there is no password authentication, just identity by IKE ID (username or certificate).
- **IKE Groups** A group becomes an IKE group when you add an IKE user to it. You can only store IKE groups locally.

Configuring IKE Users and Groups

In this example, we will create two different IKE users. First, we will configure an IKE user that will not use certificates to authenticate. This example will be followed by an IKE user that does use certificates to authenticate. Lastly, we will create a group and apply the users to that group. Figure 6.1 shows the creation of a simple ID IKE user through the Juniper WebUI.

To configure the IKE user and group through the Juniper WebUI:

1. Under **Objects | Users | Local**, click **New**.
2. Define the user **Name**.
3. Check the **IKE User** box.
4. Set the **Status** for **Enable**.
5. Select the **Simple Identity** option.
6. You may define how many concurrent connections may log in by defining **Number of Multiple Logins with the Same ID**. This is useful if you have given the same configuration file out to multiple users which would have the same login name.
7. Leave the **IKE ID** set to **Auto**, and define the **IKE Identity**. This is often set as an e-mail address, or something that will help distinguish the user. When a user configures the client VPN they must use the same ID name in their configuration. The simple identity is used when the user does not present a certificate to authenticate.
8. Click **OK**.

Next, we will create another IKE user that authenticates with certificates.

1. Under **Objects | Users | Local**, click **New**.
2. Define the **User Name**.
3. Check the **IKE User** box.
4. Set the **Status** for **Enable**.
5. Select the **Use Distinguished Name for ID** option.
6. You must at least define the **Email** address that is configured in the certificate.
7. Optionally, you may define the following attributes for the user: **CN**, **OU**, **Organization**, **Location**, **State**, **Country**, and **Container**.

NOTE

The firewall will verify the authenticity of the user by comparing the values for the attributes in the certificates to the ones you have defined. The certificate will also be checked to make sure it is signed by a trusted Certificate Authority.

8. To create an IKE group, go to **Objects | Users | Local Groups** and select **New**.

9. Now **Edit** the group to add the new IKE user. This user should appear in the **Available Members** box and can be added by highlighting the user and clicking **<<**.
10. Click **OK** to save the settings.

For our example, we will use the following settings (see Figure 6.1):

Username:	**JacksonL**
Status:	**Enabled**
IKE:	**Selected**
Number of Logins with the Same ID:	**1**
IKE Type:	**Simple Identity**
IKE Identity:	**jacksonl@mycompany.com**
Username:	**SmithyJ**
Status:	**Enabled**
IKE:	**Selected**
IKE Type:	**Use Distinguished Name for ID**
Email:	**smithyj@mycompany.com**
Group Name:	**IKE Group**
Members:	**SmithyJ, JacksonL**

Figure 6.1 Creating an IKE User through the Juniper WebUI

To set up IKE users via the Juniper CLI, perform the following steps:

```
set user "JacksonL" uid 4
```

```
set user "JacksonL" ike-id u-fqdn "jacksonl@mycompany.com" share-limit 1
set user "JacksonL" type  ike
set user "JacksonL" "enable"
set user "SmithyJ" uid 5
       set user "SmithyJ" ike-id asn1-dn wildcard
       "CN=,OU=,O=,L=,ST=,C=,Email=smithyj@mycompany.com,DC=," share-limit 1
set user "SmithyJ" type  ike
set user "SmithyJ" "enable"
set user-group "IKE Group" id 3
set user-group "IKE Group" user "JacksonL"
set user-group "IKE Group" user "SmithyJ"
save
```

XAuth User Type

XAuth is a standards-based protocol which extends IKE by providing VPN authentication, as well as IP, DNS, and WINS assignment. By using XAuth to add username and password authentication, you are further securing your network, since you have introduced another factor of authentication. XAuth also allows you to audit which users log in, rather than just permitting you to audit what IP addresses log in with traditional IKE users. We will begin this section with a discussion of how XAuth is supported within the Juniper firewalls, and then cover an example of how to implement XAuth users and groups into your firewall.

XAuth User Type Properties

The following parameters are all settings that are configurable for XAuth users. These properties include the types of authentication servers that can authenticate XAuth users, as well as the options they support.

- **XAuth User** This is the actual user account which supports XAuth authentication. Properties for this user include defining a username and password, IP address, DNS servers, and WINS servers. IP addresses may be assigned via an IP Pool shared by other XAuth users, or a static IP assigned to that specific user.
- **XAuth Groups** A group becomes an XAuth group when you add an XAuth user to it.
- **Local Authentication** Using local authentication provides user authentication, IP assignment, DNS and WINS assignment, and group membership.
- **RADIUS Integration** You can integrate RADIUS to authenticate your XAuth users. RADIUS can assign IP addresses, as well as DNS and WINS servers. You can also assign group membership in RADIUS.

- **LDAP Integration** You can integrate LDAP authentication to authenticate users, but it cannot assign IP addresses or name resolution servers.
- **SecurID Integration** You can integrate RSA SecurID to authenticate users, but this can only authenticate the user, it cannot assign IP addresses or name resolution servers.

NOTE

XAuth authentication takes place between Phase 1 and Phase 2 of the VPN tunnel establishment.

Configuring XAuth Users and Groups

We will configure two users in this example. The first user will use an IP Pool to specify what IP address will be assigned, while the second will just have a static address. Both of these users will be added to a group. See Figure 6.2 for an example of creating an XAuth user with an IP Pool in the WebUI.

To configure XAuth users and groups in the Juniper WebUI:

1. For the first user, we will be using an IP Pool, so we must create this first. Go to **Objects | IP Pool** and click **New**.
2. Give the IP Pool a **Name**.
3. Specify the **Starting** and **Ending** IP addresses of the IP Pool. Be careful not to use an IP address that is not valid, such as the network or broadcast address.
4. Click **OK** to apply the pool.
5. Select **Objects | Users | Local**, and then select **New**.
6. Fill out a **Username**, and then **Enable** the user.
7. Select the **XAuth** user type.
8. Specify the **New Password** and **Confirm Password**.
9. Select the **IP Pool** you have configured out of the drop-down list.
10. You have the option of filling in **DNS Servers** and **WINS Servers** for address resolution. These servers will be sent to the client's machine.
11. Click **OK** to add the user.

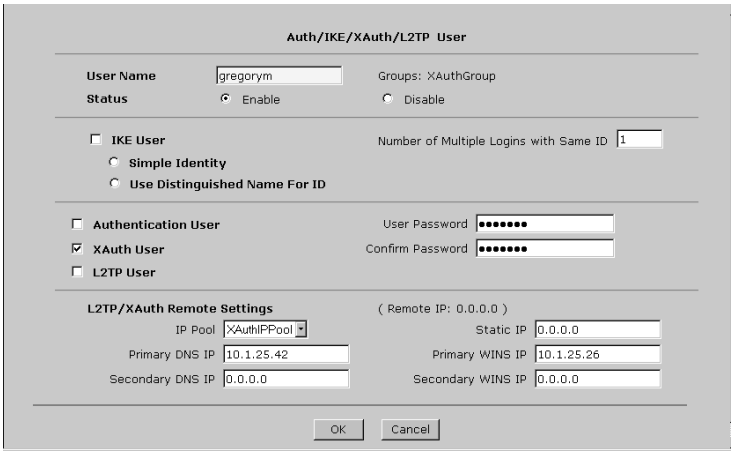
Now, we will create another user which will not use an IP Pool, but instead a static address.

1. Select **Objects | Users | Local**, then choose **New**.
2. Fill out a **Username**, and **Enable** the user.
3. Select the **XAuth** user type.
4. Specify the **New Password** and **Confirm Password**.
5. Fill in the Static IP address that the user will be assigned when they connect through XAuth.
6. Optionally, fill in the **DNS Servers** and **WINS Servers**.
7. Click **OK**.
8. Create a new group under **Objects | Users | Local Groups**, and select **New**.
9. Give the new group a **Group Name**.
10. Add the XAuth users that you would like as members of the group by selecting them in the **Available Members** box, and then clicking the **<<** arrow to add them to the group.
11. Click **OK**.

We will use the following configuration in our example:

Username	**gregorym**
Status	**Enabled**
XAuth User	**Checked**
Password	**75874jk??**
IP Pool	**XAuthIPPool**
Primary DNS Server	**10.1.25.42**
Primary WINS Server	**10.1.25.26**
Username	**janed**
XAuth User	**Checked**
Password	**Ufj*jfa0**
Static IP	**10.15.1.10**
Primary DNS Server	**10.1.25.42**
Primary WINS Server	**10.1.25.26**
IP Pool	**XAuthIPPool**
Starting Address	**10.15.1.1**
Ending IP	**10.15.1.150**
Group Name	**XAuthGroup**
Group Members	**gregorym, janed**

Figure 6.2 Creating an XAuth User with an IP Pool




```
To configure this example via the Juniper CLI:
set ippool "XAuthIPPool" 10.15.1.1 10.15.1.150
set user "gregorym" uid 8
set user "gregorym" type  xauth
set user "gregorym" remote ippool "XAuthIPPool"
set user "gregorym" remote dns1 "10.1.25.42"
set user "gregorym" remote wins1 "10.1.25.26'
set user "gregorym" password "iEdI0wh6NDGbZ5seXtCnYJaZ5vnjw50ptg=="
unset user "gregorym" type auth
set user "gregorym" "enable"
set user "janed" uid 9
set user "janed" type  xauth
set user "janed" remote ipaddr "10.15.1.200"
set user "janed" remote dns1 "10.1.25.42"
set user "janed" remote wins1 "10.1.25.26"
set user "janed" password "nbKZXUXUNfn+0msW6xCh+0rm+Dnx9nQ0hQ=="
unset user "janed" type auth
set user-group "XAuthGroup" id 4
set user-group "XAuthGroup" user "gregorym"
set user-group "XAuthGroup" user "janed"
save
```

WARNING

Use caution when defining IP Pools. As mentioned earlier, you do not want to use any IP address that is the network or broadcast address. You also do not want to use an IP address that may be assigned elsewhere in the network since this would cause an IP address conflict. Lastly, you need to make sure there are facilities in your network to properly route the traffic to the IP Pool. If you do not make sure this is properly handled, two-way communication will not be established. Packets will simply leave the firewall, and when the receiving host tries to respond, the traffic will either be misrouted, or dropped altogether.

Configuring Both IKE and XAuth for a Single User

To help enhance the security of VPN clients, you can combine XAuth and IKE within a single user account. This benefits you in two ways: First, you get the authentication of IKE which checks the identify of the user attempting to connect. Next, you get the XAuth username and password authentication for the individual user, as well as getting to assign an IP address, DNS servers, and WINS servers to the remote client. In this example, we will combine XAuth and IKE for a single user to accomplish the aforementioned security enhancements.

To configure XAuth and IKE via the Juniper WebUI:

1. Select **Objects | Users | Local** and then press the **New** button.
2. Define the **User Name** and set the status to enable.
3. Check the **IKE User** box, and define the **Number of Multiple Logins with Same ID**.
4. Select the **Simple Identity**, as well as **Auto** for the **IKE ID Type**.
5. Define the **IKE Identity**.
6. Check the **XAuth User** checkbox to make this account an XAuth user.
7. Set the **Password**, as well as the **Confirm Password** fields.
8. Select an **IP Pool** or define a **Static Address**.
9. Define the **DNS** and **WINS Servers**.
10. Click **OK**.

For our example, we will use the following:

User	**vpnuser**
Status	**Enabled**
IKE	**Checked**

Simple Identity	**Selected**
IKE Type	**Auto**
IKE Identity	**vpnuser@mycompany.com**
Number of Multiple Logins with Same ID	**5**
XAuth Type	**Checked**
Password	**48ajfUU<***
Confirm Password	**48ajfUU<***
IP Pool	**XAuthIPPool**
DNS Server	**10.1.25.42**
WINS Server	**10.1.25.26**

To configure this example through the Juniper CLI:

```
set user "vpnuser" uid 15
set user "vpnuser" ike-id u-fqdn "vpnuser@mycompany.com" share-limit 5
set user "vpnuser" type ike xauth
set user "vpnuser" remote ippool "XAuthIPPool"
set user "vpnuser" remote dns1 "10.1.25.42"
set user "vpnuser" remote wins1 "10.1.25.26"
set user "vpnuser" password "48ajfUU<*"
unset user "vpnuser" type auth
set user "vpnuser" "enable"
save
```

L2TP User Type

L2TP stands for Layer 2 Tunneling Protocol. It allows users to form a layer 2 tunnel from their client machine to the firewall. It provides IP addressing, DNS, and WINS information, and can also authenticate users based upon username and password. L2TP authentication takes place after Phase 2 of the VPN tunnel negotiation.

L2TP User Properties

The following parameters describe the various methods of authentication properties for the L2TP user. These options also describe the different settings that can be configured for L2TP users and the authentication servers which support them.

- **L2TP User** These are the actual users who support L2TP authentication. L2TP properties are similar to XAuth, and they include IP address assignment, as well as DNS, and WINS server assignment.
- **L2TP Groups** A group becomes an L2TP group when you add an L2TP user to it.
- **Local Authentication** You can authenticate L2TP users, assign addressing information, and apply group membership through local authentication.

- **RADIUS Integration** You can integrate RADIUS to authenticate your L2TP users and assign IP addresses, as well as DNS and WINS servers. You can also assign group membership in RADIUS.
- **LDAP Integration** You can integrate LDAP authentication to authenticate users, but it cannot assign IP address information or name resolution servers.
- **SecurID Integration** You can integrate RSA SecurID to authenticate users, but this can only authenticate the user, it cannot assign IP addresses or name resolution servers.

WARNING

L2TP is not an IPSec VPN. It does not natively support encryption, and is more like Point to Point Tunneling Protocol. If you require encryption, you should investigate whether you can use L2TP-over-IPSec, or IPSec with XAuth.

Configuring L2TP Users and Groups

In this example, we will configure an L2TP user using an IP Pool, and also one where the IP address is statically assigned. In addition, we will place the users in a group to create an L2TP group.

To configure L2TP users and groups in the Juniper WebUI:

1. For the first user, we will be using an IP Pool, so we must create this first. Go to **Objects | IP Pool** and click **New**.
2. Give the IP Pool a **Name**
3. Specify the **Starting** and **Ending** IP addresses of the IP Pool. Be careful not to use an IP address that isn't valid, such as the network or broadcast address.
4. Click **OK** to apply the pool.
5. Select **Objects | Users | Local**, then select **New**.
6. Fill out a **Username**, and **Enable** the user.
7. Select the **L2TP** user type.
8. Specify the **New Password** and **Confirm Password**.
9. Select the **IP Pool** you have configured out of the drop-down list.
10. You have the option of filling in **DNS Servers** and **WINS Servers** for address resolution. These servers will be sent to the client's machine.
11. Click **OK** to add the user.

Now, we will create another user that, instead of utilizing an IP Pool, will use a static address. To do so:

1. Select **Objects | Users | Local**, then select **New**.
2. Fill out a **Username**, and **Enable** the user.
3. Select the **L2TP** user type.
4. Specify the **New Password** and **Confirm Password**.
5. Fill in the Static IP address that the user will be assigned when they connect through L2TP.
6. Optionally, fill in the **DNS Servers** and **WINS Servers**.
7. Click **OK**.
8. Create a new group under **Objects | Users | Local Groups** and select **New**.
9. Give the new group a **Group Name**.
10. Add the L2TP users that you would like to be members of the group by selecting them in the **Available Members** box and clicking the **<<** arrow to add them to the group.
11. Click **OK**.

We will use the following configuration in our example

Username	**cooljam1**
Status	**Enabled**
XAuth User	**Checked**
Password	**234789aH**
IP Pool	**L2TPPool**
Primary DNS Server	**10.1.25.42**
Primary WINS Server	**10.1.25.26**
Username	**cooljam2**
XAuth User	**Checked**
Password	**74&*hflM**
Static IP	**10.16.1.160**
Primary DNS Server	**10.1.25.42**
Primary WINS Server	**10.1.25.26**
IP Pool	**L2TPPool**
Starting Address	**10.16.1.1**
Ending IP	**10.16.1.150**
Group Name	**L2TPGroup**
Group Members	**cooljam1,cooljam2**

To configure L2TP via the Juniper CLI:

```
set ippool "L2TPPool" 10.16.1.1 10.16.1.150
set user "cooljam1" uid 11
set user "cooljam1" type l2tp
set user "cooljam1" remote ippool "L2TPPool"
set user "cooljam1" remote dns1 "10.1.25.42"
set user "cooljam1" remote wins1 "10.1.25.26"
set user "cooljam1" password "NCN/JaIBNme6QpsSweCVkVvpV6nhnfiAEQ=="
unset user "cooljam1" type auth
set user "cooljam1" "enable"
set user "cooljam2" uid 12
set user "cooljam2" type l2tp
set user "cooljam2" remote ippool "L2TPPool"
set user "cooljam2" remote ipaddr "10.16.1.160"
set user "cooljam2" remote dns1 "10.1.25.42"
set user "cooljam2" remote wins1 "10.1.25.26"
set user "cooljam2" password "kzS0OAstNqr3SesKBZCQuqrPfyn4EqRJ3A=="
unset user "cooljam2" type auth
set user "cooljam2" "enable"
set user-group "L2TPGroup" user "cooljam1"
set user-group "L2TPGroup" user "cooljam2"
save
```

802.1x User Type

The 802.1x category is for users who are authenticated to the network with 802.1x. This user type is only capable of authenticating users to the network via hard-wired Ethernet or Wireless Ethernet. Users are configured on an external RADIUS server capable of supporting 802.1x. We will cover examples of configuring 802.1x users later in this chapter.

Internal Authentication Server

Juniper provides an internal authentication server known as the Local authentication server. The Local authentication server can authenticate every user type except 802.1x (which must be authenticated via RADIUS). It can also determine group membership, as well as assign specific attributes to users (such as IP addresses, DNS, and WINS where applicable). Local authentication searches the Local database where all user login credentials are contained in a proprietary format. The advantages of using local authentication include ease of configuration, support of every user type except 802.1x, and custom properties that cannot be used by some of the other authentication options. The main disadvantage of using Local authentication is the administrative burden it can cause. This is especially true if you have to manage several different devices with individual authentication mechanisms, with several users.

Local Authentication Support

The Local authentication server is built into the firmware of the firewall. It cannot be created or deleted. In fact, you can only modify the Idle timeout, and the maximum session length (Forced timeout). The following user types can be supported by the Local authentication server:

- **Auth Users** All features of Auth users are supported.
- **Admin Users** All features of Admin users are supported.
- **IKE Users** Local authentication is the only authentication mechanism that can authenticate IKE users.
- **XAuth Users** You can authenticate XAuth users, and provide assignments for IP address, DNS, and WINS servers with Local authentication.
- **L2TP Users** Local authentication can authenticate L2TP users, and provide assignments for IP address, DNS, and WINS servers.
- **User Group Assignment** You can map users to roles with the group membership capabilities of the Local authentication database.

Configuring the Local Authentication Server

Juniper has actually taken all of the hard guesswork out of configuring the local authentication server. As mentioned earlier, the local authentication server is already set up and ready to use. In this example, we will change the idle timeout for users, and implement a forced timeout that will end user sessions when they reach a maximum session length.
To configure the Local Authentication server via the Juniper WebUI:

1. Select **Configuration | Auth | Auth Servers** and next to the Local auth server click the **Edit** hyperlink.
2. To define the idle timeout, place a value in the **Timeout** field (or 0 to disable).
3. To define the maximum session length, place a value in the **Forced Timeout** field (or 0 to disable).
4. Click **OK**.

For our example, we will use the following:

Timeout	**60**
Forced Timeout	**120**

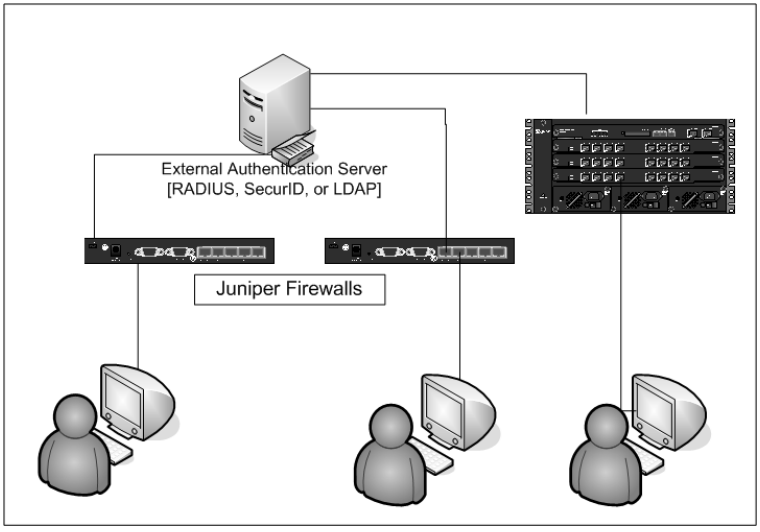
To configure the Local Auth servers with the Juniper CLI:

```
set auth-server "Local" timeout 60
set auth-server "Local" forced-timeout 120
save
```

External Authentication Servers

This section should come as a sigh of relief for any administrator who was feeling uneasy at the thought of managing users locally in an enterprise environment. Juniper provides integration support for three different authentication server types, which can make your life much easier. Each authentication server has different properties, and may support different authentication features. The main advantage of using external servers (for those of you who haven't used them before) is that you can centralize user management. In a small environment, this may not be such a big deal. But in a large environment with many users, and many devices, it would quickly become impractical to use local authentication on each device. By using external databases, users only have to keep one password, and if they change it in the external authentication server, it is changed for every device that queries the system and integrates with the authentication server. As shown in Figure 6.3, you can have multiple firewalls integrate to a single external authentication server so you don't have to configure the same users for each firewall. External servers may also provide a more detailed auditing and accounting facility, which can be a blessing for an administrator.

Figure 6.3 Integrating External Authentication to Firewalls



General Properties for External Servers

In this section, we will discuss some properties that are configurable across all authentication external authentication servers, regardless of the specific type. After we cover the general settings, we will dive into each specific authentication server and explore the details and settings on an individual basis. The general settings include the following:

- **Name** Authentication servers are essentially viewed as objects by the firewall. This means that you can configure an authentication server, and reference it in the configuration. You must give the authentication server a name.
- **IP/Domain Name** This is the IP address or resolvable hostname of the server to which you will authenticate. You can define additional servers as backups, but you must have at least one server which is defined in this field.
- **Backup Servers** To help provide redundancy for the authentication facility, the Juniper firewall provides additional backup authentication servers that you can query should the primary authentication server not respond.
- **Timeout** This refers to the idle timeout that occurs when the user becomes idle for the period of time you have configured in the authentication server. By default this is set to 10 minutes.
- **Forced Timeout** A Forced timeout essentially sets a maximum session length, which will end the user's session regardless of the network activity. By default, this value is set to 0, which means it's disabled.
- **Account Type** As mentioned before, you can configure an authentication server to authenticate different types of firewall users. You can configure the following account types on authentication servers you create: Auth, L2TP, Admin, XAuth, and 802.1x. IKE is not configurable since it may only be stored in the local database.
- **Stripping Separator** This is a unique character which determines where stripping occurs in an authentication request. You may also set the number of occurrences of this character to help prevent unintentional matching of a character.
- **Domain Name** This defines the domain name for the authentication server that is stripping the characters.
- **Fail-Over** This is a timeout value that determines how long the server should wait until it sends the authentication request to its backup server. The default is set to 0, meaning that failover is disabled.
- **Source Interface** This defines the interface that authentication requests should be sent out on. It can be especially useful for tunneling traffic through a VPN tunnel interface for a secure authentication channel.

NOTE

You cannot configure an authentication server to authenticate the Admin account type with any other account type in a single user-defined authentication server. Admin accounts must be configured in their own authentication server.

RADIUS Server

RADIUS has been a widely implemented authentication standard for quite some time. Many different systems and applications use RADIUS for central authentication. This fact makes RADIUS an attractive option for authenticating user accounts. RADIUS allows you to store users, groups, and highly extensible for custom attributes (defined in dictionary files). In the authentication process, the Juniper firewall acts as a RADIUS client, which proxies the user authentication requests, and submits them to the RADIUS server which determines whether the user's credentials are valid. The response is then returned to the firewall, which will either accept the user, or deny the user based upon the response from the server. A couple of configurable options exist for the RADIUS server, which are listed next.

RADIUS Server Properties

The following properties are unique to RADIUS servers configured in the Juniper firewall. You will have to ensure that the settings you choose are supported on your RADIUS server. Check your RADIUS manual for full details regarding your specific server.

- **RADIUS Port** This allows you to define the UDP port, which the firewall will connect to the RADIUS server on. The default is 1645, but some legacy RADIUS servers listen on 1812, or you may have chosen your own port for the server to listen on.
- **Shared Secret** RADIUS employs a shared password to provide a secure encrypted channel between the client (firewall) and the RADIUS server. The shared secret must be identical to the one you configure on the RADIUS server, otherwise authentication will fail.
- **Retry Times** This is the number of times you should attempt to send an authentication request to the RADIUS server before the server is considered unreachable. The default is three tries.
- **Retry Timeout** This is the timeout value for the amount of time that should elapse before the firewall will send another authentication request to the RADIUS server. The default is three seconds.
- **Acct-Session-ID Length** This field defines the length value of the Acct-Session-ID for accounting purposes.
- **Send Calling Station ID** Transmits the Calling Station ID to the RADIUS server if it is enabled.
- **RFC Compatibility** If this option is enabled, the firewall with be compatible with the legacy RADIUS standard RFC 2138. If it's not enabled, it will be compatible with the current standard.

- **Zone Verification** This option provides an extension to the RADIUS authentication mechanism by transmitting a custom attribute that defines what zone the user is authenticating from. This zone is compared to the firewall zone listed on the RADIUS server.
- **RADIUS Accounting** You can define what port to send accounting information to the RADIUS server. By default, this value is 1646. Legacy RADIUS servers use UDP port 1813.

WARNING

If you use multiple RADIUS servers, you must make sure they use the same *Shared Key* since you can only define one Shared Key for the primary and backup servers in the Juniper firewall. If you fail to set this up properly, authentication will fail to the backup server.

RADIUS Authentication Capabilities

The following account types are supported with RADIUS authentication. Depending on what account type you are supporting, you may need to load the Juniper firewall Dictionary file into your RADIUS server.

- **Auth Users** RADIUS can support authentication for Auth users.
- **Admin Users** RADIUS can authenticate Admin users. There are specific capabilities based upon what type of user is authenticating, and whether the dictionary file was loaded into the RADIUS server. This was covered earlier in the chapter.
- **XAuth Users** RADIUS can authenticate XAuth users, and provide assignments for IP address, DNS, and WINS servers.
- **L2TP Users** RADIUS can authenticate L2TP users, and provide assignments for IP address, DNS, and WINS servers.
- **802.1x Users** RADIUS is exclusively used to authenticate 802.1x users.
- **User Group Assignment** RADIUS supports group membership authorization of users. This is useful for mapping user roles based upon group membership.

Configuring RADIUS Servers

In this example, we will create two RADIUS servers. We will configure the first RADIUS server to authenticate Administrators, and the second to authenticate Auth, L2TP, and XAuth

users. For the second server, we will also define a backup server, set timeouts, and the source-interface to initiate the connection on.

To configure this example with the Juniper WebUI:

1. Select **Configure | Auth | Auth-Server** and click **New**.
2. Define a **Name** to reference and identify the server by.
3. Define the **IP/Domain Name** which the firewall will use to contact the server.
4. Select **Admin** as the account type.
5. Make sure that the **RADIUS** server is selected.
6. Define the **Shared Secret** password that matches the password on RADIUS server.
7. Click **OK**.
8. Create another RADIUS server by going to **Configure | Auth | Auth-Server** and clicking **New**.
9. Define a new **Name** for the authentication server.
10. Define the **IP/Domain Name** for the second authentication server.
11. Define an additional backup server in case the primary server is not reachable. This is set under the **Backup1** field.
12. Change the default idle timeout from 10 minutes to 60 minutes under the **Timeout** field.
13. Define a maximum session length of 120 minutes under the **Forced Timeout** field.
14. To set the source interface, which the traffic should initiate from, change the **Source Interface** to the appropriate interface in the drop-down menu.
15. Make sure the **RADIUS** server is selected, and set the **Shared Key**.
16. Click **OK**.
17. The **RADIUS Accounting port** may be set under **Configuration | Auth | Firewall**. At the bottom of the screen there is a field for the port. You may also set the firewall to clean up accounting sessions by checking the **Accounting Listener Action** to **Cleanup Sessions**.

For our examples we will use the following configuration.

Server Name	**Radius3**
IP/Domain Name	**10.19.1.3**
Account Type	**Admin**
Server Type	**RADIUS**
Shared Key	**84aJm0M**

Server Name	**Radius2**
IP/Domain Name	**10.15.1.10**
Backup1	**10.15.2.10**
Timeout	**60**
Forced Timeout	**120**
Account Type	**Auth, XAuth, L2TP**
Source Interface	**Ethernet1**
Server Type	**RADIUS**
Shared Key	**84a><0M**
RADIUS Account Port	**1646**
Cleanup Sessions	**Checked**

To configure the server with the Juniper CLI:

```
set auth-server "Radius3" id 3
set auth-server "Radius3" server-name "10.19.1.3"
set auth-server "Radius3" account-type admin
set auth-server "Radius3" radius secret
    "4tek2B/JNJweNTsDB3CissEAGnnurdoJTQ=="
set auth-server "Radius2" id 1
set auth-server "Radius2" server-name "10.15.1.10"
set auth-server "Radius2" backup1 "10.15.2.10"
set auth-server "Radius2" account-type auth l2tp xauth
set auth-server "Radius2" timeout 60
set auth-server "Radius2" forced-timeout 120
set auth-server "Radius2" radius secret
    "4tek2B/JNy579zs9tVCRZ6/kwPnZTSpzVg=="
set auth-server "Radius2" src-interface "ethernet1"
set auth radius accounting action cleanup-session
set auth radius accounting port 1646
save
```

Notes from the Underground…

Cracking Weak Passwords

Traditional user authentication has security limitations of its own. Administrators are all too familiar with user habits for password creation and storage. It seems by nature that most users choose weak passwords which are easy to remember. While this benefits the user, it also makes a hacker's job much easier. Freely distributed programs such as Hydra and John the Ripper have existed for years

Continued

which can guess passwords at blazing speeds. Dictionary lists are loaded into these hacking programs which can contain entire languages and other word enumerations. The trouble is the hacker only has to be right once, while you, the network defender, must be right every time. The next authentication solution gives you, the administrator, a new weapon that dramatically reduces the effectiveness of these automated cracking programs.

SecurID Server

SecurID is a commercial authentication suite offered by RSA. At the very least, SecurID consists of an authentication server, client authentication software, and hardware tokens. SecurID combats the use of weak passwords by adding an additional factor to the authentication process. In traditional authentication, usernames are authenticated by one factor, which is a single password that the user chooses. Administrators may be able to force complex passwords, but they still wouldn't be able to make the user change their passwords every 60 seconds. That's right,. With SecurID, the user's password actually changes every 60 seconds. This feat is accomplished by the use of a hardware token whose number changes every minute. Each number is chosen by a proprietary algorithm that uses the hardware serial number, and the unique seed file sent with each batch of tokens to determine what the number is at any given moment. In order for the SecurID server to be able to authenticate, the user must be given the token's serial number, which is mapped to a particular user and the seed number for the batch. Lastly, the token is synced up with SecurID server, and you now have two-factor authentication. When the user logs in, they present their username and password, which is appended with the number on the keyfab. The strength behind this approach is that even if an attacker knew a user's password, they would need to know the number keyfab at the exact moment they tried to connect. Alternatively, if the attacker had somehow gotten access to the keyfab, he would still need to know the user's password. This combination of something you have, along with something you know, is effective in defeating password cracking programs.

SecurID Server Properties

The following section describes the individual settings you can configure for your SecurID server.

- **Client Retries** This is the number of times the firewall (SecurID Client) will try to connect to the SecurID server. The default is three.
- **Client Timeout** This is the timeout value used to determine how long the firewall should wait until it considers the SecurID server unreachable. The default is five seconds.
- **Authentication Port** This is the port which the firewall tries to connect to the SecurID server. The standard is TCP/UDP port 5500.

- **Encryption Type** This is the encryption algorithm used to create a secure channel between the firewall (SecurID client) and the SecurID server. Your choices are DES of SDI.
- **Use Duress** This option lets you only allow the user to log in once with their token. Duress must be supported on the SecurID server for this option to work. After the first login, an administrator will have to reset the status of the user's account for them to log in again.

SecurID Authentication Capacities

SecurID is capable of providing support for the following types of users. Pay special attention to what user account type you are authenticating with SecurID since it may have restricted functionality.

- **Auth Users** You can use SecurID to authenticate Auth users to your firewall.
- **Admin Users** SecurID can authenticate Admin users, but if you tell it to get privileges from the authentication server, it will only give the administrator read-only privileges.
- **XAuth Users** SecurID can only authenticate XAuth users, and cannot assign IP addresses, DNS, or WINS information.
- **L2TP Users** Just like XAuth, SecurID can only authenticate L2TP users, and cannot assign IP addresses, DNS, or WINS information.

Configuring a SecurID Server

In this example, we will configure a SecurID server to authenticate Auth and XAuth users with two-factor authentication.

To configure SecurID with the Juniper WebUI:

1. Select **Configuration | Auth | Auth-Servers**, and then select **New**.
2. Give the server a **Name** to be referenced by.
3. Specify the **IP/Domain Name**.
4. If necessary, you can define additional backup servers in the **Backup** fields.
5. Specify what **Account Types** you would like this server to authenticate for.
6. If you would like to specify a source interface, you may do it in the **Source Interface** drop-down menu.
7. Make sure the **SecurID** server is selected.
8. If necessary, change the default port and connection values under **Authentication Port**, **Client Timeout**, and **Client Retry**.

9. Make sure the **Encryption Type** you define on the firewall matches the one you configure the **SecurID** server to use. Your options are DES or SDI.
10. If you would like to restrict clients to only one login, you may use the **Use Duress** option. This must also be supported on the SecurID server.

We will use the following options in our example:

Server Name	**SecurID**
IP/Domain Name	**172.16.2.20**
Account Types	**Auth, XAuth**
Server Type	**SecurID**
Encryption Type	**DES**

```
   To set the SecurID Server via the Juniper CLI:
set auth-server "SecurID" id 2
set auth-server "SecurID" server-name "172.16.2.20"
set auth-server "SecurID" account-type auth xauth
set auth-server "SecurID" type secured
save
```

LDAP Server

Lightweight Directory Access Protocol (LDAP) is a standard that was developed by the University of Michigan in 1996. It provides a hierarchical organization for objects. LDAP is widely implemented in many different types of applications. It is probably best known for its integration with Microsoft's Active Directory. The nice thing about using LDAP is that you probably already have an LDAP authentication server in your environment. The firewall acts as an LDAP client which queries an LDAP server in your environment (such as a Domain Controller).

LDAP Server Properties

The following properties are unique to the LDAP server. It would be out of the scope of this book for a full discussion of LDAP, so if you are new to LDAP you should gather a bit more information if you are unsure of these settings. The best place to start is with the manuals for your LDAP server itself, such as Active Directory, eDirectory, and OpenLDAP.

- **LDAP Port** You must specify what port the firewall will connect to on the LDAP server. By Default, LDAP uses UDP port 389.
- **Common Name Identifier** This is the property of the LDAP directory that is used to identify an individual user. By default, it is set to cn.
- **Distinguished Name (dn)** This is the LDAP path that the firewall should search when it tries to authenticate to the firewall, and is dependent on the struc-

ture of your LDAP directory. (We will take a look at some examples of the LDAP structure next.)

LDAP Authentication Capacities

LDAP can support the following user account types. Be sure to note some of the limitations of LDAP authentication before making a decision to use it.

- **Auth Users** You can use LDAP to authenticate Auth Users to your firewall.
- **Admin Users** LDAP can authenticate Admin users, but if you tell it to get privileges from the authentication server, it will only give the administrator read-only privileges.
- **XAuth Users** LDAP can only authenticate XAuth users, but it cannot assign IP addresses, DNS, or WINS information.
- **L2TP Users** Just like XAuth, LDAP can only authenticate L2TP users, and cannot assign IP addresses, DNS, or WINS information.

Configuring an LDAP Server

In this example, we will create an LDAP server which will authenticate L2TP users. We will define an additional LDAP server to query in case our server becomes unavailable. We will also configure the LDAP server to send all traffic through a VPN tunnel interface (see Chapter 11 for more information on configuring tunnel interfaces).

To configure this example through the Juniper WebUI:

1. Select **Configuration | Auth | Auth-Servers**, and choose **New**.
2. Define a **Name** for the LDAP server.
3. Specify the **IP/Domain Name** for the server.
4. Optionally specify an additional backup server.
5. Define the **Port** the firewall should send authentication requests to. The default is 389.
6. Specify the **Common Name Identifier**. This will depend on how your LDAP server is set up. The default is cn.
7. Specify the **Distinguished Name** (dn). Again, this will depend on the structure of your LDAP.

For our example, we will use the following.

Server Name	**LDAPServer**
IP/Domain Name	**192.168.2.2**
Backup1	**192.168.2.3**

Account Type	**L2TP**
Source Interface	**Tunnel.1**
Server Type	**LDAP**
Port	**389**
Common Name Identifier	**cn**
Distinguished Name	**DC=testserver DC=org**

```
To configure this example in the CLI:
set auth-server "LDAPServer" id 1
set auth-server "LDAPServer" server-name "192.168.2.2"
set auth-server "LDAPServer" backup1 "192.168.2.3"
set auth-server "LDAPServer" account-type l2tp
set auth-server "LDAPServer" type ldap
set auth-server "LDAPServer" ldap cn "cn"
set auth-server "LDAPServer" ldap dn "DC=testserver, DC=org"
set auth-server "LDAPServer" src-interface "tunnel.1"
save
```

Tools & Traps…

The Hidden Dangers of Authentication

A very common mistake that administrators make is not to secure the traffic between an authentication client and the server. Some types of authentication have encryption built into the protocol, while others such as LDAP do not. But you shouldn't lose your vigilance even if the protocol doesn't transmit the credentials in plain-text. Tools such as Cain & Abel not only sniff the traffic and capture authentication requests, but also save them for offline cracking. This presents a very dangerous problem since credentials may be captured in a passive attack, and the administrator may not know it until it's too late. Since authentication is such a sensitive operation that takes place over the network, you should ensure this traffic is properly secured. There are a couple of nifty tricks administrators employ to help lock down this traffic. Network segmentation and the isolation of sensitive traffic is a good start. Perhaps you can configure a secure VLAN that is isolated from other areas of the network. This will not provide encryption, but will make it more difficult to be able to get access to that traffic on the wire. The next step would be to configure a secure path between the authentication client and server, such as a VPN. Some authentication protocols even support tunneling the traffic in SSL. On the Juniper firewall, you can

Continued

configure a tunnel interface and set the authentication server to use that as a *source interface*. If your authentication server supports IPSec, which all newer versions of Windows, Linux, and UNIX do, you could have the traffic sent through the tunnel securely from the source to the destination.

Infranet Authentication

Infranet Authentication is a new addition to Juniper's security suite of products. This product seeks to fill gaps in traditional network security by forcing users to authenticate before they can access certain resources defined by the administrator. The Unified Access Control (UAC) suite consists of two products: the Infranet Controller (IC), and Infranet Enforcer (IE). The IC is its own appliance and handles authentication, client distribution, and all related access authorization. The IE is actually any Juniper firewall running firmware 5.3 or later. The strength of the Infranet authentication is that it can authenticate users on more than just their username and password. Infranet authentication can also evaluate the properties of a user's computer such as the applications installed and/or running, antivirus, the operating system, patch level, firewall, Registry keys, and files on the system. While this book is not meant to replace the configuration guide for the IC, certain features of it are pertinent to this chapter and will be covered here.

UAC Product Overview

In order to get the IC and IE to talk to each other, you must first configure both the IC and IE to establish a trust relationship. This relationship is established via SSL certificates. Essentially, the IC must use a certificate signed by a public key certificate, which is imported onto the IE. Assuming this is done correctly, the two devices will then be able to communicate in a secure manner over SSH.

The IC will either authenticate users with a software client known as the Infranet Agent, which is installed on the user's computer, or do so on a policy-by-policy basis. The latter will be covered in the section "Policy Based Infranet Authentication" in this chapter.

The Infranet Agent is installed by browsing to the IC. This only needs to happen once, and is initiated by the user browsing to the IP address or fully-qualified domain name of the Infranet Controller (for example, https://<InfranetController>). Once the agent is installed, it will automatically bring up an authentication window when it can connect the IC. Among other things, the user will be forced to authenticate if he or she would like access to the protected resources behind the firewall. The agent may also be configured to scan the user's computer for running processes, files, Registry keys, antivirus, the operating system in use, the firewall, and patch level as mentioned earlier. Specific actions can be taken based upon the outcome of the scan.

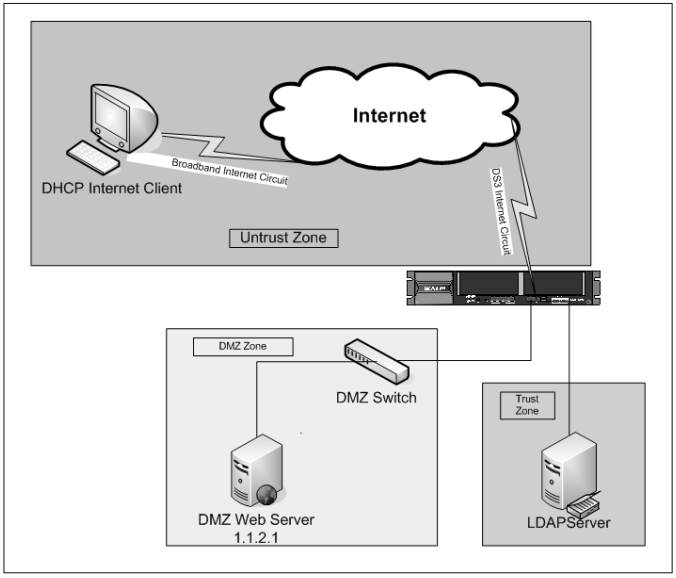
One of the key features of the UAC is that it can build IPSec VPN tunnels from the client with the Infranet Agent to the IE. The advantage of this is that it dynamically secures traffic between the user and the firewall. This is certainly a nice security enhancement, but it

may also be required by regulations or standards such as the Payment Card Industry Data Security Standard (PCI).

When the Infranet Agent is used to establish dynamic IPSec tunnels, the IC actually communicates with the IE via SSH and directs it to configure a policy-based VPN. In fact, it must create an IKE Gateway, and AutoKey IKE just like a traditional VPN. The IC then directs the Agent to set up the IPSec VPN with the IE with the appropriate settings.

Figure 6.4 outlines a typical UAC configuration. In this figure, client machines are in an unsecured zone (such as a Guest network). Before we allow them to talk to our servers in the Clean Access Zone, we first want to validate their identity, and perform some host-checking tasks to ensure their machines are not compromised. The IC will handle that part, and once it is satisfied that the client machine is not harmful, it will communicate this to the firewall, which will create a policy to allow the traffic to pass through the host into the Clean Access Zone.

Figure 6.4 Infranet Client Authentication Scenario



IE Properties

This section contains the configurable settings on the IE that are used to specify how the IE should connect to the IC. These settings will match similar fields on your IC configuration.

- **Infranet Controller Instance** This field is used to name the IC. This should be something that has recognizable significance. The IC instance is just like other external authentication servers in the sense that it can be referenced as an object in appropriate places in the configuration.

- **IP/Domain Name** This field is for you to define either the IP address of the IC, or the fully qualified domain name.
- **Port** This is the TCP port that the IE will communicate to the IC. The default is 11122.
- **Timeout** This is the timeout value in seconds that's used to determine if the IC is unreachable. The default is 60 seconds.
- **Redirect URL** This is the URL that all users will be directed to if you configured this option.
- **Source Interface** This is the interface which the IE should use to connect to the Infranet Controller on.
- **Password** This is the shared password key you will enter on both IC and IE to establish a secure connection.
- **Selected CA** As mentioned in the explanation of how the firewall makes a secure connection with the IC, you must have the public key which signed the certificate the IC is using to establish trust. In this drop-down menu, you will select the CA on the firewall that signed the IC certificate.
- **Full Subject Name of IC Cert** This is the subject name contained in the IC's certificate.
- **Contact Interval** This is found under the general settings, and is used to specify how often the IE should contact the IE.
- **Timeout Action** This specifies what should happen to the connections that are still open if the firewall should lose its connection to the IE. Your options are Close (which will close all connections opened by the IC), Open (which allows all sessions to remain open and permits new ones to form), and No Change (which allows existing sessions to continue). No matter what you choose, all new sessions will require authentication.
- **Test Only** This option puts the device in a test mode where all connections will be allowed, but the Redirection the IE would take if it was in a production mode will be logged.
- **Regular Mode** This is the production mode for the IE, which enforces all policies as specified on the IC.

Configuring the IE for Infranet Authentication

The purpose of this configuration example is to give you a good working knowledge of setting up an IE in your environment. Configuration of the IC is out of the scope of this book, but the concepts covered here will go a long way in helping you maximize your understanding of the technology.

To set up the UAC with the Juniper WebUI:

1. Import the certificate that signed the Infranet Controller certificate. The public key of the CA usually signed the Infranet Controller's certificate. To perform this step, go to **Objects | Certificates** and click the **Browse** button. Select the CA certificate and click **OK** to import it into the firewall.
2. We will create the Infranet Controller object. Select **Configuration | Infranet Auth | Controllers** and select **New**.
3. Define the name from the Infranet Controller in the **Infranet Controller Instance** field.
4. Define the **IP/Domain Name**.
5. Specify the **Port** which the Infranet Enforcer should connect to the Infranet Controller on. The default is **11122**.
6. The **Timeout** value can be implemented to determine how long the Infranet Enforcer should wait until the Infranet Controller is deemed unreachable.
7. You can configure a **Redirect URL**, which is used to redirect HTTP traffic, if that option is specified in the configuration.
8. Just as with other authentication servers, you can specify the **Source Interface** the Infranet Enforcer should initiate connections to the Infranet Controller on.
9. The NACN password that is defined in the Infranet Controller must be specified in the Infranet Enforcer under the **Password** field.
10. In order for the trusted communication to be established, you must select the appropriate **Selected CA** certificate from the drop-down menu.
11. Optionally, you can specify the **Full Subject Name of IC Cert** for further verification.
12. Click **OK**.

Next, we can modify a couple of systemwide settings:

1. To configure these settings, go to **Configuration | Infranet Auth | General Settings**.
2. Specify the **Contact Interval**, which will default to 10 seconds if nothing is set.
3. Specify whether the Infranet Enforcer should use **Close**, **Open**, or **No Change** for new and existing sessions when contact is lost between the Infranet Controller and Enforcer.
4. Specify whether the Infranet Enforcer is in **Regular** or **Test Only** mode.
5. Click **OK**.

For our example, we will use the following:

Server Name	**Infranet Controller**
IP/Domain Name	**10.1.1.50**
Port	**11122**
Timeout	**60**
Redirect URL	**http://internalredirect.local**
Source Interface	**Ethernet1**
Password	**98243JAIM**
Select CA	

Email=Tester@Test.com,CN=tester.test.com, OU=test

```
    To configure the Infranet Controller via Juniper CLI:
set infranet controller name "Infranet Controller"
set infranet controller name "Infranet Controller" host-name 10.1.1.50 port
    11122
set infranet controller name "Infranet Controller" src-interface ethernet1
set infranet controller name "Infranet Controller" password
    "SdSaBMMzN4W6BvsF5/CmNCnmkCnZYRk9HA=="
set infranet controller name "Infranet Controller" ca-hash
    "8BA3AAE570FFFA74C31113C695F4EB92E8459607"
set infranet controller name "Infranet Controller" url
    https://internalredirect.local
save
```

Policy-Based User Authentication

If you thought that basic device and VPN authentication was all you could do with Juniper's authentication facilities, you will be pleasantly surprised with the additional capabilities discussed in this section. Building on the foundations of user authentication that we covered in the previous sections, we are going to discuss how you can enforce firewall policies with user authentication.

Explanation of Policy-Based Authentication

Firewall policies are the primary security mechanism of the firewall. Juniper provides you with very granular control over your firewall behavior by allowing you to configure many of its features on a rule-by-rule basis. This is also true for user authentication. Configuring a rule to incorporate user authentication follows the same fundamentals you would follow to create a normal policy. First, you would specify the zone direction that the traffic will match. Next, you will specify the source and destination addresses that the traffic will match on, as well as the service and the action. You would probably want to specify the action as either permit or tunnel, instead of deny, since authenticating users just to drop their traffic wouldn't be very useful. Finally, you would configure the additional options you would like the firewall to perform. These options include logging, NATing, traffic shaping, and user authentica-

tion. Juniper offers three different types of authentication checks for you to configure in a policy. You have the choice on configuring one of the following on each policy: User Auth, WebAuth, and Infranet Auth. In order for the user's traffic to be allowed by the policy, they must first authenticate to the firewall. Once authentication has occurred, the users will be able to send traffic as long as their session is open. We will cover the details of each of the different authentication types in the following section.

Authenticating with User Auth

User Auth has the following properties, which can be configured on a policy-by-policy basis:

- When a user tries to pass traffic through the firewall, the request will be proxied while the user authenticates. If you set a policy to authenticate with User Auth, the user must authenticate with HTTP, FTP, or Telnet to get access. If the user is attempting to send an HTTP request, the request will get proxied and the user will be prompted for authentication credentials.
- When you use the User Auth facility, you must specify how the firewall will authenticate the user. First, you must specify what Authentication Server will be used. Next, you must specify if you will authenticate the user based upon a User Group, Group Expression, or User.

Configuring Policies with User Auth

In this example, we will configure two different policies, which will use local and external authentication servers, to aid our efforts to secure users. Figure 6.5 outlines the scenario for the second example, where you have an external user whose IP address will change. We want to provide some extra security since we cannot identify the user's source IP address in a practical manner. We will assume that you have configured an external authentication server, as described in the previous section.

To configure policy-based user authentication via the Juniper WebUI:

1. Select **Policies** to bring you to the **Policy Rulebase**.
2. From the drop-down menus, define the **From Zone** and the **To Zone** and select **New**.
3. Define the matching terms which the firewall will use to match the traffic. This includes the **Source Address**, **Destination Address**, and **Service**. You may optionally define **Anti-virus**, **Deep Inspection**, **VPN Tunnels**, **Logging**, and **Position**.
4. Click **Advanced** to enter the extended options for the policy.
5. Amongst the other options you may select (**NAT**, **Traffic Shaping**, **Counting**, and **Scheduling**), you must select the **Authentication** option.

6. Select the appropriate **Auth Server** from the drop-down menu in the **Authentication** frame.
7. Choose whether you will match on a particular **User Group**, **User Expression**, or **User** from the drop-down menus.
8. Click **OK**.

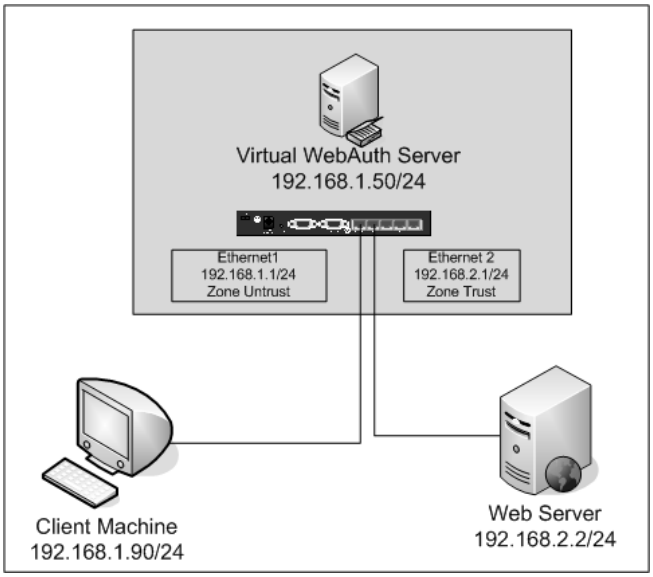
For our example, we will configure the following:

Zone	**From Untrust to Trust**
Policy Name	**Authenticator**
Source Address	**ANY**
Destination Address	**10.55.1.10/32**
Service	**FTP**
Logging	**Enabled**
Auth Server	**Local**
User	**Local – Rmolly**
Zone	**From Untrust to DMZ**
Policy Name	**External Auth**
Source Address	**ANY**
Destination Address	**1.1.2.1/32**
Service	**HTTP**
Logging	**Enabled**
Auth Server	**LDAPServer**
User Group	**Allow Any**

To set up this example through the Juniper CLI:

```
set policy id 2 name "Authenticator" from "Untrust" to "Trust"  "Any"
            "10.55.1.10/32" "FTP" permit auth user "RMolly" log
set policy id 3 name "External Auth" from "Untrust" to "DMZ"  "Any"
     "1.1.2.1/32" "HTTP" permit auth server "LDAPServer" log
save
```

Figure 6.5 Securing Resources with Policy Authentication



TIP

Securing your network can be a hard task. You have to balance user functionality with security in a graceful fashion. A common mistake that administrators make is to open up access from any source to a network or system that should have restricted authorized access. This is typically done because the administrator has no way of establishing a VPN, or locking down the source IP addresses of the connecting hosts. To help mitigate the risk of opening up access to any source, Juniper allows you to add user authentication to a policy. This way, you can leave the source addresses as any (since you may not be able to define all of them) and have the users interactively prompted for credentials to the network. If they fail, their traffic will not be permitted by the policy which is authenticating them; if they pass, they will be permitted.

Authenticating with WebAuth

WebAuth is similar to User Auth, and because of this they are often confused with one another. The differences between the two are very subtle. But after reading this section, you'll be able to show up all of your Juniper firewall buddies with your superior knowledge of WebAuth.

WebAuth Settings and Details

WebAuth has the following properties:

- Just like User Auth, WebAuth is configured per policy, and the user will be prompted to authenticate to the firewall. WebAuth can also be configured to authenticate locally, or to an external authentication source. The key difference between WebAuth and User Auth is that WebAuth is configured on a separate IP. You can configure which interface to use.
- You configure what IP address the WebAuth server should listen to on the interface which the user should be authenticating from. For instance, a user authenticating from the Internet might have a WebAuth server on the Untrust interface.
- The IP address you configure as the WebAuth server must be in the same subnet as the interface which you configured WebAuth on.
- You may specify that the WebAuth should only accept connections over SSL by setting the SSL option. A user would then be forced to authenticate on the SSL port 443 or https://<WebAuthIPAddress>.
- To authenticate to a policy via WebAuth, users must first browse to the WebAuth address, where they will be prompted for credentials. After the user is successfully authenticated, they will be able to pass traffic through the desired policy.
- When you use the WebAuth facility, you must specify how the firewall will authenticate the user. First, you must specify what *Authentication Server* will be used. You specify what server to use on the WebAuth configuration page.
- After a WebAuth Server is specified, you must specify if you will authenticate the user based upon a *User Group*, *Group Expression*, or *User*.

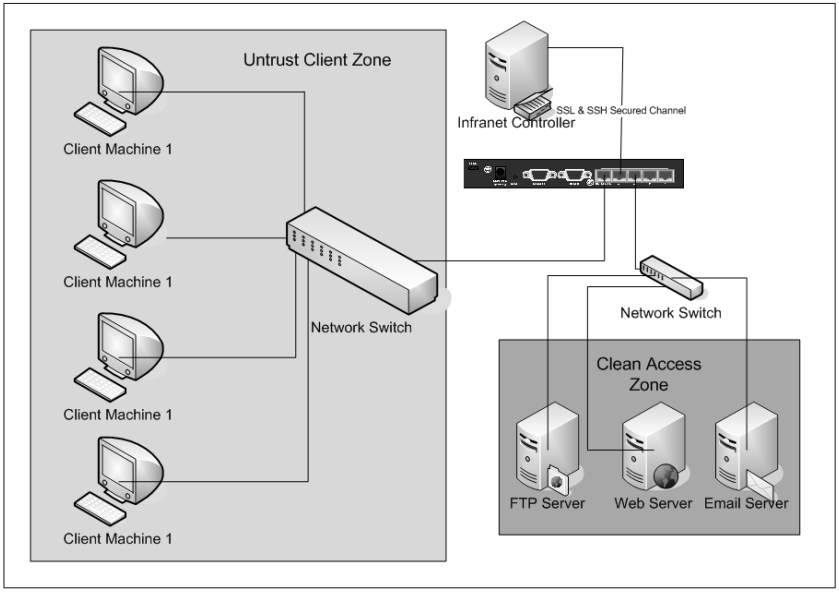
WARNING

While using User Auth and WebAuth can greatly increase the security of your network, please be aware that the firewall authenticates a specific source IP address for WebAuth and User Auth. This is of particular importance if devices behind the firewall are NAT'd since it would only take one user to authenticate, then the rest could also pass traffic. To help avoid this security issue, consider the proper placement of the firewall to help mitigate this risk.

Configuring WebAuth

In this example, we will configure a policy that uses WebAuth to authenticate users. Figure 6.6 shows how this example is set up. Remember that the WebAuth server is not a physical machine—it exists, virtually, in the firewall. The client must authenticate to this separate IP address before they are allowed to send traffic to the Web server according to the policy. We will assume you have already configured an external authentication server.

Figure 6.6 Authenticating Users with WebAuth



To configure this example via the Juniper WebUI:

1. Enter the WebAuth page by choosing **Configuration | Auth | WebAuth**.
2. Select the **WebAuth Server** you would like WebAuth to use to authenticate users. This may be a local or external authentication server, but you can only configure one WebAuth server globally on the firewall.
3. You can define a **WebAuth Banner** message which will be displayed when the users try to authenticate.
4. We will now define the interface which will accept the WebAuth connections on the virtual **WebAuth IP** address. Go to **Network | Interfaces** and select **Edit** next to the interface you would like to apply this to.
5. In the interface configuration screen, check the **WebAuth** box, and specify the virtual **IP Address** for the WebAuth server.

6. You may select to only accept WebAuth connections over **SSL**. This is recommended for security purposes so that credentials are securely transmitted.
7. Configure a policy via the **Policy** menu, then specify the **From Zone**, **To Zone**, and click **New**.
8. Define the matching terms the firewall will use to match the traffic. This includes the **Source Address**, **Destination Address**, and **Service**. You may optionally define **Anti-virus**, **Deep Inspection**, **VPN Tunnels**, **Logging**, **and Position**.
9. Click **Advanced**.
10. From the other options you may select (**NAT**, **Traffic Shaping**, **Counting**, and **Scheduling**), you must select the **Authentication** option.
11. The **WebAuth** server will be locked as the server you specified in the WebAuth page, but you can still define what **User**, **User Expression**, or **User Group** the server should verify the authentication on.
12. Click **OK**.

For this example, we will use the following configuration.

WebAuth Server	**LDAPServer**
Interface	**Ethernet**
WebAuth IP	**192.168.1.50**
SSL Only	**Checked**
Zone	**From Untrust To Trust**
Source Address	**Any**
Destination Address	**192.168.2.2./32**
Service	**HTTP**
Logging	**Enabled**
WebAuth	**Enabled**
User	**RMolly2**

To configure this example via the Juniper CLI:

```
set webauth server "LDAPServer"
set interface ethernet1 webauth ssl-only
set interface trust webauth-ip 192.168.1.50
set policy id 4 from "Untrust" to "Trust"  "Any" "192.168.2.2/32" "HTTP" permit
    webauth user "RMolly2" log
save
```

Policy-Based Infranet Authentication

As mentioned previously in this chapter, the Infranet authentication serves to help verify that clients accessing a particular resource meet certain security requirements. Infranet Auth can be implemented either with a software client called an agent, or on a policy-by-policy basis.

In this section, we will cover the latter. Of course, before you can use Infranet authentication, you must configure the authentication server as described in the previous section. Once the preliminary task of setting up the server is complete, you simply need to perform the following actions:

Infranet Authentication Settings

Under the policy of interest, you select the Infranet-Auth option in the advanced properties of the policy. You then have the following three options to specify the behavior the firewall should take on the appropriate traffic.

- **No Redirect** This option will not redirect any traffic.
- **Redirect Unauthenticated Traffic** This option will redirect all clear text traffic for Infranet Authentication.
- **Redirect All Traffic** This option will redirect all traffic for Infranet authentication.

NOTE

Users authenticating with Infranet Auth must keep their browser open for the duration of their session; otherwise, they will be forced to reauthenticate to the Infranet Controller.

Configuring Policy-Based Infranet Authentication

In this example, we will configure our firewall to integrate Infranet Authentication into a policy. This is known as **Agentless Infranet Authentication**. This differs from the **Agent Infranet Authentication** because the Infranet Agent is not installed on the client machine. This option is typically used in situations where the Infranet Agent may not be installed on the user's machine. We will specify that all traffic from the Trust zone to the DMZ that is not authenticated will trigger redirection via Infranet Auth. This example assumes you have already configured the IC settings as described in the "Configuring Infranet Authentication" section.

To configure this example via the Juniper WebUI:

1. Create a new Policy rule by going to **Policies**, selecting the **From Zone**, **To Zone**, and clicking **New**.
2. Specify the **Source Address**, **Destination Address**, **Server**, **Logging**, **Antivirus**, **Deep Inspection**, and **VPN tunnel**.

3. Click **Advanced**.
4. You may optionally define **Source NAT**, **Traffic Shaping**, **Scheduling**, and **Counting**.
5. Select the **Infranet Auth** option, and select the appropriate **Redirect action**.

In our example, we will use the following:

Zone	**From Trust, To DMZ**
Source Address	**Any**
Destination Address	**Any**
Service	**Any**
Logging	**Enabled**
Infranet Auth	**Enabled**
Redirect	**Redirect unauthenticated traffic**

To configure this example via the CLI:

```
set policy id 5 from "Trust" to "DMZ"  "Any" "Any" "ANY" permit infranet-auth redi-
    rect-unauthenticated log
save
```

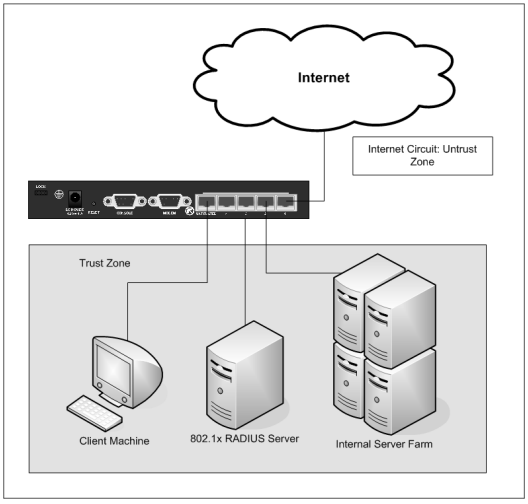
802.1x Authentication

Although there have been many security advancements from the network level and above in recent years, little has been accomplished to address the issues of security at the Data-Link layer. For instance, what would happen if an unauthorized machine got plugged into your internal network. There is still quite a lot that a hacker can accomplish with only LAN or subnet access. In the past, Data-Link security was often controlled by access lists which restricted a port's access by MAC address. This was not a perfect solution since you can easily spoof a MAC address to match one that is permitted. Additionally, other solutions would not protect a network from accessing a network from a machine that was normally allowed. To help even the odds against such malicious activities, the 802.1x standard was created. Essentially, 802.1x provides the facility for user authentication that is not limited to just username and password. Other more modern forms of authentication can be performed with 802.1x such as public key certificates and challenge-response tokens. 802.1x can be used for wireless and wired Ethernet, and is basically a proxy that passes along the authentication information to and from the end user machine (supplicant) and the authentication server. If the authentication server verifies the credentials of the user, and authorizes network access, then the firewall will let it onto the network. All right, it isn't as simple as that, but after you finish reading this section you should be an 802.1x pro!

Components of 802.1x

Before we engage in the fine details of 802.1x, it would be helpful to have a firm understanding of the components that make up this framework. Figure 6.7 provides a visual representation of how all of these components fit together. Note that in this image the Client Machine must authenticate via 802.1x before it is allowed to transmit any other traffic over the network, even within the same zone.

Figure 6.7 802.1x Component Diagram



Extensible Authentication Protocol, or EAP for short, is one of the core components of 802.1x. EAP provides the framework which the supplicant client machine can authenticate to the authentication server. Essentially, EAP provides a layer of abstraction for the authentication so that different types of authentication can be employed to suit the desired authentication method. EAP runs at layer 2 of the OSI model, which makes it independent of layer 3 addressing. EAP has four supported types:

- **EAP-TLS** Requires both client and server authentication via certificates to verify the identity of both sides.
- **EAP-TTLS** Only the server will present a certificate to verify its identity to the client, while the client presents a username and password to verify its identity to the server.
- **EAP-PEAP** Similar to EAP-TTLS but also provides a facility for key-exchange, session resumption, as well as fragmentation and reassembly.

- **EAP-MD5** Uses Challenge Handshake Authentication Protocol (CHAP) to authenticate users. Users are sent a hash value (challenge) and the client machine will make an MD5 hash of the challenge + user password, which will be returned to the authentication server for verification.

The 802.1x server is the external authentication server which validates the client machines' credentials. RADIUS is the type of authentication server supported by the Juniper firewalls for 802.1x.

The Client Machine, also known as the 802.1x supplicant, is the system that needs to authenticate to the network via 802.1x in order to be permitted network access. The supplicant must support 802.1x authentication (either natively, or with additional software).

An intermediate network device accepts the client's 802.1x authentication request and passes it on to the 802.1x server for verification. In our case, this would be the Juniper firewall. Some network switches also support 802.1x.

Configuring 802.1x Authentication

802.1x is disabled by default. It can be enabled on an interface-by-interface basis, and can have different 802.1x servers on a per-interface basis. You must select 802.1x as an account type that your authentication server supports in order for it to be used for 802.1x authentication.

802.1x Settings

Now let's discuss the following 802.1x settings:

- **Port Control** You have two options for Port Control. *Force-Unauthorized* ignores all client authentication attempts and blocks all traffic, while *Auto* allows authentication to proceed normally. If a client successfully passes authentication, they will be permitted to send traffic.
- **Control Mode** Control Mode has two options. One is *Interface*, where MAC addresses of devices connected to the interface are not authenticated as part of the 802.1x Authentication. *Virtual* mode authenticates MAC addresses. This mode is the default, and must always be used for wireless interfaces.
- **Maximum Users** This sets the maximum number of concurrent users that can be enabled on a single interface.
- **Reauthentication Period** This value determines how often a client will be forced to reauthenticate. By default, this is 3600 seconds (1 hour) but it is configurable from 0 to 86400 seconds.
- **Server Name** Any authentication server that has been configured to authenticate 802.1x will appear in this list. This server must be capable of authenticating 802.1x clients, otherwise the 802.1x clients will not work.

- **Silent Period** Specifies the time delay that the firewall will impose after a failed authentication attempt. During this period, the firewall will not attempt to authenticate the client, nor will it accept any authentication attempts. The default is five seconds.
- **Retransmissions** You can enable EAP retransmissions if the client does not properly authenticate. You also can set the number of attempts to resend and the delay that will be employed between attempts.

NOTE

When 802.1x is not enabled on an interface, this is considered a hidden mode called *Force-Authorized*, in which no authentication takes place and all traffic is processed normally.

Configuring an 802.1x Authentication Server

In this example, we will configure a server to support 802.1x authentication and force all users to authenticate via 802.1x before they are allowed to pass traffic. We will only allow them to authenticate if they are coming from a permitted zone by using **Zone Verification**. To set up this example via the Juniper WebUI:

1. Select **Configuration | Auth | Auth-Servers** and select **New**.
2. Specify the **Name** for the server.
3. Define the **IP/Domain Name** for the server.
4. Optionally, define **Backup** servers, and modify the default **Timeout** values.
5. You must specify that the server supports 802.1x as the account type.
6. If you would like to specify a specific interface for the firewall to send traffic out on, do that in the **Source Interface** drop-down menu.
7. Make sure that **RADIUS** is selected as the server type.
8. Define the applicable RADIUS server properties, including **Shared Key**, **Zone Verification**, **RADIUS Timeouts**, **RADIUS Retries**, **Send Calling Station ID**, and **Account-Session-ID Length**.
9. Click **OK**.

For our example, we will use the following configuration:

Server Name	**8021xServer**
IP/Domain Name	**10.15.2.99**

Account Type	**802.1x**
Server Type	**RADIUS**
Shared Key	**J&Fh9ak+=**

To configure this example with the Juniper CLI:

```
set auth-server "8021xServer" id 2
    set auth-server "8021xServer" server-name "10.15.2.99"
    set auth-server "8021xServer" account-type 802.1X
    set auth-server "8021xServer" radius secret "J&Fh9ak+="
    set auth-server "8021xServer" radius zone-verification
    save
```

NOTE

Only RADIUS servers are supported for authenticating 802.1x users, and you will need to load the Juniper Dictionary file into your RADIUS server.

Configuring Ethernet Interfaces to Use 802.1x Authentication

In this example, we will configure an Ethernet interface on the firewall to authenticate users with 802.1x authentication. Client machines will be directly connected into the Ethernet2 and Ethernet3 interfaces on the firewall. We will assume you have already configured an 802.1x authentication server and have appropriately set up the interfaces for network connectivity.

To configure this example via the Juniper WebUI:

1. Go to the interface you wish to configure to use 802.1x, select **Network | Interfaces** and click **Edit** next to the interface you would like to configure.
2. At the top of the screen, click the **802.1x** link.
3. Select **802.1x Enable** to turn on 802.1x for the interface.
4. For **Port Control**, make sure **Auto** is selected.
5. Select **Virtual** for the **Control Mode**.
6. Define how many users may be connected through this interface at a time.
7. If necessary, alter the **Re-Authentication Period**, which has a default value of 3600 seconds.
8. Define the 802.1x server that will be used to authenticate the users with the **Server Name**.

9. If necessary, change the **Silent Period**, which defaults to five seconds.
10. You may additionally specify if **Re-Transmissions** are **Enabled**, as well as the **Re-Transmission Period**, and the **Re-Transmission Count**.

For our example, we will use the following settings:

Interface	**Ethernet2**
802.1x	**Enable**
Port Control	**Auto**
Control Mode	**Virtual**
Maximum User	**1**
Server Name	**8021xServer**
Silent Period	**5**
Re-Transmission	**Enabled**
Re-Transmission Period	**3**
Re-Transmission Count	**3**

Interface	**Ethernet3**
802.1x	**Enable**
Port Control	**Auto**
Control Mode	**Virtual**
Maximum User	**1**
Server Name	**8021xServer**
Silent Period	**5**
Re-Transmission	**Disabled**

To configure this example with the Juniper CLI:

```
set interface ethernet2 dot1x
set interface ethernet2 dot1x auth-server 8021xServer
set interface ethernet3 dot1x
set interface ethernet3 dot1x max-user 1
set interface ethernet3 dot1x auth-server 8021xServer
unset interface ethernet3 dot1x retry
save
```

Configuring Wireless 802.1x Authentication

In this example, we will configure a Wireless on the firewall to authenticate users with 802.1x authentication. The Wireless SSID will use WPA and either TKIP or AES to authenticate and secure wireless communication. We will assume you have already configured an 802.1x authentication server and set up interface Wireless1 for network connectivity.

To configure this example with the Juniper WebUI:

1. Select **Wireless | SSID** and choose **New**.
2. Define a name for the **SSID**.
3. Select **WPA** for the encryption, and specify whether to use **TKIP**, **AES**, or let the user choose with the **Auto** option.
4. Select your **802.1x server** from the drop-down menu of the **Auth Server** menu beside the WPA configuration.
5. Specify what interface to **Bind the SSID** to.

For our example, we will use the following settings:

SSID	**8021xWiFi**
Authentication	**WPA**
Encryption	**Auth (TKIP or AES)**
Auth Server	**8021xServer**

To configure this example with the Juniper CLI:

```
set ssid name 8021xWiFi
set ssid 8021xWiFi authentication wpa encryption auto auth-server
    8021xServer
set ssid 8021xWiFi interface wireless1
save
```

Checking 802.1x Sessions and Statistics

To help provide you with visibility into the functionality and events that have taken place on your network, you can examine the active sessions and statistics on the firewall.

To check active sessions and statistics via the Juniper WebUI:

1. Go to **Network | 802.1x | Statistics**.
2. By default, it will show the statistics for every interface you have 802.1x configured on. You can filter the results by selecting the desired **interface** from the drop-down menu in the top-left corner.
3. To view active 802.1x sessions, go to **Network | 802.1x | Sessions**.

To check active sessions and statistics via the Juniper CLI:

1. You can issue the ***get dot1x*** command to view global settings.
2. Enter the ***get dot1x session*** command to view active sessions
3. To get more detail on a specific session, enter the ***get dot1x session id <session id>*** command.

4. To view interface-based 802.1x information, issue the ***get interface <interface> dot1x*** command.
5. To get the global session statistics for all interfaces, enter the ***get dot1x statistics*** command.
6. To get statistical information only for a specific interface, issue the ***get interface <interface> dot1x statistics*** command.

Enhancing Authentication

The Juniper firewall provides you with a few methods to enhance both the management of your firewall, and also provide a method to convey messages to your users. The first topic we will cover in this section is Firewall Banner messages. These are associated with authentication because these messages are displayed both before and after authentication takes place. They also have significance to the security of your firewall, as well as your network. Next, we will cover Group Expressions, a very powerful tool that allows you to extend the capabilities of your firewall and reduce the amount time and effort it takes to set up a complex authentication scheme.

Firewall Banner Messages

A task often overlooked on many security devices is configuring banner messages to be displayed when a user attempts to log in. There have been several historical reasons for configuring banners, besides the obvious reasons why you would want to post a message to instruct authenticating users. In this section, we will cover the different banners you can configure in the Juniper firewalls.

Configurable Banners

The following banners are configurable parts of the Juniper firewall configuration:

- **Telnet Banner** Users may be shown separate banners for Telnet login prompt, successful authentication, and failed authentication.
- **FTP Banner** Users may be shown separate banners for FTP login prompt, successful authentication, and failed authentication.
- **HTTP Banner** Users may be displayed separate banners for HTTP login prompt, successful authentication, and failed authentication.
- **WebAuth Banner** WebAuth allows you to configure a single banner for successful authentication.

Configuring Firewall Banners

In this example we will configure login prompt, successful authentication, and failed authentication banners for Telnet, FTP, and HTTP. We will also define the default authentication server for the firewall.

To configure this example via the Juniper WebUI:

1. Go to **Configuration | Auth | Firewall**.
2. On this page, you may define the **Default Authentication Server** for the firewall in the drop-down menu. This authentication server will be used any time the **Default** authentication server is referenced as an authentication server in the configuration.
3. Define the messages you wish users to see following the defined action. Each of the messages will only apply for the service that you define them for (for example, Telnet, FTP, or HTTP). Your configuration options are the following:

   a. **Telnet**

      - **Login** Banner message displayed before the login prompt when users connect via Telnet.
      - **Success** Message displayed when they successfully authenticate with Telnet.
      - **Fail** Message displayed when they enter incorrect credentials with Telnet.

   b. **FTP**

      - **Login** Banner message displayed before the login prompt when users connect via FTP.
      - **Success** Message displayed when they successfully authenticate with FTP.
      - **Fail** Message displayed when they enter incorrect credentials with FTP.

   c. **HTTP**

      - **Login** Banner message displayed before the login prompt when users connect via HTTP.
      - **Success** Message displayed when they successfully authenticate with HTTP.
      - **Fail** Message displayed when they enter incorrect credentials with HTTP.

For our example, we will set these messages:

Default Authentication Server	**Local**
Telnet Login Message	**Warning!! You are attempting to log in to a private system! Authorized Users ONLY! All unauthorized use will be prosecuted to the fullest extent of the law!**
Telnet Success Message	**Firewall User Authentication: Accepted**

Telnet Fail Message	**Firewall User Authentication: Failed**
FTP Login Message	**Warning!! You are attempting to log in to a private system! Authorized Users ONLY! All unauthorized use will be prosecuted to the fullest extent of the law!**
FTP Success Message	**230 Authentication - Accepted (Closed connection - reconnect to server)**
FTP Fail Message	**Firewall User Authentication: Failed Unauthorized!**
HTTP Login Message	**Warning!! You are attempting to log in to a private system! Authorized Users ONLY! All unauthorized use will be prosecuted to the fullest extent of the law!**
HTTP Success Message	**Firewall User Authentication: Accepted**
HTTP Fail Message	**401 Unauthorized**

To configure this with the Juniper CLI:

```
set auth default auth server "Local"
    set auth banner telnet login "Warning!! You are attempting to log in to a
        private system! Authorized Users ONLY! All unauthorized use will be
        prosecuted to the fullest extent of the law!"
set auth banner telnet success "Firewall User Authentication: Accepted-"
set auth banner telnet fail "Firewall User Authentication: Failed Unauthorized"
    set auth banner ftp login "Warning!! You are attempting to log in to a
        private system! Authorized Users ONLY! All unauthorized use will be
        prosecuted to the fullest extent of the law!"
set auth banner ftp fail "Firewall User Authentication: Failed  Unauthorized!"
    set auth banner http login "Warning!! You are attempting to log in to a
        private system! Authorized Users ONLY! All unauthorized use will be
        prosecuted to the fullest extent of the law!"
set auth banner http success "Firewall User Authentication: Accepted-"
set auth banner http fail "401 Unauthorized!"
save
```

WARNING

Use caution when defining your authentication banners. You should not configure messages that include words such as *welcome*, *enter here*, and *open access* on systems that should have restricted access. In the past, Hackers have been able to successfully argue that a system with a message welcoming users was confusing and did not properly disclaim private access.

Group Expressions

Building on the foundations we have covered, Group Expressions will stretch your administrative options even further. Of course, you could probably accomplish your goals with simple objects, but Juniper has provided you with a way to further customize how Juniper evaluates objects. Group Expressions use simple AND, OR, NOT logic to provide another dimension to evaluating objects.

Group Expression Properties

Essentially, Group Expressions consist of the two operands (Users, User Groups, User Expressions) and an operator (AND, OR, NOT). The result of the expression will be used by the firewall to perform the necessary operation. A couple of important things you should know about Group Expressions include the following:

- Group Expressions can be applied in policies to determine if an authentication should take place on a user if the user matches in the Group expression.
- Group Expression operands may only be taken from external Auth servers. You cannot use locally defined objects for Group Expressions.

Configuring Group Expressions

In this example, we will make three different group expressions based upon External Group objects. We will use the AND, OR, and NOT operators.

To configure this example with the Juniper WebUI:

1. First, we will create three External Groups by going to **Objects | Users | External Groups** and selecting **New**.
2. Specify the **Group Name**, and select what **Group Types** will be supported. You can define **Auth**, **XAuth**, and **L2TP**.
3. Create a Group Expression by going to **Objects | Group Expressions** and selecting **New**.

4. Define the **Group Expression** name.
5. Select the **Operator** (AND, OR, NOT) and define the operators in the adjacent fields. These operators may be the External Groups previously defined.
6. Click **OK** to apply the Group Expression.

For our example, we will use the following settings:

External Group	**AllAuthExtGroup**
Group Type	**Auth, XAuth, L2TP**
External Group	**L2TPExtGroup**
Group Type	**L2TP**
External Group	**XAuthExtGroup**
Group Type	**XAuth**
Group Expression	**Exp1**
Expression	**L2TPExtGroup OR XAuthGroup**
Group Expressions	**Exp2**
Expression	**NOT L2TPExtGroup**
Group Expression	**Exp3**
Expression	**AllAuthExtGroup AND XAuthExtGroup**

To configure this example with the Juniper CLI:

```
set user-group "AllAuthExtGroup" id 10
set user-group "AllAuthExtGroup" location external
set user-group "AllAuthExtGroup" type auth l2tp xauth
set user-group "L2TPExtGroup" id 8
set user-group "L2TPExtGroup" location external
set user-group "L2TPExtGroup" type l2tp
set user-group "XAuthExtGroup" id 9
set user-group "XAuthExtGroup" location external
set user-group "XAuthExtGroup" type xauth
set group-expression "Exp1" id 0
set group-expression "Exp1" "L2TPExtGroup" or "XAuthExtGroup"
set group-expression "Exp2" id 1
set group-expression "Exp2" not "L2TPExtGroup"
set group-expression "Ext3" id 2
set group-expression "Ext3" "AuthExtGroup" and "XAuthExtGroup"
save
```

Summary

This chapter has focused on the features and functionality of user authentication in the Juniper firewall. We began the chapter with a discussion of the different user types, attributes, and practical implementation. Each user type has unique qualities which lend to different possible solutions for different authentication needs. Building upon the different user types, we explored the different types of authentication servers which can be used to authenticate users for the firewall. We learned about the differences between the various account types, as well as some best practices for implementing the account type that most appropriately fits your needs. In conjunction with user types and authentication servers, we studied the different ways of extending user authentication to firewall policies, and to secure the network at the Data-Link layer with 802.1x. Lastly, we expanded your administrative capabilities with extensions to the firewall logic with Group Expressions.

Solutions Fast Track

User Account Types

- ☑ The Juniper firewall supports six different user account types: Auth, IKE, XAuth, L2TP, 802.1x and Admin.
- ☑ Admin users may be authenticated locally or externally. An external server configured in the Juniper firewall may not support any other user accounts if it supports Admin accounts.
- ☑ When users are added to a group, that group inherits the user types of its members.
- ☑ IKE users may only be stored locally on the firewall. If additional authentication is required, XAuth users make a good choice.
- ☑ 802.1x users are only supported through RADIUS authentication servers loaded with the Juniper dictionary file.

Local and External Authentication Servers

- ☑ The local authentication server exists on the firewall, while external authentication servers integrate third-party authentication solutions to the firewall.
- ☑ Juniper currently supports RADIUS, SecurID, LDAP, and Infranet Auth as external authentication servers.
- ☑ When Juniper authenticates to an external authentication server, it acts as a client requesting authentication.

- ☑ Backup authentication servers must have the same properties as the primary server (such as a shared key) in order for authentication server failover to work properly.

Policy-Based User Authentication

- ☑ Firewall policies may be employed to authenticate users before they are allowed to pass traffic matching that policy.
- ☑ Policy-based authentication can support a firewall Auth, WebAuth, and Infranet Auth on a policy-by-policy basis.
- ☑ WebAuth provides an artificial authentication server which the user must authenticate to before being permitted to pass traffic through. This compares to firewall Auth, where a user will get prompted for credentials when they try to pass traffic through a particular policy.

802.1x Authentication

- ☑ 802.1x functions at the Data-Link layer, and provides administrators a way to authenticate users before they can access the network. This can operate on both wired and wireless Ethernet.
- ☑ 802.1x provides the capability to audit the users who log in to your network.

Authentication Enhancements

- ☑ Group Expressions provide a way to use additional logic to build expressions for matching users. Group Expressions can be used with External Groups only.
- ☑ Banners provide a way to post a message to users attempting to authenticate to the firewall, as well as display custom messages for successful authentication, and failure.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: Can I make a user a member of multiple different account types?

A: Yes, you can certainly make a user a member of different account types such as IKE, L2TP, XAuth, and Auth. However, there are some downsides to configuring a user as a member of multiple account types. For instance, IKE users may only be stored locally on the firewall, so if a user is configured as an IKE user, you cannot get the benefits of using an external server to authenticate the L2TP, XAuth, and Auth. It definitely pays to take some time to define your requirements and choose the scheme that will make the most sense.

Q: What happens if I have a user try to log in as an L2TP-over-IPSec VPN when both L2TP and XAuth are enabled?

A: L2TP and XAuth should not be used together in a single session. This is because they provide similar services, and will create conflicts between each when they are both enabled. Instead, you should only enable one per connection.

Q: What happens to local authentication if I have a user that is an Auth user as well as an L2TP or XAuth user?

A: If a user is both Auth and L2TP or XAuth, you will need to make sure the user has the same username and password.

Q: Can I use Local Groups in Group Expressions?

A: No, you can only use External Groups in Group Expressions.

Q: Can External Groups support multiple account types?

A: Yes, just like Local Groups, you can combine account types. You can have any combination of Auth, L2TP, and XAuth per External Group.

Q: How come an authentication server that supports the Admin account type will not authenticate other account types (such as Auth, L2TP, XAuth, or 802.1x)?

A: External authentication servers that support Admin users will not support a combination of the other account types. This is only true of Admin users; you can still combine Auth, L2TP, XAuth, and 802.1x.

Q: What types of External Authentication servers support 802.1x?

A: Only RADIUS supports 802.1x, and you will need to make sure your actual RADIUS server supports the 802.1x extensions.

Q: Can I put client machines behind an Ethernet switch on an 802.1x interface?

A: No, clients must either be directly connected to the firewall's Ethernet interface, or must be behind a hub. The reason for this is because a switch will not authenticate via 802.1x to the firewall, so any client behind the switch will not be able to authenticate since the 802.1x interface on the firewall will remain in a blocking state. This is only true for Ethernet, and is not the case for wireless devices.

Q: Authentication servers have several optional settings such as *Stripping Separator*, *Domain Name*, *Revert Interface*, *Send Calling Station ID*, and *Account-Session-Length ID.* How do I know when to use these?

A: The use of these optional parameters will depend on your external authentication server and how you have it configured. Discussion of how each of these parameters should be set (beyond an explanation of their functionality) is outside the scope of this book. You should consult your authentication server user guide for clarification on how it functions, and what parameters it requires.

Chapter 7

Routing

Solutions in this chapter:

- Virtual Routers
- Static Routing
- Routing Information Protocol
- Open Shortest Path First
- Border Gateway Protocol
- Route Redistribution
- Policy-Based Routing

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

Since the dawn of networking, routing has provided administrators with the ability to configure and scale enterprise environments. Traditionally, routing has been performed by purpose-built hardware, which only performed that particular function. Recent trends have blurred the lines between a router, a switch, and even a firewall. Advancements in hardware, along with innovative software designs have enabled manufacturers to integrate many of these features into a single device. Network administrators have reaped the rewards of these advancements by creating more robust and secure networks, with less hardware.

Juniper has gone to great lengths to equip their firewalls with routing capabilities equivalent to traditional purpose-built routers. With the advent of the SSG series firewalls, which can support WAN interfaces, the need for integrated routing is even more pressing. The NetScreen and SSG series firewalls support a wide range of routing protocols, as well as multicast and IPv6 traffic. After all, proper routing architecture can lead to better performance and security, so it should come as no surprise that these functions have been integrated within a single device.

In this chapter, we will begin our routing expedition with an overview of the routing concepts conventions within the Juniper firewalls. Following that, we will get down to business with an in-depth look at the routing protocols supported by the Juniper firewalls. This discussion will include both static and dynamic routing protocols. Lastly, we will cover policy-based routing, which is a brand new feature of the Juniper firewall family. By leveraging the extended routing support of your firewall, you should be able to posture your network to balance both security and performance.

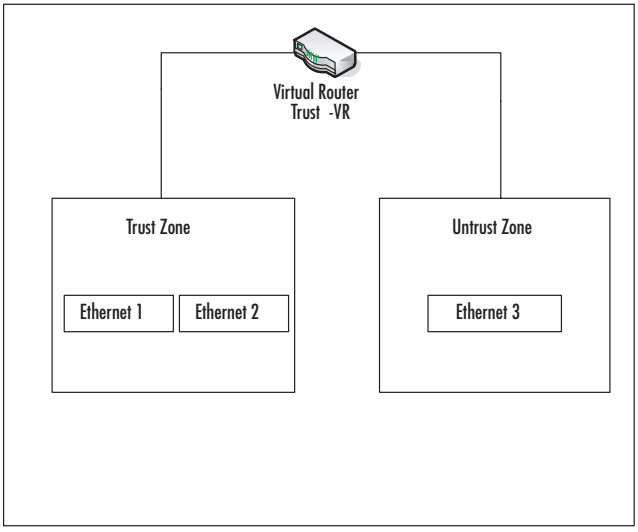
Virtual Routers

At the center of the Juniper firewall's routing infrastructure lies the virtual router (VR). The VR provides the same routing support as a physical router, but allows you to create multiple routing instances on a single box. Functionality-wise, each VR operates independent of the other VRs located on the same firewall or elsewhere in the network. They even maintain their own routing tables and protocol configurations. The real power behind VRs is the option to granularly control routing within the Juniper firewall. If you feel granularity is a common theme in many Juniper products, you're right! This is no coincidence; it's a well–thought out design. As you know, interfaces are applied to zones, and zones are applied to virtual routers. Therefore, VRs empower you to segment the routing of your network to a completely different router. Of course, you don't have to segment the routing on your firewall, but you have the capability to do this if you wish. In this section, we will begin with a review of what facilities the firewalls come with by default, followed by a look at the capabilities of the virtual routers.

Virtual Routers on Juniper Firewalls

Juniper firewalls come configured with two VRs right out of the box. The Trust-VR and Untrust-VR cannot be deleted, although they can be modified. By default, firewall considers the Trust-VR the default router. Optionally, you can add other VRs, or delete custom VRs, as you see fit. VRs can even route traffic between each other. Again, think of them as separate physical routers, although they are really located on a single device. You may be asking yourself, how does the firewall know which VR to use to route traffic if multiple VRs exist on the device. The decision of which VR will handle the routing for that particular traffic is made based upon what interface the traffic arrived on, and subsequently the zone that belongs to that specific VR. See Figure 7.1 for a visual representation of how the interface applies to a zone, which applies to a VR. As mentioned before, a VR supports a full set of routing functionality. It can support static and dynamic routing, and even multicast and IPv6, all on a single VR.

Figure 7.1 The VR–Zone–Interface Hierarchy



NOTE

The term *router* here can be used interchangeably with the word *firewall*. Since the Juniper firewall contains a very extensive routing suite, we can also think of it as a router. The word *peer* may also be used interchangeably with the word *neighbor*, meaning another router that participates in the respective routing protocol.

Different Route Types

Juniper firewalls essentially can support five different route types in each virtual router. Each route type has a specific method for learning its routes. Juniper has created a few route types that are unique to the Juniper firewall platforms, so those of you that have lots of prior routing experience might learn a new concept or two!

- **Directly Connected Routes** These routes are automatically generated and placed into the routing table based upon the subnet an interface is attached to. For instance, if you have an interface 192.168.1.1/24, then the route 192.168.1.0/24 will be placed into the routing table. These routes cannot be modified, and will only be removed when the interface is removed or the address changes.
- **Host Routes** The term *Host Route* refers to a route that is automatically generated for an interface on the box. The firewall will automatically generate a route for the interface and place it in the routing table. For instance, if you have an interface 192.168.1.1/24, a route 192.168.1.1/32 will be placed in the routing table. This route cannot be modified, and will only be deleted if the interface is removed or the IP address is changed.
- **Static Routes** Static routes are manually created and direct the firewall to the next routing device, which then routes the traffic to its ultimate destination. Static routes are created by the user. For example, a static route might route traffic to a remote destination network *10.1.1.0/24* and to the next hop *192.168.2.254/24*. (We will cover many examples of this in the section "Static Routing" later in the chapter.)
- **Dynamic Routes** These routes are learned by a dynamic routing protocol such as Routing Information Protocol (RIP), Open Shortest Path First (OSPF), or Border Gateway Protocol (BGP). The administrator does not have to manually program routes to remote networks since they are calculated by the protocol. The best route (according to the configuration) will then be injected into the routing table. (We will cover different routing protocols extensively later in this chapter.)
- **Routes to Other VRs** The last type of route is a route that directs traffic to another VR located on the firewall itself. For instance, you might say to route traffic to *10.1.2.0/24* to the Untrust-VR instead of another network. Of course, the Untrust-VR must have some sort of routing knowledge to be able to forward the traffic further.

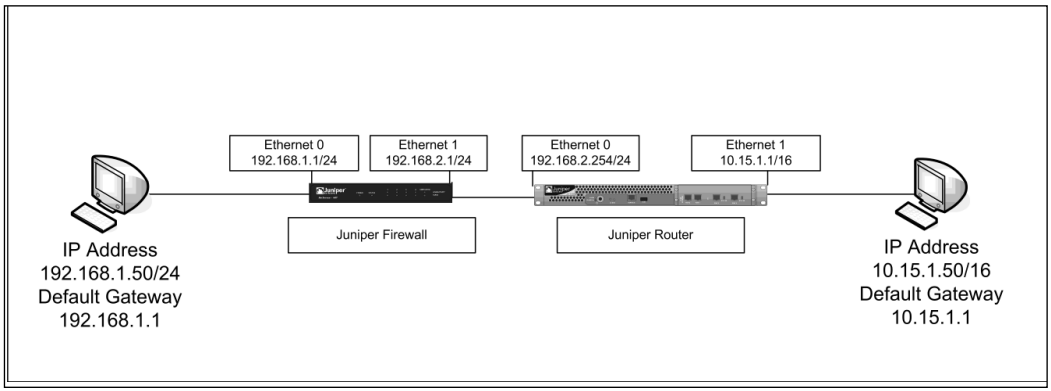
Different Routing Tables

Four different types of routing tables can technically exist on each VR. These routing tables route traffic based upon different networking fundamentals. Each routing table is covered in this section.

Destination-Based Routing Table

This is what most network administrators would typically think of a routing table as. The destination-based routing table contains unicast routes, in which routing decisions are made based upon where the traffic is slated to go. An example of a destination-based routing decision could be the following: A host at the IP address 192.168.1.50 wants to send traffic to 10.15.1.50. The source host has a default route pointing to 192.168.1.1. The routing table on the firewall might have a route of 10.15.0.0/16 via 192.168.2.254, so the firewall would direct this traffic to 192.168.2.254 as its next hop. This decision to route traffic to 10.15.1.50 via 192.168.2.254 would be made based upon where the traffic is going (10.15.0.0/16). Figure 7.2 is a visual layout of this example.

Figure 7.2 Destination-Based Routing



Source-Based Routing Table

Source-based routing is a form of routing where decisions on where to route traffic to, are based upon where the traffic came from. This is a less common form of routing, but real-world application of this technique is still implemented today. An example of source-based routing would be the following: Traffic with a source IP of 192.168.1.50, slated for 10.15.1.50, would be routed via 192.168.2.254 on the firewall based upon the fact that it came from 192.168.1.0/24. So, essentially, the routing decision would send traffic from 192.168.1.0/24 to the next hop of 192.168.2.254 (regardless of the destination). See Figure 7.2 for an example of this layout.

Source Interface–Based Routing

Known as SIBR for short, source interface–based routing makes decisions on where to route traffic according to the inbound interface that the traffic arrived from on the firewall. Again, this type of routing is not as common as destination-based routing, but it is still practiced. An

example of this would be a host 192.168.1.50 sending traffic to another host at 10.15.1.50. The traffic would be sent to the firewall, which would then forward it to 192.168.2.254 based upon the fact that the source interface that the traffic arrived on was Ethernet 0. So, basically, any traffic arriving on Ethernet 0 will be routed to 192.168.2.254 regardless of its source or destination IP address. See Figure 7.2 for a visual representation of this layout.

Multicast Routing Table

The Juniper firewall holds a separate routing table for multicast traffic. Traffic from a multicast source can be directed out a specific interface based upon a combination of the following: source IP, multicast group, and incoming interface. That would be an example of a static multicast route. The firewall can also translate the multicast group based on this route. Multicast routes can also be dynamically defined based upon the Protocol Independent Multicast (PIM). Juniper supports this protocol, but it is not within the scope of this book.

Routing Selection Process

With so many options regarding what route types and routing tables to use, you might think the process that the Juniper firewall takes to select the route is extremely complex. But from a high-level perspective, it is actually pretty straightforward. The firewall evaluates the routing decision based upon the following order and criteria for unicast traffic:

1. The router must first determine which VR to use. It looks at what interface the traffic arrives on, then what zone that interface belongs to, and, finally, what VR the zone belongs to.
2. The VR next determines whether to use the Destination-Based routing table, Source-Based routing table, or Source Interface–Based routing table. It determines which one to check based upon the preference you set for each routing table (in the VR). The VR evaluates each table (if applicable), from the lower preference table to the highest preference table, trying to find a match for the route.
3. Routes are chosen from the most specific to the least specific when matching. For instance, both 10.1.1.0/24 and 10.1.0.0/16 would match for the host 10.1.1.50, but since 10.1.1.0/24 is more specific, it would match that route first.
4. The VR would next determine which route protocol to use for the route if the same route is announced by different protocols. For instance, if both OSPF and a static route are configured for the same destination of 10.1.1.0/24, then static by default would win because it has a lower route preference (Static=20, OSPF=60).
5. If there are multiple static routes for the same destination, then the route with the lowest metric for that protocol would be selected. For instance, if there are two routes for OSPF to the destination 10.1.1.0/24, one with a metric of 20, and the other with a metric of 30, then the route with the lower metric (20) would win.

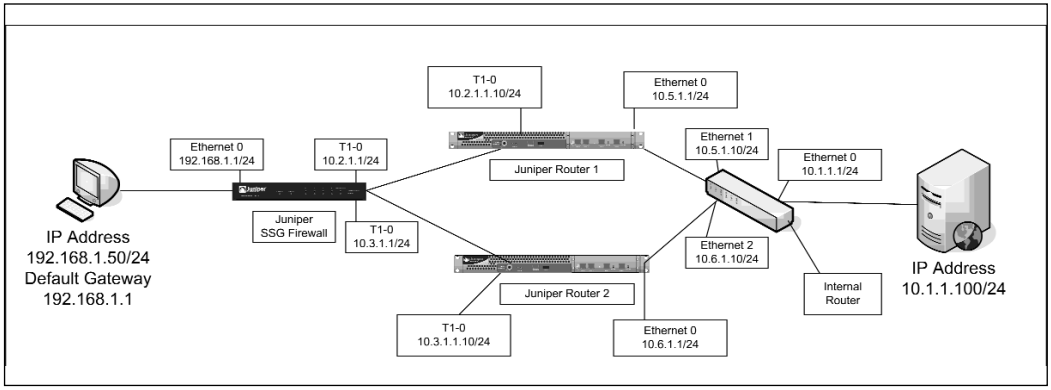
NOTE

Remember that route preference and route metrics are not the same thing. Route preference is used to choose which routing protocol the VR will select the route from (between two or more routing protocols). The protocol with the lowest preference will win. On the other hand, route metrics are used to pick which route within a protocol will be selected to route the traffic, with the lower value being selected. Of course, these attributes are only evaluated if the same route exists in the same VR table, but is advertised by multiple protocols, or multiple routes are announced. If you only have one route, that will be the only route available to use.

Equal Cost Multiple Path

If you were paying especially close attention in the preceding "Route Selection Process" section, you might have been left wondering, could there be multiple routes to the same destination, within the same protocol, using the same route metric? This case is not as unique as you might think, and the Juniper firewall would be happy to support this! The most common use of Equal Cost Multiple Path (ECMP) is to load-balance traffic over two or more routes to the same destination. Let's say you had two WAN links to the same network, and you want to utilize them equally to get maximum throughput between your remote site, you could use ECMP. This is shown in Figure 7.3. In this example, you have a host with an IP address, 192.168.1.50, which would like to send traffic to a server at IP address 10.1.1.100. The host has a default route pointing to the firewall's Ethernet 0 interface 192.168.1.1. The firewall is connected to two WAN routers, which connect to a switch on the remote end, which also connects into the server. The firewall has two routes to 10.1.1.0/24, both with the same metric, say 15. So essentially, the firewall will take turns choosing which WAN link to send the traffic over to reach the exact same destination.

Figure 7.3 Equal Cost Multiple Path Routing



NOTE

ECMP routes traffic on a per-session basis, not a per-packet basis. This is especially useful if you have a stateful firewall that the traffic passes through downstream. If you did ECMP on a per-packet basis, the stateful firewall would drop traffic since it would often be out of state. So, essentially, if a TCP session goes through the firewall, the rest of the traffic for that session would follow that route, but if another TCP session goes through the firewall, it could be routed on a different route. The routes are chosen in a round-robin fashion to ensure equal utilization at the session level.

Virtual Router Properties

Since the VRs are so flexible and can cover such a wide range of types, one would be correct in assuming there are many configuration options that exist within each VR. Luckily for you, we will simplify the maze of configuration options with a byte-for-byte explanation of each function (well, pretty close to byte-for-byte). Each of the following terms are configurable properties for each VR.

- **Virtual Router Name** Each VR must have a name that is unique on each separate firewall.
- **Virtual Router ID** You may either leave this value *non-initialized*, or provide a *Custom* ID for this VR. This is commonly configured as a unique routable IP address on your network.
- **Maximum Route Entry** Each VR allows you to set the maximum number of route entries for that particular VR. You can either manually set a maximum route limit, or you can allow it to support an unlimited number of routes. Note that this is not truly unlimited, but rather is limited to the capabilities of the individual platform for how many routes it and its VRs can support.
- **Maximum ECMP Routes** ECMP stands for Equal Cost Multiple Path. It allows you to define different routes to the same destination. Each of these routes is considered to be equally preferred by the router. In this field, you can specify how many equal cost paths can be defined to a single destination. You can either have it Disabled, or you can define whether there can be 2, 3, or 4 ECMP routes.
- **Route Lookup Preference** This field defines the order which the VR will search for routes between the Destination-Based routing table, Source-Based routing table, and Source Interface–based routing table. You set the preference (1 to 255) for each table, with evaluation going from the lowest to the highest valued routing table.

- **Shared and Accessible by other VSYS** Enabling this setting allows the VR to be accessed by other VSYS besides the one the VR was defined for.
- **Ignore Subnet Conflicts for Interfaces on this VRouter** This setting allows the VR to ignore subnet overlaps between different interfaces on the router. If you do not have this checked, the firewall will return an error if you try to configure overlapping subnet addresses on an interface. This is off by default.
- **Make this VR the Default VR for the System** By default, the Trust-VR is the VR for the system. By setting this, you will configure the VR to be the default value.
- **Auto-Export Routes into Untrust-VR** This automatically exports routes in the VR which it has configured into the Untrust-VR.
- **Enable Source-Based Routing** This allows you to enable source-based routing on the VR.
- **Enable Source Interface–Based Routing** This allows you to enable source interface–based routing on the VR.
- **Advertise Routes on Inactive Interfaces** This setting lets you advertise routes for interfaces that are not active. This is not a default behavior and must be enabled manually.
- **Sync VR to NSRP Peer** You would want to select this option if you have a cluster of firewalls and need to enable the VR to sync to its peer.
- **Route Preference** This allows you to set the route preference for each routing protocol. The lower preference routing protocol will be selected over a higher preference routing protocol if contention occurs. Default values for each of the protocols are as follows:

  Connected: 0

  Static: 20

  Auto-Exported: 30

  iBGP: 40

  OSPF: 60

  RIP: 100

  Imported: 140

  eBGP: 250

  OSPF External Type 2: 254

Configuring a Virtual Router

In this example, we will configure a new virtual router on our firewall. This VR will be a new instance, which can be used in conjunction with the predefined VRs to segment the routing table.

To configure this example via the Juniper WebUI:

1. Select **Network | Routing | Virtual Routers**, and then click the **New** button in the top right-hand corner.
2. Define the **Virtual Router Name** that will be used to reference this VR in the rest of the system. This must be a unique VR name. The number of VRs you can create on the firewall is dependent on what hardware platform is used.
3. Define a unique **Virtual Router ID** for this VR instance. You can use the **System Default**, or you can configure your own **Custom** value.
4. If you would like to limit the number of routing entries this VR can handle, you may define that in the **Maximum Route Entry**, or leave it set to **Unlimited** so as not to impose any restriction.
5. If you are performing ECMP, you must define how many ECMP routes you will support on this VR under the **Maximum ECMP Routes**.
6. By default, the firewall supports destination-based routing. If you would like to support other types such as source-based routing (SBIR), you can define what order the routes should be selected in by using the **Route Preference**.

NOTE

You must define the preference values for destination-based routing, source-based routing, and SBIR.

7. Since we are not using any other VSYS in this example, we will leave the **Shared and Accessible by Other VSYS** box unchecked.
8. We will not check the **Ignore Subnet Conflict for Interfaces on this VRouter** since we do not want this functionality is.
9. We want to keep the Trust-VR as its default, so we will not make the new VR we created the default for this example. If you did want to make this the default, you would select the **Make this VRouter the Default VRouter for the System** option.

10. It's not necessary to check the **Auto-Export Routes to Untrust-VR** option in this example, but if we wanted to do so without having to configure Route Maps, we could select it.
11. In this example, we will only use destination-based routing, so we won't enable source-based or SBIR.
12. By default, you usually do not want to advertise **Routes on Inactive Interfaces**, so we will leave this disabled.
13. This example won't use an NSRP peer, so we will not enable that for this instance, but if you did have high availability enabled, you would probably want to check **Sync VR with NSRP Peer**.
14. Leave all of the **Route Preferences** at their default value.
15. Click **OK**.

In this example, we are using the following values:

Virtual Router Name VR-1025

Virtual Router ID System Default

Maximum Route Entry Unlimited

Maximum ECMP Routes Disabled

Route Lookup Preference Disabled

Shared and Accessible Disabled

Ignore Subnet Conflict Disabled

Make this the default VR Disabled

Auto Export Routes to Untrust-VR Disabled

Enable Source Based Routing Disabled

Enable SIBR Disabled

Advertise Routes on Inactive INT Disabled

Sync VR with NSRP Peer Disabled

Route Preferences Default Values

To configure this example on the CLI, perform the following steps:

```
set vrouter name "VR-1025" id 1025
unset vrouter "VR-1025" nsrp-config-sync
set vrouter "VR-1025" auto-route-export
save
```

Changing Default Route Preferences

In this example, we will alter the default route preferences for OSPF, BGP, RIP, and static routes. This technique is often used when you need to customize how the firewall should prefer one routing protocol over another (such as preferring OSPF routes over BGP routes). We will configure this example on VR-1025.

To configure this example via the Juniper WebUI:

1. Go to **Network | Routing | Virtual Routers**.
2. Click the **Edit** hyperlink next to the VR in which you would like to change the Route Preferences.
3. Change the values needed next to the appropriate protocol. In our example, we will set **Static** to **40**, **OSPF** to **50**, **RIP** to **140**, and **BGP** to **60**.
4. Click **OK**.

In this example, we will use the following configuration:

Virtual Router VR-1025

Static 40

OSPF 50

RIP 140

eBGP 60

To configure this example via the Juniper CLI:

```
set vrouter "VR-1025" preference static 40
set vrouter "VR-1025" preference ebgp 60
set vrouter "VR-1025" preference ospf 50
set vrouter "VR-1025" preference rip 140
save
```

Using Destination-Based Forwarding

Destination-based forwarding is enabled by default, so there isn't much to do to make this work. We will cover the creation of routes for destination-based forwarding, as well as the various dynamic routing protocols, in the "Static Routes" section later in this chapter.

A Source-Based Routing Example

To help solidify your knowledge of source-based routing, we will now present a scenario where you would like to forward traffic based upon the source address of the host sending the traffic. To do that, you must turn on source-based routing, which is what we will do in this example. This example builds upon our previous configuration of the new virtual router VR-1025.

To enable source-based routing in the Juniper WebUI:

1. Select **Network | Routing | Virtual Routers** and click the **Edit** hyperlink next to the VR you wish to enable source-based routing on.
2. Select the **Enable Source Based Routing** checkbox.
3. Click **OK**.

To configure this in the CLI, set vrouter "VR-1025" source-routing enable and save.

A Source Interface–Based Routing Example

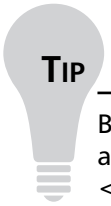
For thoroughness, we will cover an example that uses SIBR to route traffic based upon the firewall interface it arrived on. In this example, we will enable SIBR on the router, and then set the route preference to 1 for SIBR, 2 for Source Based Routing, and 3 for Destination Based Routing. This will cause SIBR to be checked first, then source-based, and lastly destination-based routing.

To configure this example through the WebUI:

1. Select **Network | Routing | Virtual Routers** and click the **Edit** hyperlink next to the VR you wish to have SIBR on.
2. Enable the **Route Preference** checkbox.
3. Define **Destination Based Routing** as **3**, **Source Based Routing** as **2**, and **Source Interface Based Routing** as **1**.
4. **Enable Source Interface Based Routing**.
5. Click **OK**.

To configure this example in the CLI:

```
set vrouter "VR-1025"
set route-lookup preference destination-routing 3 source-routing 2 sibr-routing 1
set sibr-routing enable
save
```



TIP

By typing in the *set vrouter <vrouter_name>* command, you enter the VR; thus, all subsequent commands do not have to be preceded by the *set vrouter <vrouter_name>* command. This can save you time, especially when you have many commands to type under a single VR.

Route Maps and Access Lists

As if virtual routers, dynamic protocols, and equal cost multiple path routing wasn't enough, Juniper also supports another method to perform several routing tasks. Since the Juniper firewall has router functionality beneath the firewall surface, it shouldn't be a surprise that this is supported on the firewall platforms, too. Some network administrators feel a bit intimidated by route maps at first glance, but I can assure you they are not as painful as they might appear. Before we get too far into explaining route maps, however, let's discuss the access-list, which is often the building block of a route map.

Access Lists

Access lists have been widely implemented across many different types of platforms in computer networks. They are typically thought of as a way to permit or deny traffic based upon certain matching conditions, such as source IP, destination IP, service, and so on. However, when discussing access lists in this chapter, we will refer to them as a mechanism to match IP addresses and subnet masks for routes. Note that just like the firewall rulebase, access lists are evaluated from the top of the list down until a match is found. The action will either be *permit* or *deny*. In terms of access-lists for routing, *permit* means you match this route and perform some other actions, while *deny* means you have matched the route, but do not take further action.

NOTE

The access lists discussed here are simple routing access lists. These differ from the policy-based routing extended access lists, which we will discuss later in the chapter. Routing access lists only match on an IP address / subnet mask, while the extended access lists can match on many more properties than that. Please remember that they are not interchangeable.

Access List Properties

Juniper allows you to configure the following properties for each access list:

- **Virtual Router** Each access list is created within a specific VR.
- **Access-List ID** This number uniquely defines the access list. It can be set to any number between 1 and 99.
- **Sequence Number** This number specifies the position in the access list that this particular entry will be placed. Remember, order is important!
- **IP Address / Netmask** This is the route you want to match with the access list.

- **Action** You can either permit or deny the route if it matches a particular rule, as explained earlier.

Configuring an Access List

In this example, we will create an access list within a VR to be used in later examples with route maps. This access list will match two routes which it will deny, while it will permit the third route in the access list.

To configure this example in the Juniper WebUI:

1. Enter the VR you would like to configure the access list under by going to **Network | Routing | Virtual Routers** and clicking the number under the **Access List** column of the VR you would like to configure the access list on.
2. Click the **New** button in the upper right-hand corner.
3. Define an **Access List ID**.
4. Give a numerical **Sequence Number** for this access list entry.
5. Put an **IP Address** and **Netmask** in the respective fields you would like to match a route in.
6. Select either **Permit** or **Deny** depending on the action you would like to take when this route is matched.
7. Repeat the last three steps as many times as needed to add statements to an access list. Of course, you can also create a whole new access list, which will have a different access list ID.
8. Click **OK**.

In this example, we are using the following settings:

Virtual Router VR-1025

Access List ID 1

Sequence Number 1

IP Address/Netmask 192.168.1.0/24

Action Deny

Sequence Number 2

IP Address/Netmask 192.168.2.0/24

Action Deny

Sequence Number 3

IP Address/Netmask 192.168.3.0/24

Action Permit

To configure this example in the Juniper CLI:

```
set vrouter "VR-1025" access-list 1
set vrouter "VR-1025" access-list 1 deny ip 192.168.1.0/24 1
set vrouter "VR-1025" access-list 1 deny ip 192.168.2.0/24 2
set vrouter "VR-1025" access-list 1 permit ip 192.168.3.0/24 3
save
```

Route Maps

Route maps take the concept of access lists to the next level. Route maps allow you to match routes on more specific values, such as metrics, communities, and next hops to name a few. Route maps also allow you to change values in routes if you desire. The main use of route maps is to filter routes that are either advertised to the firewall (inbound) or advertised from the firewall to other routing devices (outbound). They not only allow you to restrict what is advertised (such as removing a route from the advertisement that would normally be passed to another routing peer), but they also allow you to change the values on the routes as well. We will begin this section by covering the properties of the route map. Afterward, we will use some examples to solidify the concepts covered.

Route Map Properties

The following list is composed of the properties you can configure for each route map. You must create an access-list first in order to be able to reference it in the route map to match the route.

- **Virtual Router** This is the VR the route map belongs to. Route maps are created within a VR and only apply to that VR.
- **Map Name** The name of the route map in this field.
- **Sequence No** Just like an access list, you can define where the route map is placed in a list of route maps evaluated top to bottom.
- **Action** You can either permit or deny the route with this route map. If you permit, you can additionally define changes; if you deny, the route won't be allowed to advertised.
- **Match Properties** These are properties of the route that the route map will use to match on. You do not have to use all of these options, only the ones you need.
  - **Metric** This is the metric of the route which you want to match.
  - **Community List** You can define the BGP community list if desired to help match the route.

- **Tag** This refers to the route tag which you can add as a property to match.
- **AS Path** The BGP path you can match within the route map.
- **Access List** In this field, you can select any access lists you want to use to match the routes. The access lists will match the IP addresses, while the route map's other match properties will essentially act as an extension to the access list. It's usually best to create the access lists first before you create the route map.
- **Next-Hop** Allows you to match next hops for routes.
- **Route Types** This lets you match on a routing protocol type of route. These include internal OSPF, iBGP (local BGP), external OSPF type1, and external OSPF type2.
- **Interface** This matches what interface the route will use for the outbound interface sending the packets.

- **Set Properties** You can use route maps to set values in the routes you are advertising after a match has been made. You can set the following properties for advertising routes.
  - **Metric** You can change the value of the route's metric.
  - **Off Metric** This value allows you to increment the metric of the route by the *Off Metric*. So, essentially, this calculation is the *Advertised Metric = Base Metric + Off Metric.*
  - **Metric Type** This field is for OSPF, and allows you to specify the metric as *Type 1* or *Type 2*.
  - **Tag** This allows you to set the route tag.
  - **AS-Path** You can use this field to set the AS Path
  - **Community List** This allows you to set the community list
  - **Next Hop** Setting a value here will change the Next Hop of the route, which will then be advertised to other routing peers.
  - **Weight** This is a value which the VR can use to help determine the attractiveness of the route within the VR.
  - **Local Preference** This value allows you to set the local preference for a BGP route.
  - **Origin** This specifies how a route was learned. Your choices are to leave it as *None*, *IGP*, or *Incomplete.*
  - **Preserve** This allows you to preserve the metric or the local preference setting when putting the route through the route map.

Route Map Example 1

In this example, we will configure a route map which will match routes defined by an access list, then set their metric to 150. We will use the same VR (VR-1025) which we created in a previous example, as well as Access List 1, which we also previously created.

This example can be configured on the WebUI as follows:

1. Go to **Network | Routing | Virtual Routers** and click the number under the **Route Map** column for the VR you want to configure the route map in.
2. In the upper right-hand corner, click the **New** button.
3. Define a **Name** for the route map.
4. Just like an access list, you must define a **Sequence Number** for each statement in the route map.
5. Select what properties of the route you would like to use as match criteria, as well as the values to match with. In this example, use Access List 1 (which was created in an earlier section) to match routes. You must check the **Access List** checkbox under the Match Properties, and then highlight **1** for access list #1.
6. Change the metric of the route to 150 for this example. Enable the **Metric** option under **Set Properties**, and enter the metric into the field beside it.
7. Click **OK**.

For this example, we are using the following settings:

Virtual Router VR-1025

Map Name Route Map 1

Sequence Number 1

Action Permit

Access List Enabled

Access List (to Match) 1

Metric Enabled

Metric (to Set) 150

To configure this example through the Juniper CLI, enter the following commands:

```
set vrouter "VR-1025"
set route-map name "Route Map 1" permit 1
set match ip 1
set metric 150
exit
exit
save
```

We will cover more route map usage later in this chapter, and offer other examples, all of which will help your practical understanding of this topic.

Route Redistribution

By default, routes learned in one routing protocol are not propagated into other routing protocols, even within the same VR. For instance, if you learned a route to 10.1.1.0/24 from RIP, it would not be advertised by OSPF. Sometimes you might want to change the default behavior and advertise some or all of the routes from one protocol to another. Say you have an old Unix system that only supports RIP, but your routing infrastructure uses OSPF. You could have all routes learned by OSPF advertised into RIP so your Unix system would know of these routes. In order to advertise routes learned from one protocol to another, you must create redistribution rules that define what protocol to advertise to/from. Additionally, you can use route maps to either restrict or alter route values when using redistribution. A common example of this is when you advertise from one protocol to another and the metric values are different. For instance, RIP uses hop counts as its metrics (how many routes the traffic passes through). OSPF uses a different cost metric, which if directly advertised, could be much larger than the RIP's 15-hop limit. So you could use a route map to change the metric value for the routes being advertised to RIP to something that will be useful. Of course, it doesn't really stop with just metrics. You can get pretty fancy with route maps to help alter the routes you are advertising. We will cover individual examples of how to use redistribution in later sections for each routing protocol.

NOTE

You can only redistribute routes between protocols on the same VR. However, if you have multiple VRs that act as routing peers, they can still exchange routes between themselves in the routing protocols. An example of this would be if you had an OSPF instance on one VR, and another OSPF instance on a second VR, and they exchanged routing information via OSPF with each other. Of course, route redistribution is different than route importing and exporting, which can be done between VRs.

Importing and Exporting Routes

When you have two different VRs on a firewall and you want to share routing knowledge from one VR to another, you can use route exporting. This allows you to specify what routes you would like to export, along with setting attributes for the routes. Importing routes is optional, and is used to compare routes to a route map before they are imported into a VR. If you do not specify an import rule, it will simply import all routes matched by

the export rule. Although exporting routes is similar to redistribution, it is different since redistribution can only be done within an AS. We will next cover the properties of exporting and importing routes, along with an example.

Export Properties

The following properties are configurable as part of route export from one VR to another.

- **Source Virtual Router** This is the VR the routes will be exported from.
- **Destination Virtual Router** This is the VR the routes will be sent to.
- **Route Map** This is the route map that will be applied to exported routes.
- **Protocol** This is the routing protocol that routes will be exported from on the source VR and sent to the destination VR.

Import Properties

The following properties are configurable when you are importing routes from one VR to another.

- **Source Virtual Router** This is the VR the routes will be exported from.
- **Destination Virtual Router** This is the VR the routes will be sent to.
- **Route Map** This is the route map that will be applied to exported routes.
- **Protocol** This is the routing protocol which routes will be exported from on the source VR and sent to the destination VR.

Configuring an Export and Import Rule

In this example, we will configure an export rule from the VR-1025 virtual route into the Untrust-VR. We will also create an import rule for routes imported on the Untrust-VR from VR-1025.

To configure this example on the Juniper WebUI:

1. Select **Network | Routing | Virtual Routers**.
2. Click the **number** in the **Export Rules** column that is in the row of the source VR.
3. Specify the **Destination Virtual Router**.
4. Specify a **Route Map** regarding which routes to match and which values to set for the route attributes.
5. Select the **Protocol** you would like to export from the drop-down menu.
6. Click **OK**.

7. Configure an import rule next by going to **Network | Routing | Virtual Routers** and clicking the **number** in the **Import Rules** column, and the VR that you would like to configure the import rule on.
8. Click the **New** button in the upper right-hand corner.
9. Select the **Source Virtual Router** from the drop-down menu.
10. Specify the **Route Map** from the drop-down menu to filter routes and set attributes.
11. Select the **Protocol** you would like to apply for the imported routes.
12. Click **OK**.

In this example, we are using the following settings:

Export Rule Configured on VR-1025

Source Virtual Router VR-1025

Destination Virtual Router Untrust-VR

Route Map Route Map 1

Protocol OSPF

Import Rule Configured on Untrust-VR

Source Virtual Router VR-1025

Destination Virtual Router Untrust-VR

Route Map Attribute-Set

Protocol OSPF

To configure this example through the Juniper CLI, enter the following commands:

```
set vrouter "VR-1025"
set export-to vrouter untrust-vr route-map "Route Map 1" protocol ospf
exit
set vrouter "Untrust-VR"
set import-from vrouter VR-1025 route-map "Attribute-Set" protocol ospf
exit
save
```

Static Routing

Any device that routes traffic to other networks must have knowledge of where the networks are, and what path it should use to get the traffic to the destination. Of course, there

are a couple of ways to go about spreading this knowledge to different devices on the network. One of the most common ways of setting up this functionality is to use static routing. Essentially, static routing allows you to manually define the path that a router should send traffic on to get to a destination. This is opposed to dynamic routing where routers exchange information about networks which they know how to reach in an effort to automatically propagate this information. In a network where there might only be one path out of the network, it might make more sense to use static routing, which can be much simpler to implement with less overhead on the network. In this section, we will cover how static routes are configured on the firewall, followed by examples to reinforce the concepts.

Using Static Routes on Juniper Firewalls

Many people familiar with static routing just think of it as a way to route traffic to a destination network. Traditionally, static routes are used to provide the device with routing knowledge for destination hosts, networks, or even act as a catchall default route. Static routes are thus placed into the routing table, and it's as simple as that. They also have some additional uses that the Juniper firewalls allow you to take advantage of, functions which aren't typically seen in other vendor products.

The first example would be to use a static route to route traffic between two different VRs. You can essentially point the traffic to another VR which will handle the routing decisions. When the firewall is in transparent mode, you must have a static route for the management of the device. The management interface will have an IP address attached to it, and you will probably need a route to be able to route outside the subnet of the management interface. Next, if you are using route-based VPNs where you create a virtual tunnel interface, you can use static routing to route traffic into the tunnel interface to reach a remote destination. For instance, if you have a remote network with a subnet address 172.31.1.0/24, and a tunnel interface Tunnel.1 on the firewall, you would route 172.31.1.0/24 to tunnel interface Tunnel.1 to reach the destination. Lastly, you can configure static routes for source-based routing, source interface–based routing, and multicast routing.

Destination-Based Static Routes

Destination-based static routes use the destination network to determine what path to take to get the traffic to the destination. They have the following properties:

- **Virtual Router** This is the VR the static route will be applied in.
- **IP Address/Netmask** The IP address and subnet mask you want to create a route for. This is known as the destination network.
- **Next Hop** You have two options for specifying the next hop for a static route. The first is Virtual Router. Use this option to specify what VR the traffic should be forwarded to for routing. The second option is Gateway. Once a selection is made, the firewall then forwards the traffic to the destination of this next hop which then continues to pass it along to the destination.

- **Interface** This is the interface to forward the traffic out of.
- **Gateway IP Address** This is the next hop's IP address to forward the traffic to.
- **Permanent** This allows you to keep a route in the routing table even if it is not active.
- **Tag** This allows you to set the route tag.
- **Metric** This is the metric for this route.
- **Preference** This is the routing preference for this route. This is used to determine whether the route should be preferred over the same route announced by another routing protocol.

Configuring Destination-Based Static Routes on the Firewall

We will begin our discussion of static routes with an example of implementing three different destination-based routes. The first route will pass traffic out an interface, the next will route the traffic to a next hop gateway, and finally, the third route will point to the Untrust-VR.

To configure this example through the WebUI:

1. Open **Network | Routing | Destination**.
2. In the upper right-hand corner, select the appropriate **VR** from the drop-down menu, and then click **Next**.
3. Define the route you would like to provide a static destination-based entry for by filling in the **IP Address/Netmask** fields.
4. Under **Next Hop**, select **Gateway**.
5. Enter the **Next Hop** of the route in the **Gateway IP Address** field. Remember that this is the next logical hop in the route.
6. If you would like to alter the **Metric** or the **Route Preference**, you may do so by entering the appropriate values in those fields.
7. The route can also be made **Permanent** so it does not leave the routing table even if it becomes inactive. You can also define a route **Tag** in the respective field.
8. Click **OK**.
9. Now we will create a second route that will just pass traffic out an interface.
10. Create the route the same way you did with the previous route.
11. Define the **IP Address** and **Netmask** for the route you want to define an entry for.
12. Select the **Gateway** option for the **Next Hop**.
13. Choose the **Interface** you wish to forward the traffic out of for this route.

14. You can define the additional routing options such as **metric**, **preference**, and **tag** in the respective fields.
15. Click **OK**.
16. Define a route where we forward traffic to another VR.
17. Create the route in the appropriate VR the same way you did in previous examples.
18. Define the route you want to create a static entry for in the **IP Address/ Netmask** fields.
19. Make sure the **Virtual Router** option is selected for the **Next Hop**.
20. From the drop-down menu on the right, select the **Virtual Router** you would like to forward this traffic to.
21. Define the appropriate route options, and then click **OK**.

In this example, we are using the following settings:

Virtual Router VR-1025

Route #1

IP Address/Netmask 10.1.1.0/24

Next Hop Gateway

Gateway IP Address 192.168.45.254

Route #2

IP Address/Netmask 10.1.2.0/24

Next Hop Gateway

Gateway IP Address 192.168.45.254

Route #2

IP Address/Netmask 10.1.2.0/24

Next Hop Gateway

Interface Ethernet1

Route #3

IP Address/Netmask 10.1.3.0/24

Next Hop Virtual Router

Virtual Router Trust-VR (which VR to forward traffic to.)

Metric 50

Preference 30

To configure this example in the CLI:

```
set vrouter "VR-1025" route 10.1.1.0/24 interface null gateway 192.168.45.254
preference 20
set vrouter "VR-1025" route 10.1.2.0/24 interface ethernet1 preference 20
set vrouter "VR-1025" route 10.1.3.0/24 vrouter "trust-vr" preference 30 metric
50
save
```

Source-Based Static Routes

As described earlier in this chapter, source-based static routes are routes used for determining where the traffic should be routed, given the packet's source IP address. These routes have the following properties:

- **Virtual Router** This is the VR the static route will be applied in.
- **IP Address/Netmask** This is the IP address and subnet mask you want to create a route for. It is the source network the traffic came from.
- **Next Hop** You have two options for specifying the next hop for a static route. The first is Virtual Router. Use this option to specify what VR the traffic should be forwarded to for routing. The second option is Gateway. Once a selection is made, the firewall then forwards the traffic to the destination of this next hop which then continues to pass it along to the destination.
  - **Interface** This is the interface to forward the traffic out of.
  - **Gateway IP Address** This is the next hop's IP address to forward the traffic to.
  - **Permanent** This allows you to keep a route in the routing table even if it is not active.
  - **Tag** This allows you to set the route tag.
  - **Metric** This is the metric for this route.
  - **Preference** This is the routing preference for this route. This is used to determine whether the route should be preferred over the same route announced by another routing protocol.

Configuring Source-Based Static Routes on the Firewall

In this example, we will configure a source-based static route. Remember that before you can perform source-based routing on your firewall, you must enable it in the VR you would like to perform the source-based routing in. See the "Source-Based Routing Example" section earlier in the chapter for a refresher in enabling this feature.

To configure this example through the WebUI:

1. Go to **Network | Routing | Source** to view the source-based routing table.
2. In the upper right-hand corner, select the appropriate **VR** in the drop-down menu, then click **New**.
3. The configuration for this route should look familiar if you've seen the Destination-Based Routing table. But there is a key difference. The route you enter in the **IP Address/Netmask** is the source IP address or subnet you would like to route for.
4. Select whether you would like to forward to a **VR** or **Gateway** for the next hop. In this example, we will use the **Gateway** and define a **Gateway IP Address**.
5. Click **OK**.

In this example, we are using the following values:

Virtual Router VR-1025

Route Type Source Based

IP Address/Netmask 10.1.50/24

Next Hop Gateway

Gateway IP Address 172.16.2.1

To configure this example on the CLI:

```
Set vrouter VR-1025 route source 10.1.5.0/24 interface null gateway 172.16.2.1
preference 20
save
```

Source Interface–Based Static Routes

Routing decisions that are made based upon what interface the traffic arrived on are called source interface–based routing. These routes can be generated by putting static routes into the source interface routing table. You must have SIBR enabled on the VR in order to process it. See the "Source Interface–Based Routing Example" for more information on enabling it. The following are the properties of the source interface–based routes.

- **Source Interface** This is the interface the traffic arrives on.
- **IP Address/Netmask** This is the IP address and subnet mask you want to create a route for.
- **Next Hop** You have two options for specifying the next hop for a static route. The first option is Virtual Router. Use this option to specify what VR the traffic should be forwarded to for routing. The second option is Gateway. Once a selection is made, the firewall then forwards the traffic to the destination of this next hop which then continues to pass it along to the destination.

- **Interface** This is the interface to forward the traffic out of.
- **Gateway IP Address** This is the next hop's IP address to forward the traffic to.
- **Permanent** This allows you to keep a route in the routing table even if it is not active.
- **Tag** This allows you to set the route tag.
- **Metric** This is the metric for this route.
- **Preference** This is the routing preference for this route. It's used to determine whether the route should be preferred over the same route announced by another routing protocol.

Configuring Source Interface–Based Static Routes

Now we will configure an example to complement our discussion of SBIR. This route will route traffic arriving on a certain interface and direct it through the configured route. For this example, we will pass the traffic to another VR, but just like the other examples, you could forward it out another interface, or to another hop. Please remember you must turn on source interface–based routing on the VR you are configuring this route on in order to make this functionality work.

To configure this example through the WebUI:

1. Select **Network | Routing | Source Interface**.
2. In the upper right-hand corner, select the **Interface** for SBIR, and click **New**.
3. Define the **IP Address** and **Netmask** for the source addresses of the route you want to match. If you want to match any traffic on this interface, define it as 0.0.0.0/0.
4. For our example, we will forward traffic to the **Untrust-VR** as our **Next Hop**.
5. You can define the appropriate route options just like in the other examples.
6. Click **OK**.

In this example, we are using the following properties:

Interface Ethernet2 (trust-vr)

Route Type Source Interface Based Route

IP Address/Netmask 0.0.0.0/0

Next Hop Virtual Router

Virtual Router Untrust-VR

To configure this example in the CLI:

```
set vrouter trust-vr route source in-interface ethernet2 0.0.0.0/0 vrouter
"untrust-vr" preference 20

save
```

Multicast Routing

Although multicast has not really been as widely implemented as many first thought, it is slowly showing its qualities to the network world. Of course, multicast has had entire books written about it, so it will be out of the scope of this book to cover every detail, but we would like to show you how to create multicast static routes in the Multicast Routing table. We will begin with an explanation of the properties of multicast routes:

- **Source IP** This is the source IP address of the multicast host which is transmitting the traffic to traffic via multicast.
- **MGroup** This is the multicast IP group address that is being sent.
- **Incoming Interface** Defines what interface the traffic will be arriving on. This may be important since, potentially, multicast could arrive on multiple interfaces.
- **Outgoing Interface** Defines what interface the multicast traffic should be forwarded out of.
- **Translated MGroup** You can translate the multicast group address if you so wish with this option.

Configuring Static Multicast Routes on the Juniper Firewall

In this example, we will configure a static multicast route on the Juniper firewall. Of course, the rest of your network must be configured to support multicast routing for this to be truly effective.

To configure this example through the WebUI:

1. Go to **Network | Routing | MCast Routing**.
2. In the upper right-hand corner, select the appropriate VR to configure this route on, and then click the **New** button.
3. Define the **Source IP** of the host which is sending the multicast traffic.
4. Define the **MGroup**, also known as the multicast group address. This is the multicast destination address, such as 224.0.0.18.
5. You must also define what **Interface** the multicast traffic should arrive on at the firewall. This is known as the **Incoming Interface**, and it is important to define this since the traffic could arrive on more than one interface.
6. The **Outgoing Interface** must be defined from the drop-down menu. This is the interface the traffic will be forwarded out of.

7. Optionally, you can define the **Translated MGroup** if you need to perform NAT on the multicast traffic.
8. Click **OK**.

NOTE

The Incoming Interface you define must be part of the VR you defined this route in, or else you will be given an error.

In this example, we used the following configuration:

Virtual Router Trust-VR

Source IP 10.1.7.1

MGroup 224.0.0.18

Incoming Interface Ethernet1

Outgoing Interface Ethernet2

To configure this example via the CLI:

```
set vrouter trust-vr mroute mgroup 224.0.0.18 source 10.1.7.1 iif ethernet1 oif
ethernet2
```

save

Routing Information Protocol

Although static routes may work just fine for some networks, they do have their obvious drawbacks. In a large network with plenty of routes and frequent changes, static routing would be a nightmare to manage. Static routes also make it more burdensome to configure backup routes in case a link goes down. Network engineers quickly saw the drawbacks of this solution early on in the dawn of computer networking and scrambled to create a solution to alleviate the problem. One of the earliest routing protocols was the distance vector routing protocol RIP. Routers learn other routes from other routing peers, which share the state of the links with RIP. It has a very simple routing cost algorithm…one hop per router that the traffic has to pass through. It does not take into account anything about the link speed, or custom metrics. Furthermore, many devices (including mainframes) have supported this protocol for a very long time, so it might make sense to implement it in your network.

Notes from the Underground…

RIP Version 1

On the original RIP specification there was no mechanism to authenticate the routing updates you received through RIP. That meant a hacker could plug in a rogue router, and begin advertising RIP updates into your routing domain. Hackers could do anything from causing a Denial-of-Service attack (by misrouting traffic) to routing traffic through their own machines to capture it and sniff the network. It is highly recommended you use RIP version 2 with MD5 password hashing.

RIP Overview

Before we get into the fine details of how to configure RIP on the Juniper firewalls, we would like to present you with a bit of a RIP overview of the concepts, components, and terminology. Of course, like other protocols this will not be the definitive RIP guide, but we hope it will jog your memory on the key fundamentals. Of course, if you're feeling confident, feel free to skip ahead to the good stuff.

RIP Concepts

As mentioned before, RIP is a distance vector routing protocol that communicates with other routing peers to ensure the routing infrastructure is dynamically updated. RIP is an internal routing protocol used by Autonomous Systems (AS). It is not used as an external gateway routing protocol such as BGP. RIP communicates with its neighbors directly to pass the routing updates. Each neighbor, which is a fellow RIP router, will tell the other directly connected neighbors what networks it can route to, and how many hops it will take to route to it. These updates are sent out every 30 seconds by default. Each router then takes the updates and calculates what the best path is to get to every destination network based upon the number of hops to get there. Of course, since these routers are sharing information, it needs to be accurate, and since the updates are passed from router to router, you can run into routing loop issues.

To help prevent this, RIP implements a hard limit of 15 hops as to how far away a host can be. Anything routed further than that is considered unreachable, and is known as *count to infinity*. A couple of other ways exist to help prevent routing loops. *Split Horizon* does not advertise routes back out the interface they are heard on. For instance, if a RIP peer advertises the route 172.16.1.0/24 to your router on your ethernet1 interface, your router will not

advertise that route back out interface ethernet1 to help ensure it does not cause a routing loop. Next, the router can set the value to a destination of 16, which is greater than the maximum hop limit. This is called *poison reverse*. Lastly, the router can use a combination of different *hold-down timers* to ensure they do not make routing changes too quickly.

The reason for this is that since the information is passed along and not given to everyone all at once, convergence can be slow, and you don't want to have routes flapping up and down. By implementing a timer, this helps ensure you avoid the issue more gracefully. Convergence can also be affected by the number of routes you have on your network, or by a particularly poor routing infrastructure. The larger and more complex the routing infrastructure, the longer the convergence will be for RIP. Of course, RIP is meant to be a less-complex routing protocol, so hopefully you aren't trying to overburden it in your network. If you need to use something with a bit more horsepower and scalability, I suggest OSPF.

Two versions of RIP exist: version 1 and version 2. The main difference between the two is that RIP version 2 allows you to route with subnet masks instead of the routes automatically assumed to be routed on the classful lines. For instance, in version one, you only send the routes themselves without any subnet mask. RIP version 1 would take a route like 10.1.1.0 and automatically summarize it with a subnet mask of 255.0.0.0 or a Class A network. If RIP received a route for 172.31.2.0, it would summarize it as a Class B network with a subnet mask of 255.255.0.0. Lastly, it would summarize a Class C network such as 192.168.2.0 with a subnet mask 255.255.255.0. So essentially, you have no control over what subnet masks you would like to use with RIP version 1. In RIP version 2, you are able to define the subnet mask in the routing updates so you can define the networks based upon their appropriate subnet mask rather than the old classful guidelines. Another big difference between RIP version 1 and version 2 is that you can use authentication to help protect your routing infrastructure. Of course, the support for these additional features has altered the RIP packet structure between version 1 and version 2.

RIP Properties in a VR

RIP instances can be configured within VRs on the Juniper firewalls. Each VR can support a single RIP instance. You can configure several options when using RIP to dynamically route traffic in your network—but not to worry, Juniper really makes it simple in their firewall. RIP options include the following:

- **Protocol RIP** Selecting this option will enable RIP on the VR you are configuring this on.
- **Version** You may define a particular instance to be version 1 or version 2. Version 2 is backwards-compatible with version 1 but not vice versa. You can also configure what version you will send and receive for your routing updates.
- **Reject Default Route Learned from RIP** This option does not allow a default route announced by another router to be placed into your routing table. This can be a useful option if you would like to lock down the default route for

that particular router, or know that you do not want RIP advertising the default route.

- **Allow Neighbors from Different Subnets** By default, you will only become a peer with routers that are directly connected in the same subnet. This option allows you to override that functionality and lets you peer with neighbors in different subnets.
- **Advertise Default Route** This option automatically advertises its default route in RIP. Two options are available. The *always* option always advertises the default route in the RIP instance. You can also set the *metric* for the default route when you advertise it.
- **Metric for Redistributed Routes** This is the value you will select as the metric for routes that you redistribute. Of course, you can override this with route maps, or not use this at all and only use route maps to set the metric.
- **Maximum Alternative Route** This allows you to keep more than one primary route in the routing table. By default, in the active routing table, RIP will only put one route in there. If that route becomes inactive, you can have RIP recalculate the network for the next best route. If you select this option, you can define how many additional routes to keep in the routing table for this destination.
- **Hold Down Timer** This is the value for how long the router should wait before making a change to a routing table when RIP detects that the route has a new higher metric. This is used to help prevent a count to infinity. By default, it is 90 seconds.
- **Retransmit Timer** This is the value in seconds that defines how often the VR should try to send responses for on-demand circuits.
- **Retransmit Count** The number of retransmission that should be sent before the route is considered in a POLL state.
- **Poll Interval** This is how often the VR should poll to see if the on-demand circuit is backed up. It is 180 seconds by default.
- **Poll Retry** The number of poll messages sent before the route is considered down.
- **Periodic Route Table Update** By default, RIP will send a complete routing update every 30 seconds, and when it detects a change. You can change this value.
- **Route Invalidation Interval** This is the value in seconds before a route is invalidated. This value begins when the RIP neighbor stops announcing the route, which then begins the time to put the route in an invalidated state. This is 180 seconds, by default.

- **Route Flushing Interval** This is the value in seconds before an invalid route is removed from the system. It is 120 seconds, by default.
- **Maximum Number of Packets per Update** This is the value of the maximum number of routing updates a VR will take for RIP in an update interval. This can be an important value to set if you are concerned about a Denial-of-Service attack against RIP.
- **Trusted Neighbors** This allows you to specify the IP addresses of routers you decide to trust as RIP neighbors. You can lock this down to only the neighbors you want to be a part of the routing domain.
- **Maximum Neighbors** This allows you to define the maximum number of RIP neighbors. It may be helpful against a Denial-of-Service attack.
- **Incoming Route Map Filter** This allows you to define the route map that incoming routes (those advertised by a RIP neighbor) will be compared to. Routes may be passed, rejected, or altered as configured in the route map.
- **Outgoing Rout Map Filter** This is the route map used to filter out routes before they get announced from the local RIP router to other peers. You can pass routes, reject them, or alter them as you wish with the route map configuration.

WARNING

You should not alter the default timer values for routing protocols, unless you have a specific reason, and you know what you are doing. Altering these values can either cause convergence problems, or overload your routers and your network with routing protocol traffic. Proceed with caution.

RIP Settings Per Interface

Once you have configured a RIP instance in the respective VR, you may then configure a few properties on an interface-by-interface basis. Those properties are covered in the following:

- **RIP Instance** This allows you to set this interface as an instance of the RIP protocol.
- **RIP Protocol** This lets you enable the RIP protocol on the interface.
- **Summarization** This allows you to summarize RIP routes announced out this interface. You must configure summarization in the VR.

- **Update Version** This is the version of RIP updates you will send, and also accept from other neighbors.
- **Sending v1** You will send only version 1 updates. **v2** You will send only version 2 updates. **v1/v2** You will send both v1 and v2 updates. **VR-Default** You will send whatever the VR is configured to send, instead of configuring it on the interface.
- **Receiving v1** You will receive only version 1 updates. **v2** You will receive only version 2 updates. **v1/v2** You will receive both v1 and v2 updates. **VR-Default** You will receive whatever the VR is configured to send, instead of configuring it on the interface.
- **Metric** This is the metric value on the interface.
- **Authentication** This is for RIP version 2, and allows you to either use no authentication or MD5 password hashing.
- **Password** Allows you to set a plaintext password.
- **Passive Mode** This interface won't announce any routes, but will listen for them and update their routing table accordingly.
- **Incoming Route Map Filter** This allows you to define a route map for routes received on this interface only.
- **Outgoing Route Map Filter** Allows you to configure a route map for outbound routes to be advertised to routing peers.
- **Split Horizon** This allows you to set the following split horizon settings:
  - **Enable Split Horizon without Poison Reverse** This prevents routes from being advertised back out the interfaces they were advertised to the firewall on, but doesn't perform the poison reverse on the route.
  - **Enable Split Horizon with Poison Reverse** This alters the routes that get advertised back out the interface where they were heard on. Such routes are advertised with a value of 16, which is seen as unreachable.
  - **Disabled** This disables poison reverse on this interface.
- **Static Neighbor IP** This is the IP address of the neighbor on this interface.

WARNING

Disabling Split Horizon is not recommended in most circumstances. Doing so could create routing loops in your network and cause major connectivity issues. Be sure that if you need to disable it, you know exactly what you're doing, and what the consequences could be.

Enabling RIP within a VR

In this example, we will configure a RIP instance within a virtual router. In the next example, we will configure RIP on an interface.

To configure this example via the Juniper WebUI:

1. Go to **Network | Routing | Virtual Routers.**
2. Click the **Edit** hyperlink next to the VR you would like to configure the RIP instance in.
3. Near the bottom of the screen, click the **Create RIP Instance**.
4. Check the **Protocol RIP Enable**.
5. Use **Version V2** for the RIP version.
6. Select the **Reject Default Route Learnt by RIP** box so you don't import a default route from RIP.
7. Uncheck **Allow Neighbors from a Different Subnet** and **Advertise Our Default Route.**
8. Leave all of the other predefined options at their default.
9. Click **OK**.

In this example, we are using the following values:

Virtual Router Trust-VR

Protocol RIP Enable

Version v2

Reject Default Route Enable

All other options Left at default

To configure this example in the CLI:

```
set vrouter "trust-vr"
set protocol rip
set enable
set reject-default-route
exit
exit
save
```

Configuring RIP on the Interface

In this example, we will configure RIP on the individual interfaces on the firewall. You must enable RIP on the individual interfaces of the firewall in order for it to participate with

other RIP routers. Additionally, you can configure some options on an interface-by-interface basis.

To configure this example via the Juniper WebUI:

1. Select the appropriate interface you would like to configure RIP on by going to **Network | Interfaces** and then clicking the **Edit** hyperlink next to the chosen interface.
2. At the top of the screen, click the **RIP** hyperlink at the top of the screen.
3. Configure RIP on the interface by checking the **RIP Instance** checkbox.
4. To enable RIP on this instance, check the **Protocol RIP Enable** checkbox.
5. You won't perform summarization in this example, but if you wanted to summarize the routes advertised, you would check the **Summarization** checkbox.
6. You can configure which RIP version you use to send and receive updates. Here, use the **VR-Default** version, which is configured as version 2. If you have another RIP router which only supports V1, you can configure the interface to send and receive both versions.
7. Leave the **Metric** for this interface as 1. You might want to change this if you have a link with particularly low bandwidth you want to account for.
8. To help secure the RIP infrastructure from unauthorized updates, use **MD5** authentication.
9. Define a **Key** which is the password to use between the routers. You must also specify the **Key ID** for this key. Lastly, define that this key is **Preferred**.
10. For this example, you don't want to use **Passive Mode** since you will want this router to actively participate in RIP routing.
11. If you want to filter what routes RIP will accept from routing updates, as well as what routes you will advertise to other routers, this can be configured by implementing **Incoming and Outgoing Route Filters**.
12. Configure your **Split Horizon** settings. For this example, use **Split Horizon with Poison Reverse**.
13. You also have the ability to define what router you will allow to be the **RIP neighbor** for this interface. For this example, MD5 authentication will suffice.
14. Click **OK**.

In our example, we are using the following settings:

Interface	**Ethernet1**
RIP Instance	**Enabled**
Protocol RIP	**Enabled**

Summarization	**Disabled**
Update Version Send	**VR-Default**
Update Version Receive	**VR-Default**
Metric	**1**
Authentication	**MD5**
MD5 Keys	**8JFoa+[]**
Key ID	**1**
Passive Mode	**Disabled**
Incoming Route Filter	**None**
Outgoing Route Filter	**None**
Split Horizon	**Enabled with Poison Reverse**
Static Neighbor IP	**Disabled**

To configure this option via the Juniper CLI:

```
set interface ethernet1 protocol rip
set interface ethernet1 protocol rip enable
set interface ethernet1 protocol rip split-horizon poison-reverse
set interface ethernet1 protocol rip authentication md5 "8JFoa+[]" key-id 1
set interface ethernet1 protocol rip authentication active-md5-key-id 1
save
```

Tools & Traps…

Attacking Routing Protocols

Several programs exist that are created for educational or malicious purposes to glean information from routing protocols. Cain and Abel allows you to sniff the network for routing protocol messages, which it will automatically parse to display the routers participating. It can also display the authentication key (if any) and has facilities to crack weak MD5 passwords. Other applications such as the Unix/Linux routing software like Zebra allow you to configure routers on PC hardware. It is important to remember that a router doesn't have to be a specialized piece of hardware, and that such routing software could be used for malicious purposes.

Controlling What Routes RIP Learns and Advertises

Sometimes you may want to restrict what routes RIP will either learn or advertise in your routing architecture. This is actually a simple task to perform. You essentially need to create a route map specifying what routes you would like to match, and then define the route map as either an incoming route map or an outgoing route map for the RIP protocol.

To configure this example via the WebUI:

1. Go to **Network | Routing | Virtual Routers** and click the number hyperlink in the **Access List** column and row for your VR.
2. Click **New** in the upper right-hand corner.
3. Define a **Name**, a **Sequence No**, an **IP Address / Netmask**, and an **Action**. We will do this four times in this example, once for each of the three private address ranges which we will deny, and then a fourth time to allow any other range.
4. Create the route map next, which references the access list we just created. Go back to the **Network | Routing | Virtual Routers** page and click the number under the Route Map column, as well as on the row of your VR.
5. Click the **New** button in the upper right-hand corner of the screen.
6. Just like other examples, we will define a **Name**, **Sequence Number**, and **Action**. We will match the routes based upon an **Access-List** and the **Access List ID**.
7. Apply the route map to the RIP instance by going to **Network | Routing | Virtual Routers**. Then, click **Edit** next to the VR you would like to edit.
8. Scroll to the bottom of the screen and click the **Edit RIP Instance** hyperlink.
9. Towards the bottom of the screen, you can select the route maps you would like applied for incoming routes learned from other routers, as well as outgoing routes to be advertised to other routers. These route maps can be applied in the **Incoming Route Map Filter** and the **Outgoing Route Map Filter**, respectively.
10. Click **OK**.

In this example, we configured the following properties:

Virtual Router	**VR-1025**
Access-List	**60**
Sequence No.	**1**
IP / Netmask	**192.168.0.0/16**
Action	**Deny**

Sequence No.	**2**
IP / Netmask	**172.16.0.0/12**
Action	**Deny**
Sequence No.	**3**
IP / Netmask	**10.0.0.0/8**
Action	**Deny**
Sequence No.	**4**
IP / Netmask	**0.0.0.0/0**
Action	**Permit**
Route Map	**RIP-Match**
Sequence No.	**1**
Action	**Permit**
Access-List	**Checked**
Access-List (ID Chosen)	**60**
Incoming Route Map Filter	**RIP-Match**
Outgoing Route Map Filter	**RIP Match**

To configure this example in the Juniper CLI:

```
set vrouter "VR-1025"
set access-list 60
set access-list 60 deny ip 192.168.0.0/16 1
set access-list 60 deny ip 172.16.0.0/12 2
set access-list 60 deny ip 10.0.0.0/8 3
set access-list 60 permit ip 0.0.0.0/0 4
exit
set route-map name "RIP-Match" permit 1
set match ip 60
exit
set protocol rip
set route-map "RIP-Match" in
set route-map "RIP-Match" out
exit
exit
save
```

NOTE

You can also enable Router filtering on an interface-by-interface basis under the RIP configuration for the interface to which you would like to apply it.

RIP Informational Commands

Once you have RIP configured, you will probably want to verify that it is functioning as you expect. Of course, if there are problems, you will probably want to troubleshoot them. Juniper offers you everything you need to view and verify the configuration and operation of your RIP instance. We will cover commands to retrieve information about the configuration, interface settings, established neighbors, route database, and routes in the routing table.

Summarizing RIP Information

When you need to get a summary of the overall configuration and state of the RIP protocol on a particular router, you can issue the *get vrouter <vroutername> protocol rip* command. This will give you information about everything from what interfaces are participating in RIP, to the timing values and routing filters. See the following example for a screenshot of the output captured from running this command on a RIP instance of the firewall.

```
ns5gt-adsl-wlan-> get vrouter trust-vr protocol rip
VR: trust-vr
------------------------------------------------------------------------------
State: enabled
Version: 2
Default metric for routes redistributed into RIP: 10
Maximum neighbors per interface: 16
Not validating neighbor in same subnet: disabled
Next RIP update scheduled after: 23 sec
Maximum number of Alternate routes per prefix: 0
Advertising default route: disabled
Default routes learnt by RIP will not be accepted
Incoming routes filter and offset-metric: not configured
Outgoing routes filter and offset-metric: not configured
Update packet threshold is not configured
Total number of RIP interfaces created on vr(trust-vr): 2

Update| Invalid|   Flush| DC Retransmit| DC Poll| Hold Down (Timers in seconds)
-------------------------------------------------------------
    30|      180|         120|             5|               180|         90
```

```
Flags: Split Horizon - S, Split Horizon with Poison Reverse - P, Passive - I
       Demand Circuit - D
Interface   IP-Prefix           Admin      State       Flags   NbrCnt Metric Ver-
Rx/Tx
-------------------------------------------------------------------------------
eth1        192.168.1.1/24    enabled    enabled    P            0          1
2/2
eth2        192.168.45.1/24  enabled    disabled  P            0          1
2/2
```

Retrieving the RIP Config

Juniper has provided you with a command that allows you to grab all of the related RIP information from the firewall: *get vrouter <vroutername> protocol rip config*. This can save you from having to sift through the entire configuration in the firewall. The output from running this command on the firewall is shown next:

```
ns5gt-adsl-wlan-> get vrouter trust-vr protocol rip config
VR: trust-vr
-------------------------------------------------------------------------------
set protocol rip
set enable
set reject-default-route
exit
set interface ethernet1 protocol rip
set interface ethernet1 protocol rip enable
set interface ethernet1 protocol rip split-horizon poison-reverse
set interface ethernet1 protocol rip authentication md5 "8JFoa+[]" key-id 1
set interface ethernet1 protocol rip authentication active-md5-key-id 1
set interface ethernet2 protocol rip
set interface ethernet2 protocol rip enable
set interface ethernet2 protocol rip split-horizon poison-reverse
set interface ethernet2 protocol rip authentication md5 "8JFoa+[]" key-id 1
set interface ethernet2 protocol rip authentication active-md5-key-id 1
```

Displaying the RIP Interface State

Sometimes you just want to view what state the RIP interface is in on the firewall. This provides you with a concise view of the RIP configuration of the interface, including the interface name, IP address and netmask, the RIP state, the number of neighbors, the metric, and what send and receive version you have configured on that interface. The result from running this command on our firewall was the following:

```
ns5gt-adsl-wlan-> get vrouter trust-vr protocol rip interface
VR: trust-vr
------------------------------------------------------------------------------
Flags: Split Horizon - S, Split Horizon with Poison Reverse - P, Passive - I
       Demand Circuit - D
Interface      IP-Prefix          Admin      State    Flags   NbrCnt Metric Ver-
Rx/Tx
------------------------------------------------------------------------------
eth1        192.168.1.1/24    enabled    enabled  P            0            1
2/2
eth2        192.168.45.1/24  enabled    disabled P            0            1
2/2
```

Displaying the RIP Neighbors

You can use the *get vrouter <vrouternam> protocol rip neighbor* command to display the neighbors associated with each RIP interface on your firewall. This will also display some statistics regarding the information advertised.

```
ns5gt-adsl-wlan-> get vrouter VR-1025 protocol rip neighbors
VR: VR-1025
------------------------------------------------------------------------------
Flags : Static - S, Demand Circuit - T, NHTB - N, Down - D, Up - U, Poll - P,
        Demand Circuit Init - I
------------------------------------------------------------------------------
IpAddress        Version      Age     Expires BadPackets  BadRoutes  Flags
------------------------------------------------------------------------------
192.168.45.254 v2           -       -                   0          0  SU
```

Showing the RIP Routes and RIP Database.

When RIP just isn't working as you expect, or you want to verify what routes RIP knows, Juniper has a couple of useful commands. The first is the *get vrouter <vrouternam> route protocol rip* command, which shows the routes that RIP is advertising to its neighbors. Of course, you could issue the *get route* or *get vrouter <vrouternam> route* commands to get all the routes on your firewall or all of the routes in a particular VR, but if you just want to see the RIP routes, the *get vrouter <vrouternam> route protocol* command will do the trick.

Of course, these are the best routes that RIP knows, but most likely, RIP will have knowledge of other routes to reach a particular destination as well. If you would like to see the database which the firewall uses to build the routes that RIP will advertise, issue the *get vrouter <vrouternam> protocol rip database* command.

```
ns5gt-adsl-wlan-> get vrouter VR-1025 protocol rip database
VR: VR-1025
```

```
------------------------------------------------------------------------------
Total database entry: 0
Flags: Added in Multipath - M, RIP - R, Redistributed - I,
       Default (advertised) - D, Permanent - P, Summary - S,
       Unreachable - U, Hold - H
DBID   Prefix               Nexthop        Ifp     Cost Flags   Source
1       192.168.100.0/24      192.168.45.254   eth1     3     R  192.168.45.254
2       192.168.101.0/24      192.168.45.254   eth1     4     R  192.168.45.254
3       192.168.102.0/24      192.168.45.254   eth1     5     R  192.168.45.254
------------------------------------------------------------------------------
```

Juniper Support for RIPng

Although it is out of the scope of this book, we wanted to mention that Juniper supports the IPv6 version of RIP, known as RIPng. This protocol is very similar to RIPv2, with the main differences revolving around the fact that it is routing IPv6 traffic, instead of IPv4. We encourage you to visit Juniper's support site at www.juniper.net/support/ if you would like more information on implementing this protocol in the firewalls.

Open Shortest Path First

Although RIP was a big step up from having to program routes in manually, it did not scale very well in medium- and large-sized networks. Convergence and routing loops were also a factor with RIP. For those reasons, OSPF has probably become the most widely implemented routing protocol for internal networks. There are other proprietary routing protocols which have also been popular, but they do not offer the same possibilities to integrate with such a wide range of platforms.

OSPF is a link state routing protocol, and works very differently than RIP. Every OSPF router announces the routes that it connects to, and all other OSPF routers are given the same information via multicast. This allows every router to have a picture of the whole network, instead of just knowing about your neighbor. The bird's eye view that OSPF has on a routing domain helps protect it from being subjected to routing loops like its distance vector cousin. That isn't to say you can't mess up an OSPF implementation, just that there are certain safeguards built in to the protocol itself.

OSPF also offers more segmentation and scalability than other distance vector routing protocols. In this section, we will begin by explaining the key concepts and terminology of OSPF. We will then move onto the implementation of OSPF in the firewalls, and include some examples.

Concepts and Terminology

OSPF is not exactly a protocol that you simply enable and then forget about. Truth be told, it isn't extremely difficult to work with, but it certainly helps to have a good working knowledge of routing in general. OSPF has become so popular because it is robust, scalable, flexible, and is an open standard. OSPF is based upon the *Shortest Path First* algorithm, sometimes call *Dijkstra's Algorithm*. Shortest Path First essentially takes a map of the whole network (which each router helps to build) and discovers the shortest path for every destination. Let's start with an overview of the concepts that are important to understanding and implementing OSPF.

Autonomous System

A network that is controlled by a single organization or entity is called an Autonomous System, or AS for short. When referring to an internal network composed of multiple areas, we refer to them collectively as being part of the same AS. This term is also used in BGP with a similar meaning, which will be discussed later in this chapter.

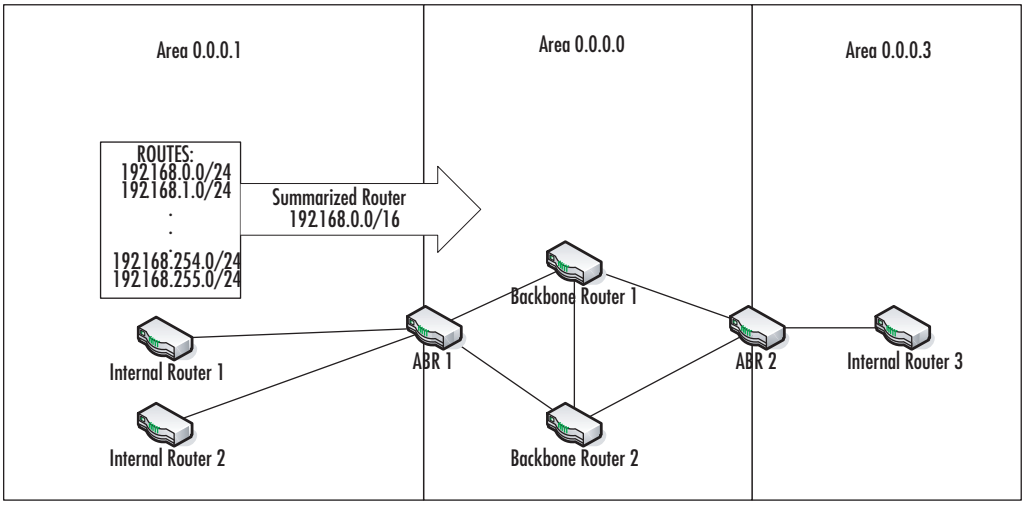
Areas

Perhaps the true power behind OSPF is its ability to simplify some of the routing infrastructure by segmenting it. This technique utilizes a concept called *areas* which allows you to separate routing information into separate groups. You could liken areas to routing, the same way subnetting is connected to IP networks. Much like how subnetting contains broadcast traffic, areas contain routes. As you may know, the more hosts you put on a subnet, the more of a performance impact is felt (especially when you include several hundred, or even thousands, of hosts). The same is true for routing. The more routes a router must account for and calculate in the routing table, the more load you will put on that system. Of course, the bigger issue is how long it might take all the routers to properly calculate the accurate picture of the network. This is known as *convergence*, and it is a state you definitely want your network to be in!

Essentially, areas contain routes within a group that you configure. You can create many areas, but your network must contain a backbone which all other areas connect to. This is known as the backbone area with the number 0 (or 0.0.0.0). Each area is labeled with a number to distinguish it from other areas. An area will contain a group of routers that share routing information with each other. There is also a router which sits between an area, and the backbone. This router is known as the Area Border Router (ABR) and is covered in depth next. Suffice to say that it is the job of the ABR to announce routes from its area into the backbone, and vice versa. If your network is properly architected, you can summarize your routes and simplify the number of routes that your routers must know about. For example, say you have 256 networks in an area 192.168.0.0/24 through 192.168.255.0/24. If you didn't have areas, then every single router outside of that area would have to know about all of those routes, including every change that might occur. With areas, you could

summarize those routes on the ABR to one single route 192.168.0.0/16. The summarized route can be passed on from the backbone area, into other areas attached to the backbone, as a single route. This effect can dramatically improve the speed of convergence, and simplify your routing infrastructure. See Figure 7.4 for a visual representation of this discussion.

Figure 7.4 OSPF Areas and Summarization



Virtual Links

There is one exception to the rule that every area must be directly connected to the backbone via an ABR. The ability to have an area not directly connected to the backbone is called creating a *Virtual Link*. Basically, you must configure both the areas appropriately to simulate a connection directly to the backbone, when in fact the remote area must first connect to another area, which then connects to the backbone. Using virtual links is generally considered a bad practice, but it is part of the OSPF standard and is thus supported by the Juniper firewalls.

Types of OSPF Areas

Building on our explanation of OSPF areas, there are actually a few different types of areas that serve different purposes. We will cover each of them in the following:

- **Backbone Area** This is the central area in your OSPF routing domain which all areas must connect to. This area is labeled area 0 (or the longhand version 0.0.0.0). When traffic needs to pass from one area to another, it must traverse the backbone. Routes are advertised to all other areas from the backbone.

- **Stub Area** This is an area which connects to the backbone. Traffic does not pass through the area, but rather originates from, or is destined to, this area. It must have a unique label value within the routing domain and cannot be labeled area 0. Routes learned here are advertised into the backbone, as well as other areas (often in a summarized form).
- **Totally Stubby Area** This is similar to a Stub Area, except it does not accept any summarized routes to be injected into it.
- **Not So Stubby Area** This is an area which connects to the backbone, but it might also connect to a remote network. In other words, traffic *may* traverse through this area. This area may advertise routes learned to other areas. Sometimes a Not So Stubby Area (NSSA) will contain VPN connections to other trading partners, or similar scenarios.

Routers within Each Area

OSPF has different terminology for routers, depending on where in the routing domain they exist. This information is useful for determining what specific tasks the router must perform to maintain its role in the routing domain.

- **Backbone Router** This router is located within the backbone area. It is responsible for routing traffic across the backbone, but it does not connect to any other areas.
- **Internal Router** This is a router located within an area (not within the Backbone). It may be responsible for announcing routes to other routing peers, as well as routing traffic within the area.
- **Area Border Router** We mentioned this router before. Basically, an ABR sits between the backbone, and another area. It is responsible for routing traffic between the backbone and the area it touches, as well as summarizing routes between both areas. An ABR is the router which connects the backbone to the stub area.
- **Autonomous System Boundary Router** This is a router which sits on the edge of an area, and advertises external routing information into the routing domain. It might be a router which is announcing routes from another protocol, or it might be announcing routes for a remote location which it sits on the edge of. This router is a part of an NSSA.
- **Designated Router** This is a router within a particular area that's responsible for propagating the routing information. It is a special router which is elected based upon a priority value. It helps to reduce the amount of traffic which is sent because the routers don't need to update each other, but rather, update the Designated Router (DR) which helps perform this task. DRs are used on broad-

cast links primarily. There are ways to manually configure this process on non-broadcast multiple access networks. They are never used on point-to-point links, however, since there is only one path upon which the data can travel.

- **Backup Designated Router** This is a router with the second highest priority that listens to the routing updates and keeps an up-to-date topology table in case the DR should fail. If that occurs, the BDR will be promoted to the DR.

OSPF Neighbor Relationships

Link state routing protocols must have a mechanism to cooperatively propagate routing information across the routing domain. OSPF tackles this task by forcing the routers to form relationships between each other. Depending on what type of router (Backbone, Internal, ABR, ASBR), OSPF may form different types of relationships to communicate the information. We will begin our discussion of neighbor relationships with an introduction to the different types of networks, establishing neighbor relationships, and propagating the routing information.

OSPF Network Types

Depending on what type of network you run OSPF on, OSPF will operate a little differently. The three types we will cover here are as follows:

- **Broadcast** This is a type of access where, within a network, a machine may broadcast so that every machine may receive the traffic. The most common example of this is Ethernet. Broadcast networks are typically used on a LAN.
- **Point-To-Point** Networks which connect one machine directly to another are known as point-to-point networks in OSPF. A common example of this would be a WAN link, which directly connects two routers. It might use a layer 2 protocol such as PPP or HDLC.
- **Non-Broadcast-Multiple-Access** This is a type of network that does not support network broadcasting, but does allow multiple devices to connect to it. An example of this would be ATM or Frame-Relay.

Establishing an OSPF Relationship

OSPF forms neighbor relationships by announcing its presence on the network. Since OSPF is its own protocol, it has a format for the packets which it sends out onto the network. The *Hello* packet is exchanged between routers via multicast, on the address 224.0.0.5. This packet provides information to a potential neighbor, such as its area number, priority, and IP address. The Hello packet is also used to determine which router is going to be elected the DR and BDR based upon its priority. Lastly, the Hello packet helps establish that a router is up and running. When no response is heard from a router after a certain time period, it is

considered down. If the DR goes down, the BDR will take its place. If both go down, an entire new election will need to take place to determine the new DR and BDR (similar to when the network is first established). Since these packets contain the area that the router is in, an OSPF router from one area will ignore a router from another. Of course, an ABR will send different hello packets out the respective interfaces for the area they are in.

Link State Advertisements

Link State Advertisements (LSAs) are messages communicated via multicast to other routers in the OSPF domain. They are sent from internal routers to the DR/BDR routers to announce changes. This communication occurs on multicast address 224.0.0.6. The DR will announce changes to the other routers via multicast address 224.0.0.5. Several different types of LSAs exist in OSPF. We will focus on 1 thru 5 and 7, which are the ones you have to be concerned with for the functionality of OSPF on the firewall.

- **Router LSA (1)** This LSA is sent from an internal router to the DR/BDR routers to announce a change in the network. Specifically, they define the state of their interfaces, as well as associated costs.
- **Network LSA (2)** The DR will send this LSA out to the other routers in its area to announce the topology information based upon what it has gathered from other LSA types.
- **Summary LSA (3)** This LSA is sent between areas to announce routes from one area to another. By summarizing your network, you take advantage of using this LSA, since you reduce the number of routes you have to announce. These are sent by the ABR.
- **ASBR Summary LSA (4)** This is essentially a re-advertised version of the LSA 5 packet. Originally, the ASBR advertises the type 5 packet, but because some of the next hop information may not be known by remote networks, this packet is translated by the ABR to type 4, which is then passed on to other areas.
- **External LSA (5)** These are sent from the ASBR and are often the result of some sort of routing redistribution process from one routing protocol to another. They are sent to all areas, but may be filtered on areas such as stub and totally stubby areas.
- **NSSA LSA (7)** These are sent from routers in an NSSA to ABRs to redistribute into the OSPF area. The ABR will translate these into type-5 OSPF packets.

Configuring OSPF

Before we dive into some real examples, we would like to cover the different properties of OSPF within the Juniper firewall. OSPF requires that you configure it both on the VR, but also on the individual interfaces that will be participating in the OSPF protocol.

OSPF Properties within a VR

Before you can begin using OSPF, you must configure the appropriate VR to enable the OSPF protocol. We will cover the configuration options to do so in this section.

- **Advertising Default Route** This allows you to either enable or disable the advertising of the default route from this router.
  - **Metric** You can define the metric for this route. The values can be between 1 and 65525.
  - **Metric Type** You can define if this is an OSPF Type 1 or Type 2 metric.
  - **Always** This means you should always advertise the default route.
- **Automatically Generate Virtual Links** This option allows the firewall to automatically generate OSPF virtual links if you have an area that does not directly connect to the backbone.
- **Do Not Add Default-Route Learned by OSPF** If you enable this option, OSPF will not import any default route it learns into the routing table.
- **Prevent Hello Packet Flooding Attack** A Hello packet flooding attack can come either as a malicious DoS attempt, or an improperly configured router. In either case, an exceptional number of hello packets are set out. Each of these packets forces the router to perform some processing, so if too many are sent, it can cause issues.
  - **Off** Leave this off on this instance.
  - **On** Turns on the protections
  - **Maximum Hello Packet** This is the maximum number of Hello packets that can be received per second.
- **Prevent LSA Flooding Attack** An LSA flooding attack is similar to the Hello flooding attack except that it deals with LSA packets. Just like the Hello packets, it can cause issues on the router if it is overburdened with processing the updates.
  - **Off** Leave this option off.
  - **On** Turns on the protection.
- **Maximum LSA Threshold Time** Define the time period for each cycle in seconds.

- **Maximum LSAs** This is the number of LSA updates. Essentially, you define the threshold (explained earlier) and this is the number of LSAs that can appear within that threshold.
- **RFC-1583 Compatible** This option enables the OSPF instance to be compatible with the RFC-1583 standard.
- **OSPF Enabled** Enables OSPF for this instance.

Configuring OSPF in a VR

We will begin our OSPF examples by creating an OSPF instance with a VR. We will use this example to build upon this example to extend the OSPF functionality throughout this chapter.

To configure this example through the Juniper WebUI:

1. Select **Network | Routing | Virtual Routers**.
2. Click the **Edit** hyperlink next to the VR you want to create an OSPF instance in.
3. Scroll down to the bottom of the screen and select **Create OSPF Instance**.
4. If you would like to **Advertise the Default Route**, you may select it.
5. In our example, we will not **Automatically Generate Virtual Links**, or **Add the Default Route Learnt by OSPF**
6. You can provide additional DoS protection by enabling the **Prevent Hello Flooding Attack** and **Prevent LSA Attack** options.
7. If you need to be compatible with **RFC-1583**, you can enable that option.
8. **Enable OSPF** to actually enable this instance on the firewall.
9. Click **OK**.

In this example, we are using the following settings:

Virtual Router	**Trust-VR**
Advertise Default Route	**No**
Automatically Generate Virtual Links	**No**
Do Not Add Default Route Learnt by OSPF	**Enabled**
Prevent Hello Flooding Attack	**On**
Max Hello Packet	**20**
Prevent LSA Attack	**On**
LSA Packet Threshold Time	**60**
Maximum LSAs	**180**

RFC-1583 Compatible	**No**
OSPF Enabled	**Yes**

To configure this example via the CLI:

```
set vrouter "trust-vr" protocol ospf enable
set vrouter "trust-vr" protocol hello-threshold 20
set vrouter "trust-vr" protocol lsa-threshold 60 180
set vrouter "trust-vr" protocol reject-default-route
save
```

NOTE

If you create an OSPF instance on a custom virtual router, you must set a *Router-ID* before you can configure the instance. This can be done with the *set vrouter <vroutername> router-id <x.x.x.x>* command, where the vroutername is the name of your VR (such as VR-1025) and the router-id is an IP address for the router-id to be uniquely identified by.

Area Properties

Each OSPF instance can contain multiple areas, each of which you must configure according to your topology. You must specify at least one area that your router belongs to. The following properties can be used when setting up an OSPF Area:

- **Area ID** This is the area number you would like to configure. It will take the dotted decimal for x.x.x.x, where each x is a value from 0 thru 255. Note that this is not an IP address, but just a unique way of identifying the instance.
  - **Type** You must configure what type of area this is.
  - **Normal** This would most likely be the backbone area.
  - **Stub** This is a stub area.
  - **NSSA** This would be a Not So Stubby Area which will advertise external routes into OSPF.

Configurable Properties within an Area

Each area can have the following specific properties configured within them:

- **Area Range** You can configure the summary address for your area here with the following properties:

- **IP Address/Netmask** This would be the summary route you would advertise to other routers from this area. For instance, if you summarized 256 routes, 192.168.0.0/24 throug 192.168.255.0/24, then you would configure that here.
- **Type** This is a type of route you can configure:
  - **Advertise** This route will be advertised out this area.
  - **No Advertise** You will not advertise this route out of this area (withhold the route).
- **Bound Interfaces** You must select what interfaces you would like to enable in this area. Remember, you must always define what area an interface is in. This helps the router determine which OSPF packets it should accept, as well as what mode it should be in.

Configuring an OSPF Area and Creating a Summary Route

OSPF areas must be configured within a VR and are needed for OSPF functionality. In this example, we will create and configure a new stub area within the Trust-VR. The backbone area is automatically configured on the firewall. We will assume you have already created the OSPF instance in the Trust-VR as described in the last example.

To configure this example in the WebUI:

1. Go to **Network | Routing | Virtual Routers**.
2. Click the **Edit** hyperlink next to the VR where you want to add an OSPF area to the OSPF instance.
3. In the VR screen, scroll down to the bottom and click the **Edit OSPF Instance** hyperlink.
4. Now that you are in the OSPF configuration screen, click the **Area** hyperlink at the top.
5. At long last, you can configure areas within the OSPF instance. By default, OSPF automatically adds the backbone area 0.0.0.0.
6. You can create another area which the firewall will belong to by defining the **Area ID** and setting the **Type**.
7. Click **Add**.
8. Define what routes to summarize, and what interface the area belongs to, by clicking the **Configure** button next to the area you would like to configure.
9. Define what the summary address is by filling this out in the **IP / Netmask** fields.

10. Specify that we would like to advertise this route by selecting the **Advertise** option.
11. Click **Add** to add this route to the configuration.
12. Click **OK** to exit. We will define how you can add interfaces to an OSPF area in the next example.

We are using the following values for this example:

Virtual Router	**Trust-VR**
Area	**9.9.9.9**
Area Type	**Stub**
Summary Route	**192.168.0.0/16**

To configure this example on the CLI:

```
set vrouter "trust-vr" protocol ospf area 9.9.9.9 stub
set vrouter trust-vr protocol ospf area 9.9.9.9 range 192.168.0.0 255.255.0.0
advertise
save
```

Interface Properties

Certain properties of OSPF can only be configured on an interface-by-interface basis. We will cover these settings in this section:

- **Bind to Area** An area that participates in OSPF must be configured to be in a specific area. Depending on what type of router this is, it may or may not be in the same area as other interfaces on the box (think ABR, backbone, or internal routers).
- **Protocol OSPF Enable** You can configure whether this protocol is enabled on an interface-by-interface basis. This allows you to configure it on the interface, while exempting the interface from participating in OSPF.
- **Reduce Flooding** This prevents LSAs from being flooded from the interface on a specific time interval, but rather they will only be sent when a change occurs.
- **Authentication** You have the following options to incorporate authentication on a per-interface basis:
  - **None** Do not use any authentication for OSPF on this interface.
  - **MD5 Keys** You can define the MD5 keys to use on this interface.
  - **Key ID** You can define the ID for this key; this will identify the key to the other OSPF peer.
  - **Preferred** Whether this key value is preferred over the other keys.

- **Password** Defines a plain-text password to use to authenticate OSPF to other peers.
- **Link Type** Defines what type of link this is—broadcast, point-to-point, or NBMA.
- **Passive Mode** Puts this interface into passive mode. This will not send out any routing information to other peers, but will only passively receive the routing updates.
- **Priority** This is the priority of the interface which will be used for DR router elections.
- **Cost** This is the cost associated with the interface.
- **Hello Interval** Defines a different Hello interval for this interface.
- **Retransmit Interval** The Retransmit Interval is used by the firewall to determine how long it should wait before retransmitting updates that were previously rejected by other transmissions.
- **Transit Delay** This is the amount of time which the firewall will wait before it will re-advertise the information it received on the interface.
- **Neighbor Dead Interval** Lets you define how long the firewall should wait until it considers a neighboring OSPF peer to no longer be active.
- **Neighbor List** Lets you define what your neighbors are on the interface. This is useful for NBMA networks.

Are You Owned?

Malicious OSPF Router Insertion

Many network administrators overlook the importance of authenticating their routing protocols, especially when they are configured on their "secure" internal networks. The truth is that attacking internal routing protocols by inserting a rogue router is very easy. All you have to do is place a malicious router in the network and have it participate in the routing infrastructure. In the case of OSPF, you would place the router in the network, and have it advertise itself as a member of the same area. The router would then almost instantly have knowledge of all the routes in that OSPF area (and maybe others as well). From there, an attacker could do anything, from advertise a better metric for a route, to redirect traffic, or even cause a DoS attack. To be sure this doesn't happen to you, make sure

Continued

your routing protocols use a strong authentication, such as MD5, for their message communications. Additionally, you should check your routers to see who your neighbors are, and that there aren't any unknown routers participating in the protocol.

Configuring OSPF on Interfaces

Several OSPF properties are configured on the interface level. In this example, we will show you how to add an interface to an OSPF area, and set some interface-specific properties of OSPF.

To configure this example in the WebUI:

1. Select **Network | Interfaces** and click the **Edit** hyperlink next to the interface you would like to configure OSPF on.
2. At the top of the screen, click the **OSPF** hyperlink to enter the interface-specific configuration.
3. Check the **Bind to Area** checkbox, and then select the appropriate **Area** to configure this interface in from the drop-down menu.
4. Check the **Protocol Enable** checkbox to enable OSPF on this interface.
5. Leave the **Reduce Flooding** box unchecked for this example, but you can use this option if you would like to prevent periodic OSPF LSA flooding.
6. Define **MD5 Authentication** for this interface to make sure your OSPF traffic is authenticated with a secure hash.
7. You must define an **MD5 Key**, which is a string of alphanumeric and non-alphanumeric values, as well as a **Key ID** for the key. You may also specify that this key is **Preferred**.
8. If this interface is on Ethernet, you should leave it in **Broadcast** mode, but if it is over a point-to-point WAN link, you should set it to **Point-to-Point**.
9. We will not make this interface **Passive** since we would like it to participate actively with other routers.
10. We will also leave the priority values, costs, and timing values at their default values, but if you would like to alter them, you would specify those values here at the interface.
11. You may also specify a **list of neighbors** you would like to participate in OSPF, but we will not do so in this example.

We are using the following settings in this example:

Bind to Area	**9.9.9.9**
Protocol OSPF	**Enabled**
Reduce Flooding	**Disabled**
Authentication	**MD5**
MD5 Key	**jf8*jfkal**
Key ID	**5**
Preferred	**Yes**
Link Type	**Broadcast**
Passive Mode	**Disabled**

To configure this example via the Juniper CLI:

```
set interface ethernet1 protocol ospf area 9.9.9.9
set interface ethernet1 protocol ospf enable
set interface ethernet1 protocol ospf authentication md5 "jf8*jfkal" key-id 5
set interface ethernet1 protocol ospf authentication active-md5-key-id 5
save
```

Configuring OSPF to Work with Tunnel Interfaces

OSPF is also capable of working with tunnel interfaces to bring dynamic routing capabilities over VPNs. This is a very attractive option when you have to provide routing for remote networks over VPNs. In this example, we will configure OSPF to work over tunnel interface Tunnel.1. It will be a part of area 0.0.0.0 and will advertise routes on to, and from, the remote office connected to this interface. We will assume you have already configured OSPF on the Trust-VR with Area 0.0.0.0, as well as created the Tunnel.1 interface and the VPN associated with it.

1. To configure this example, go to **Network | Interfaces** and click the **Edit** hyperlink next to the interface you would like to configure OSPF on.
2. At the top of the screen, click the **OSPF** hyperlink.
3. Check the **Bind to Area** box and select the appropriate **Area**.
4. Uncheck the **Demand Circuit** checkbox.
5. We will specify **Authentication** for our OSPF instance by using **MD5** authentication.
6. Specify a **MD5 Key**, **Key-ID**, and whether it is **Preferred**.
7. We will configure this link as **Point-To-Point** since it is going over a WAN interface.

8. If you would like, you can alter the default timing and cost values. For this example, we will alter the **Cost** to **50** since this is going over a VPN/WAN link.
9. Click **Apply**.
10. Now go back to the top of the screen and check the **Protocol Enabled** checkbox to enable it.
11. Click **OK**.

In this example, we are using the following configuration:

Virtual Router	**Trust-VR**
Interface	**Tunnel.1**
Protocol	**OSPF**
Bind to Area	**0.0.0.0**
Protocol Enabled	**Yes, after initial apply**
Demand Circuit	**No**
Authentication	**MD5**
MD5 Key	**8JFoa+[]**
Key-ID	**1**
Preferred	**Yes**
Link Type	**Point-to-Point**
Cost	**50**

To configure this example via the CLI:

```
set interface tunnel.1 protocol ospf area 0.0.0.0
set interface tunnel.1 protocol ospf enable
set interface tunnel.1 protocol ospf retransmit-interval 5
set interface tunnel.1 protocol ospf cost 50
set interface tunnel.1 protocol ospf authentication md5 "8JFoa+[]" key-id 1
set interface tunnel.1 protocol ospf authentication active-md5-key-id 1
save
```

NOTE

Other routing protocols can operate with tunnel interfaces to support VPNs, such as RIP and BGP. Remember, however, that these only function with route-based VPNs, since policy-based VPNs do not use a tunnel interface.

OSPF Informational Commands

You should familiarize yourself with a few OSPF-related commands on the Juniper firewalls. These commands will help you determine the state of the routing table, OSPF relationships, OSPF configurations, and other OSPF statistics. In this section, we will review valuable commands to know when working with OSPF.

Showing the Summarized OSPF Configuration

If you would like to print a summarized output of the OSPF configuration on the firewall, you can do so by issuing the command *get vrouter <vrouternname> protocol ospf*. This command will allow you to view the Router ID, whether OSPF is enabled, the number of areas, and the information specific to each area. See the following output which shows the output from a production system we ran this command on.

```
ns5gt-adsl-wlan-> get vrouter trust-vr protocol ospf
VR: trust-vr RouterId: 192.168.45.1
---------------------------------
Status:                                 enabled
State:                                  internal router
Auto-Vlink creation:                    disabled
Number of areas:                        3
Number of external LSA(s):              0
External LSAs with DNA:                 0
Advertising default-route lsa:          disabled
Default-route learnt by ospf:           will not be added to the routing table
RFC 1583 compatibility:                 disabled
Hello packet flood protection:  enabled (threshold is 20 packets per hello-
interval)
LSA flooding protection:                enabled (threshold 180 packets per 60
second(s))
Maximum Retransmit limit:               For nbrs on demand-circuits 12
                                        For nbrs on non-demand-circuits 24
Area 0.0.0.0
        Total number of interfaces is 0, Active number of interfaces is 0
        Intra-SPF algorithm executed 5 times
        Last Intra-SPF executed before 00:25:55
        Number of LSA(s) is 0
Area 3.3.3.3 (Stub)
        Total number of interfaces is 1, Active number of interfaces is 1
        Intra-SPF algorithm executed 5 times
        Last Intra-SPF executed before 00:25:55
        Number of LSA(s) is 0
Area 9.9.9.9 (Stub)
```

```
        Total number of interfaces is 1, Active number of interfaces is 1
        Intra-SPF algorithm executed 5 times
        Last Intra-SPF executed before 00:25:56
        Number of LSA(s) is 1

Inter-SPF algorithm executed: 5 times
Last Inter-SPF executed before 00:25:56
Extern-SPF algorithm executed: 5 times
Last Extern-SPF executed before 00:25:56
SPF Aborted: 0 times
```

Getting the OSPF Configuration

There is a simple way to capture all of the OSPF configuration without having to sift through your entire configuration. Simply issue the command *get vrouter <vrouternname> protocol ospf config*. This only prints the OSPF information from your configuration, omitting the other commands, as shown next.

```
ns5gt-adsl-wlan-> get vrouter trust-vr protocol ospf config
VR: trust-vr RouterId: 192.168.45.1
---------------------------------
set protocol ospf
set enable
set area 3.3.3.3 stub
set area 9.9.9.9 stub
set area 9.9.9.9 range 192.168.0.0 255.255.0.0 advertise
set hello-threshold 20
set lsa-threshold 60 180
set reject-default-route
exit
set interface ethernet1 protocol ospf area 9.9.9.9
set interface ethernet1 protocol ospf enable
set interface ethernet1 protocol ospf cost 10
set interface ethernet1 protocol ospf authentication md5 "jf8*jfkal" key-id 5
set interface ethernet1 protocol ospf authentication active-md5-key-id 5
set interface ethernet2 protocol ospf area 3.3.3.3
set interface ethernet2 protocol ospf enable
set interface ethernet2 protocol ospf cost 10
```

Showing the OSPF Interface Status

You can issue the *get vrouter <vrouternamе> protocol ospf interface* command to display the current state of the interfaces participating in OSPF. This can be useful to glean OSPF interface information in a concise manner.

```
ns5gt-adsl-wlan-> get vrouter trust-vr protocol ospf interface
VR: trust-vr RouterId: 192.168.45.1
---------------------------------
D - Down, L - Loopback, W - Wait, PTP - Point-to-Point
DR - Designated Router, BDR - Backup Designated Router
O - Other
Int            IpAddr        NetMask          AreaId        Status   St
------------------------------------------------------------------------------
eth1     192.168.1.1   255.255.255.0   9.9.9.9       enabled  DR
eth2     192.168.45.1  255.255.255.0  3.3.3.3       enabled  D
```

Showing OSPF Neighbors and the LSA Database

Sometimes things just don't go as planned and so you have to dig deeper to figure out what is causing the issue. Two OSPF commands that particularly come in handy for this task are the *get vrouter <vrouternamе> protocol ospf neighbor* and the *get vrouter <vrouternamе> protocol ospf database* commands. The first will print out a list of all of the OSPF routers that have established a neighbor relationship with your firewall, and the second command will list off the LSAs it has received, as well as their age. The issue many people often have is not forming the proper neighbor relationship with other OSPF peers. You need to make sure the areas match properly, and that if you have configured authentication, the keys match. Once a neighboring relationship has been established, it is important that the proper LSAs are transmitted to the neighbors, and that you are receiving an accurate picture of the network. That is where showing the LSA database can come in handy. It basically shows you everything that OSPF knows about your network in its raw form. The database does not have the routes calculated itself, but the information which it contains is used by the SPF algorithm to determine the shortest routes to each destination.

```
ns5gt-wlan-> get vrouter trust-vr protocol ospf neighbor
VR: trust-vr RouterId: 192.168.225.1
---------------------------------
                Neighbor(s) on interface tunnel.1 (Area 0.0.0.0)
IpAddr/IfIndex  RouterId        Pri State    Opt  Up           StateChg
------------------------------------------------------------------------------
192.168.222.1   192.168.223.1     1 Full     E    4d;05:41:58  (+6 -0)

                Neighbor(s) on interface wireless1 (Area 0.0.0.0)
```

```
                Neighbor(s) on interface trust (Area 0.0.0.0)


ns5gt-wlan-> get vrouter trust-vr protocol ospf database
VR: trust-vr RouterId: 192.168.225.1
---------------------------------

                        Router LSA(s) for area 0.0.0.0
  Link-State-Id   Adv-Router-Id      Age  Sequence# CheckSum
-------------------------------------------------------------------------------
  192.168.223.1   192.168.223.1      643 0x80002ab2   0x2f68
  192.168.225.1   192.168.225.1      645 0x80000a82   0xd421
    200.168.2.1     200.168.2.1      803 0x80001785   0x d84

                        Network LSA(s) for area 0.0.0.0
  Link-State-Id   Adv-Router-Id      Age  Sequence# CheckSum
-------------------------------------------------------------------------------
 192.168.222.20     200.168.2.1      290 0x8000174e   0xc481


                        AS External LSA(s)
  Link-State-Id   Adv-Router-Id      Age  Sequence# CheckSum
-------------------------------------------------------------------------------
        0.0.0.0   192.168.223.1      763 0x800023f3   0x6167
```

Displaying the OSPF Routing Table

As you may know, you can display the entire routing table by entering the *get route* command. If you would like to only view the routes that have been learned by OSPF, you can enter the *get route protocol OSPF* command. This will also show which VR knows the routes that are displayed (also useful for troubleshooting). If you would like to show a particular VRs routing table, you can issue the *get vrouter <vroutername> protocol ospf routes* command.

```
ns5gt-wlan-> get vrouter trust-vr route protocol ospf
H: Host C: Connected S: Static A: Auto-Exported
I: Imported R: RIP P: Permanent D: Auto-Discovered
iB: IBGP eB: EBGP O: OSPF E1: OSPF external type 1
E2: OSPF external type 2

Total 15/max entries
```

```
   ID           IP-Prefix       Interface         Gateway   P Pref    Mtr      Vsys
-------------------------------------------------------------------------------
  378           0.0.0.0/0           tun.1         0.0.0.0  E1   60     11      Root
  377   192.168.223.0/24            tun.1         0.0.0.0   O   60      2      Root
  376   192.168.222.0/24            tun.1         0.0.0.0   O   60      2      Root
* 375       10.55.0.0/24            tun.1         0.0.0.0   O   60     12      Root
```

Border Gateway Protocol

One of the earliest tasks network administrators faced was how to cooperatively route traffic with networks that were out of their administrative control. RIP and OSPF seemed to work well for internal use, but did not offer the control features many engineers felt were needed to route traffic between separate organizations. The creation of BGP solved this problem by allowing engineers to determine how to route traffic not strictly on the shortest path, but on an agreed path. This may sound odd, but since the Internet is actually run mostly by companies (AT&T, Sprint, Verizon, and so on) these companies form agreements on how they can pass traffic, and in what volumes. BGP is also very scalable, which is obviously a key factor in any protocol that must be distributed across the entire world.

As traditional routing functionality has been incorporated into firewalls, the need for support for BGP within firewalls has also increased. BGP has been supported for quite some time in the Juniper firewalls, but the recent introduction of the SSG firewalls has made this feature even more attractive. Since most Internet connections are on WAN-type interfaces (T1s, DS3s) it makes even more sense to forego an Internet router altogether and just place an SSG firewall in its place to handle that task.

In this section, we will lightly cover the main concepts of BGP—those pertinent to its application on the Juniper firewalls. We will begin with an overview of the functionality of the BGP protocol, and then cover the BGP properties configurable within the Juniper firewall. Lastly, we will delve into some BGP examples in the Juniper firewalls.

Overview of BGP

As mentioned earlier, BGP is a protocol that allows the use of additional routing variables to help add a customizable touch to routing decisions. BGP is considered to be a Path Vector routing protocol, due to its reliance on paths to determine routes. The current version of BGP in real-world use is BGP version 4, and is considered the de facto exterior routing protocol. BGP implements several techniques to handle the enormous task of routing traffic for the entire Internet. These techniques include using autonomous systems to represent entire networks, incremental updates, and classless interdomain routing (CIDR). BGP peers actually use TCP port 179 to communicate routing updates between each other in an efficient and reliable manner.

Autonomous Systems

An autonomous system (AS) is a network under administrative control by a single organization. A path is composed of a series of autonomous systems which can be used for routing decisions. An AS number must be unique and be assigned to an organization that would like to participate in BGP, although there are private AS numbers which can be used for internal purposes. AS numbers are distributed by IANA to organizations which can prove the need to have their own AS.

BGP Peers

Routers that participate in BGP routing must be connected to at least one other BGP router, known as a BGP peer. BGP peers exchange communication over TCP port 179. Updates are sent incrementally to their peers as new information becomes available. This helps reduce the overhead on the network from periodic routing updates. Peers must be configured to talk to their neighbor, or else the BGP session won't be established. There are ways to create multiple connections to your peers should you want to load balance, or account for a failed link.

BGP Attributes

Approximately 13 attributes are used by BGP to assist and enhance the routing process. Some attributes must be specified for a route, while others are optional. Additionally, some attributes are passed from AS to AS or router to router, while others are not, depending on whether the attribute is known by the system. A brief explanation of the attributes commonly used by the Juniper firewall follow:

- **AS Path** This attribute is mandatory for each route passed by BGP. It defines the AS numbers the route has traversed. This helps determine the best path and avoid routing loops. Additionally, you can enforce route decisions based upon the ASs the route has traversed. This attribute is passed on to other routers that the route is announced to, with the AS number of each AS appended to the path as it passes through the respective AS.
- **Next-Hop** This attribute contains the next-hop IP address that the router will take to reach the destination. It is both a well-known and mandatory attribute that must be included in routing updates to neighbors.
- **Origin** The origin attribute describes how the route was learned. This could be from an internal gateway protocol (IGP) such as OSPF, an exterior gateway protocol (EGP) like BGP, or it may not be known and so considered incomplete. This is thought to be a well-known mandatory value to be passed to peers.

- **Local Preference** This is the preferred exit point for an AS for routes that travel through it. This is a well-known attribute, but it isn't necessary it to be included in the route when passedn to a neighbor.
- **Atomic Aggregate** When a route is summarized into a less-specific route due to aggregation, this attribute may be set to let the BGP neighbor know the route was summarized, and the original path was lost. This is a well-known discretionary attribute.
- **Aggregator** When a route is aggregated, a BGP peer may specify which router ID and AS number was used for the aggregation. This is known as the Aggregator attribute. This is an optional attribute which can be transferred to neighbors.
- **Community** This is a customizable attribute that can be used to group routes together based upon their attributes. This is optional and may be passed to other BGP neighbors.
- **Multiple Exit Discriminator** This value is advertised to BGP peers to indicate what value is preferred into the network when multiple paths exist into the AS. This is an optional parameter that is not passed if a neighbor does not understand the attribute.
- **Cluster List** This attribute is defined for iBGP and is used to determine what route reflector the route has gone through.

BGP Messages

We already know that BGP communicates on TCP port 179. Within that communication channel, BGP uses several different control messages to perform BGP tasks.

- **Open Message** The open message is the first message exchanged between two BGP peers after the TCP session on port 179 has been initiated. This formally establishes the BGP connection between the peers.
- **Update Message** When BGP wants to announce or withdraw routes, it does so with information contained in the update message.
- **Notification Message** This message is exchanged when there is some sort of error detected by BGP. It will contain the reason for the message, and is followed by the closing of the BGP session.
- **Keep-Alive Message** This message is exchanged between BGP peers to keep the TCP session established, and to let each one know the other peer is still active.

IBGP

While EBGP deals with inter-autonomous system routing, iBGP handles BGP routing within an AS. IBGP is used primarily when BGP AS Path information must be maintained throughout an AS to other BGP peers. An iBGP network must be fully meshed, or have equivalent behavior with route reflectors or confederations. This is important to help maintain a loop-free topology. The rule-of-thumb here is that one iBGP peer will not re-advertise a route to another iBGP peer.

Route Reflection

One of the methods that iBGP can employ to help reduce the complexity of a BGP internal network is to use route reflectors. The main problem with any fully meshed network is the excessive cost and overhead associated with managing every combination of links between peers. The complexity grows very quickly with every peer added to the domain. Route reflectors help to subdivide an iBGP domain into a manageable topology. Route reflectors work similarly to designated routers in OSPF. All iBGP routers within a route reflector group send the routes to the route reflector, which then distribute them to other directly connected route reflector peers. Route reflectors have the following properties:

- **Route Reflector** Specifies whether the BGP instance should act as a route reflector or not.
- **Cluster ID** Specifies what the cluster ID should be for this route reflector instance.

Confederations

The other method to help achieve a full mesh environment for iBGP is to use *confederations*. These allow you to make sub-ASs, which are essentially ASs that are segmented further. The advantage here is that you don't have to mesh an entire iBGP network, but rather, you only have to mesh within sub-ASs, and then mesh those together with the other ASs. Again, this can greatly reduce the number of links you must maintain to help reduce complexity and cost. Confederations have the following properties:

- **Confederation** This specifies whether confederations are enabled or disabled.
- **ID** This is the AS number of the confederation.
- **Peer Member Area ID** This is the AS number of the sub-AS, which is part of the confederation.

Route Flapping

The Internet is a big place, and you can't control all of the events that occur. Sometimes, there may be a host somewhere on the Internet which is having connectivity issues and keeps going up and down. This change could ripple across the Internet over and over causing instability. To help counteract this effect, *Route Flapping* is a mechanism which scores routes based upon how stable they are. The value used to keep track of the routes trustworthiness is called the *figure of merit*. Basically, a route starts with a certain value, and when the route goes down or changes, its figure of merit will decrease. Once the value reaches a certain threshold, it will no longer be trusted. If the route does continue to flap, it will slowly regain its figure of merit through an algorithm which determines the value over time. Once the route reaches the figure of merit that is acceptable to trust a route again, it will be returned to the active state.

Configuring BGP

Now that we have reviewed the various concepts of BGP, let's explore how it is really implemented in the Juniper firewalls. We will begin our configuration discussion with an overview of the configuration properties within the Juniper firewall. We will then use these configuration properties to build real examples of BGP instances.

Configuration Properties

We will now cover the various properties that can be configured within BGP. There are a few places where it is configured, both at the virtual router and interface level. A single BGP instance can be configured per VR, and the number of routes you can have on the firewall is dependent on what firewall platform you are running BGP on. This is true for other protocols, but usually BGP is the protocol which requires the large routing table. As complex as BGP sounds, it actually isn't that bad to implement, as you will see.

VR Properties

VR properties are those configured per VR:

- **AS Number** This is required for each BGP instance. This should be the AS number you have been assigned by IANA, unless you are using a private AS number.
- **Keep Alive** This is the value you can set for how often a keep-alive message is sent to a peer.
  - **Use Node Default** Uses the default value
  - **Custom** You can define a custom value in seconds.

- **Always Compare MED State** This is used to always compare the MED values of routes to determine the best way into a network.
- **Route Flap Dampening State** This option enables Route Flap Dampening.
- **Ignore Default Route from Peer** Enabling this option will block a default route from being imported from a BGP peer.
- **Advertise Default Route** This will automatically advertise the default route to other BGP peers.
- **Hold Time** You can define what value will be used to determine how long to wait between message transmissions to a peer.
- **Disable** This will disable the hold-time functionality for this example.
- **Enable** To configure a hold-time to be employed, select this option.
- **Value** This is the time in seconds that will be used for the hold time.
- **Default Local Preference** This is the default local preference value to apply to a route.
- **Default MED** This is the default MED that will be advertised to another BGP peer.
- **Route Reflector** You can configure a BGP instance to be a route reflector, and also which cluster ID for the cluster group it will be a part of:
- **Disable** Do not configure this instance as a route reflector.
- **Enable** Set this option to enable route reflection on this VR instance.
- **Cluster ID** If you enable route reflection, you must set a Cluster ID for the route reflector to reflect for.
- **Synchronize with IGP** This option enables BGP to synchronize routing information with IGP.
- **BGP Enabled** Enable this BGP instance.

Configuring a BGP Instance in a VR

We will begin our BGP examples by configuring a BGP instance within a VR. This is the first step you must take before configuring BGP on individual interfaces.

To configure this example via the Juniper WebUI:

1. Go to **Network | Routing | Virtual Routers**.
2. Click the **Edit** button next to the VR you would like to create a BGP instance in.

3. Scroll down to the bottom of the screen and click the **Create BGP Instance** hyperlink.
4. Define the **AS Number** your firewall will use for communications within the BGP protocol.
5. You can define a custom **Keep-Alive** value, or leave it at the **Node Default**. This value will be used to determine how often it should send keep-alive messages to BGP peers to keep the TCP session established.
6. Each BGP instance supports several options, such as **Always Compare MED**, **Route Flap Damping State**, **Ignore Default Route from Peer**, and **Advertise Default Route**. For our example, we will enable **Route Flap Damping** and **Ignore Default Route from Peer**.
7. You can define the **Hold Time** BGP will use for this routing instance, or you can disable the hold time.
8. Define the **Local Preference** and **MED Values** to be used for this instance. Of course, you can always alter these on a route-by-route basis with Route Maps.
9. If this firewall is participating in iBGP, you can enable it as a **Route Reflector**, as well as define the **Cluster ID**.
10. If you are using iBGP, you might want to select **Synchronize with IGP** to ensure the routing protocols exchange any advertised routing information.
11. **Enable BGP** in order to turn it on within the VR you have configured.
12. Click **OK**.

For this example, we are using the following Properties:

Virtual Router	**Trust-VR**
AS Number	**65512**
Keep Alive	**Use Node Default**
Always Compare MED	**Disabled**
Route Flap Damping State	**Enabled**
Ignore Default Route from Peer	**Enabled**
Advertise Default Route	**Disabled**
Hold Time	**Enabled (180 Seconds)**
Default Local Preference	**100**
Default MED	**0**
Route Reflector	**Disabled**

Synchronize with IGP	**Disabled**
BGP Enabled	**Checked (Enabled)**

To configure this example via the Juniper CLI:

```
set vrouter trust-vr protocol bgp 65512
set vrouter trust-vr bgp enable
unset vrouter trust-vr protocol bgp synchronization
save
```

Neighbor Properties

You must configure each neighbor that your BGP instance will peer with. You can do that under the BGP neighbor properties, described next:

- **AS Number** This is the AS number of the neighbor you wish to peer with.
- **Remote IP** This is the BGP neighbor's IP address.
- **Local IP/Netmask** This is the local IP address and netmask which will connect to the neighbor.
- **Peer Group** You can configure a BGP peer group, and you can specify it here.
- **Outgoing Interface** Determines what interface you would like BGP to connect to its neighbor on.
- **EBGP Multihop** You can configure this neighbor as being multiple hops away. If you enable this you can configure the value that defines how many hops it is.
  - **Disable** Disables the option.
  - **Enable/Value** This is the value for how many hops away the peer is.
- **Keep Alive** The value for the keep alive
  - **Use Node Default** The default value for the system.
  - **Custom** Defines a custom value.
- **Peer Authentication** Peer authentication can be used to authenticate the identity of a neighbor, as well as the integrity of routing updates.
  - **Disabled** Disables peer authentication.
  - **Enabled/Password** This is the MD5 password, which is used to calculate the hash.
  - **MED** The MED value that you will pass to this peer.
  - **Send Default Route to Peer** This will advertise the default route to your BGP peer.

- **Ignore Default Route from Peer** By enabling this option, you will not accept a default route advertised from this peer.
- **Self as Next Hop** This option defines the next hop as yourself to this neighbor.
- **Send Community** Sending the community will pass this value on to your neighbor instead of not transmitting it.
- **Force Reconnect** Should this neighbor lose the BGP connection, the Force Reconnect option will attempt to reconnect to this peer.
- **Remove Private AS** The Remove Private AS option will remove a private AS from the AS path list if it is present.
- **Hold Time** The amount of time between keep-alive and update messages for this BGP peer.
- **Disable** Do not enable the hold time for advertisements.
- **Enable/Value** The value in seconds that defines what the hold time should be.
- **Retry Time** The amount of time between retrying to connect to this neighbor.
- **Weight** Weight is an attribute value that can be applied to a neighbor.
- **Incoming Map Tag** This is the incoming Map-Tag that should be applied to routes learned from this neighbor.
- **Outside Map Tag** This is the Outgoing Map-Tag that will be applied to routes advertised to this neighbor.
- **Peer Enabled** This option allows you to enable this peer. If it is disabled, the router will not communicate with the neighbor.

Configuring a BGP Neighbor

BGP requires you to configure your directly connected neighbors for proper functionality. In this example, we will configure a neighbor which will be a BGP peer.

To configure this example via the Juniper WebUI:

1. Select the VR to configure neighbors for by going to **Network | Routing | Virtual Routers** and clicking the **Edit** button next to the chosen VR.
2. Scroll to the bottom of the screen and click the **Edit BGP Instance**.
3. Click the **Neighbors** link at the top of the screen.
4. Once in the neighbor configuration, begin defining neighbors by defining the **AS Number** of the BGP neighbor.
5. Define the **Remote IP** address (the IP address of the peer BGP router).

6. The **Local IP/Netmask** is the IP address on the firewall (and netmask) you would like the BGP instance to form a peer relationship. The local IP address is what your BGP neighbor would consider their remote IP address for your host.
7. The BGP **Peer Group** for this neighbor can be configured by selecting it in the drop-down menu.
8. Configure which interface the BGP peer is located on by setting the **Outgoing Interface**.
9. Click the **Add** button to add the neighbor to the configuration.
10. Once the neighbor is added, configure some neighbor-specific attributes for each neighbor by clicking the **Configure** hyperlink next to the desired neighbor.
11. Define whether the BGP peer should be considered eligible for **EBGP Multihop**. If you enable it, you can also define the number of hops away it is.
12. Override the default BGP **Keep Alive** settings on a peer-by-peer basis if needed. This can be done by using a **Custom** value in seconds in this node.
13. If your neighbor is using **Peer Authentication**, you can define the **MD5 Password**, as well as enable peer authentication.
14. The **MED** value can be set for each neighbor, or else it will take the default value.
15. Advertise your default route to this neighbor by selecting the **Send Default Route to Peer** option.
16. If you wish to prevent BGP from learning a default route from your peer (without using a route map), select the **Ignore Default Route from Peer** option.
17. Sometimes you will want to set the **Self as Next Hop** option in the AS path announced to the BGP peer. This can be configured here.
18. Since Communities are an optional attribute, you can set the **Send Community** or not advertise your community string to this neighbor.
19. The **Force Reconnect** option can be used to attempt to reconnect a BGP session if it should be lost
20. If you are using iBGP, or your router is at the edge of a private BGP network, you may need to configure the **Remove Private AS** option to remove the private AS number from the BGP AS path.
21. If you would like to configure the **Hold Time** for routes from this neighbor, you may do so here.
22. The **Retry Time** may also be configured per neighbor.
23. Optionally, you may define a **Weight** value for this route.

24. You can configure both an **Incoming Route Map** and an **Outgoing Route Map** on a per-neighbor basis. This helps define what routes will be accepted and advertised to a particular neighbor.
25. The last option to configure is enabling the BGP peer by selecting the **Peer Enabled** option. You must do this for your firewall to recognize the peer.
26. Click **OK**.

In this example, we used the following settings:

Virtual Router	**Trust-VR**
BGP Neighbor	**192.168.99.60**
Neighbor AS	**65513**
Local-IP	**192.168.45.1/24**
Reject Default Route	**Enabled**
Force Reconnect	**Enabled**
Peer Enabled	**Yes**

To configure this example via the Juniper CLI:

```
set vrouter trust-vr protocol bgp neighbor peer-group "BGP1"
set vrouter trust-vr protocol bgp neighbor 192.168.99.60 remote-as 65513 local-ip
192.168.45.1/24                      outgoing-interface ethernet2
set vrouter trust-vr protocol bgp neighbor 192.168.99.60 enable
set vrouter trust-vr protocol bgp neighbor 192.168.99.60 reject-default-route
set vrouter trust-vr protocol bgp neighbor 192.168.99.60 force-reconnect
save
```

AS Paths

BGP calculates its routes based upon several attributes which can contribute to the overall preference of the route. AS path's define what AS systems the route has traversed. This can be a critical attribute when conforming to agreements with partner ISPs or other organizations. An AS Path also allows you to evaluate where the route came from to help determine its trustworthiness. Juniper firewalls allow you to match routes based upon the AS path with an AS access list as follows:

- **AS Path Access List ID** This is the access list ID value for this entry.
- **Permit** You can define whether to *permit* or *deny* this route based upon whether it matches or not.
- **AS Path String** You can either define an AS number, or a regular expression to define how to match an AS path.

- Regular Expression Symbols:

  ^ specifies an AS number at the beginning of the expression.

  $ specifies that the number is at the end of the path.

  { and } specify the beginning and end of the AS set, respectively.

  (and) specify the beginning and end of the Confederation AS set.

  . matches a single character

  * will match zero or more characters in the AS

  .+ matches one or more of any character

  – matches either zero or one instance of a character

  [and] specifies a set of characters to be matched

  - matches a range of AS numbers

Advertising BGP Routes

Once you have BGP set up, you must define which routes you would like BGP to advertise to other networks. BGP has the following properties for configuring a network which it will advertise:

- **IP/Netmask** This is the IP Address and the subnet mask of the network you would like to advertise.
- **Weight** This is the weight you should apply to this route on the firewall.
- **Route Map** This is the route map that will set the attributes of this route.
- **Check Reachability** This option will enable BGP to check to see if the route prefix is reachable before advertising the route.

Configuring a Route to Advertise via BGP

In this example, we will configure a network to advertise via BGP. We will use the virtual router VR-1025, and employ the Attribute-Set route map to set the route. Lastly, we will check to make sure the route is reachable before advertising it.

1. Go to **Network | Routing | Virtual Routers** and click the **Edit** button next to the VR you would like to advertise the route on.
2. Scroll to the bottom of the screen and click the **Edit BGP Instance** hyperlink.
3. In the BGP configuration screen, click the **Networks** hyperlink at the top of the page.
4. Specify the **IP/Netmask** for the network you would like to advertise.

5. Define the **Weight** for the route. If you do not specify a weight, it will be set to 32768.
6. You can specify the attributes for the route you are advertising by using a route map. Select the **Route Map** in the drop-down menu.
7. If you would like to check the reachability of the route, specify to check it, and also define the **IP/Network** the firewall should check before the route is advertised.
8. Click **Add**.

In this example, we are using the following examples:

Virtual Router	**VR-1025**
IP / Netmask	**192.168.0.0/16**
Weight	**32768**
Route Map	**Attribute-Set**
Check Reachability	**172.16.1.0/24**

To configure this example in the Juniper CLI:

```
set vrouter "VR-1025" network 192.168.0.0/16 route-map "Attribute Set" check
172.16.1.0/24
save
```

Examples of an AS Path

We would like to point out a couple of regular expressions to help show how these can be used to form AS paths.

- The path ^111 stands for any AS that starts with 111, so this could include 11122, 111, 1110, and so on.
- The path $650 stands for any AS that ends in 650, which could include 11650, 2650, 650, and so on.
- The path 1000-2000 specifies every AS number between 1000 and 2000.
- The path 99.1 would match 9901, 9911, 9921, 9931, 9941, 9951, 9961, 9971, 9981, and 9991.

Communities

BGP allows you to add a customizable touch to enterprise routing by using communities. A community is essentially a custom grouping that allows you to make routing decisions based upon the community string. You can think of a BGP community list as like an access list for

BGP communities. It can be used to match BGP communities, or group BGP routes together. Route maps can use BGP communities to match routes. The creators of BGP pretty much leave it up to you to call the shots with communities, but the following are the properties you can configure in the Juniper firewall:

- **Community List ID** This is the ID which the firewall will reference the list by. It will be an integer value.
- **Permit** You can define whether to *permit* or *deny* a route matching this community value if the community list is used to match a route.
- **Comm Type** There are several options you can perform actions upon with the *Comm Type*. They are as follows:
  - **AS** These options will not advertise the AS number defined to other autonomous systems.
  - **AS number** This is the AS number.
  - **final two octets** This is the ID number of the community string you wish to match this community access list upon.
  - **No-advertise** The firewall will not advertise the listed AS numbers to any other peer device.
  - **No-Export** The firewall will not advertise the route to other EBGP peers, except to other sub-AS values.
  - **No-Export-Subconfed** This option will not advertise the route to any sub-AS.
  - **None** This will not perform any of the preceding Comm Type actions on the community.

Configuring a BGP Community

In this example, we will configure a BGP community within a BGP instance. As mentioned earlier, BGP communities can be useful in helping group routes and apply policies upon custom configuration.

To configure this example via the Juniper Web UI:

1. Enter the configuration for the BGP instance you wish to use to configure the BGP community. You can do this by going to **Network | Routing | Virtual Routers** and clicking **Edit** next to the VR you would like to configure the community on. Afterward, scroll to the bottom of the screen and click the **Edit BGP Instance** hyperlink.
2. At the top of the next screen, click the **Community** hyperlink to enter the community configuration.

3. Define a **Community List ID** using an integer value between 1 and 99.
4. Define whether to **permit** or **deny** the matched community value.
5. The **Comm Type** can match different BGP communities, as well as define how the route should or should not be advertised. This is achieved by defining the Community as **AS**, **No-Advertise**, **No-Export**, **No-Export-Subconfed**, or **None**. You must also define the **AS Number**, as well as the **Final Two Octets**.
6. Click **Add** to add each statement to the Community List.

In this example, we are using the following configuration:

Virtual Router	**Trust-VR**
Community List	**1**
Permit	**Permit**
AS	**Set**
AS Number	**65513**
Final Two Octets	**65513**

To configure this example via the Juniper CLI:

```
set vrouter trust-vr protocol bgp community-list 1 permit as 65513 65513
save
```

Route Aggregation

As the number of routes a router must account for increases, so does the complexity of the network, and the hardware costs it entails. More routes can also affect convergence time and network stability. One way of minimizing these ill effects is to summarize several more specific routes into one or a few general routes. *Route aggregation* in BGP allows you to do just this. We will cover the properties of route aggregation within the Juniper firewall in this section.

- **Aggregate State** You can configure the virtual router to either aggregate routes (**Enable**) or not to aggregate routes (**Disable**).
- **IP/Netmask** This is the summarize route you will advertise in place of the more specific routes you aggregate.
- **AS Set** This option defines whether or not the AS numbers that helped form these routes should be included in the aggregate routes.

- **Suppress Option** These are options you can configure for this route:
  - **Summary-Only** This option will not advertise any more specific routes.
  - **Route Map** If you define a route map here, you can define what more specific routes you would like to advertise.
- **Advertise Map** You can use an advertise map (route map) to select which routes will be used to build this aggregate route.
- **Attribute Map** This map allows you to use a route map to set attributes of the routes you would like to define.

Configuring Route Aggregation

In this example, we will configure BGP route aggregation to summarize a list of addresses to match in a route map, as well as set attributes on the route from another route map.

To configure this example via the Juniper WebUI:

1. Route aggregation is configured within the BGP instance for the VR. To configure the route aggregation, go to **Network | Routing | Virtual Routers** and click **Edit** next to the VR you would like to configure route aggregation on.
2. Scroll to the bottom of the screen and click **Edit BGP Instance**.
3. At the top of the screen, click **Aggregate Address** to enter the route aggregation configuration.
4. You must set the **Aggregate State** to **Enable** to turn on aggregation.
5. Define the **IP Address / Netmask** which will be the summarized route.
6. You can choose to set the AS on this aggregate route by selecting the **AS-Set** option.
7. Set the **Suppress Option**. Your choices are **Summary-Only** or **Route Map**.
8. Define the routes to be summarized with the **Advertise Map**, which can be selected from the drop-down menu.
9. The **Attribute Map** can be selected to specify what attributes of each route you would like to set as part of this aggregate route.
10. Click **Add**, and then click **OK**.

In this example, we are using the following values:

Virtual Router	**VR-1025**
Aggregate State	**Enable**
IP / Netmask	**192.168.0.0/16**
AS-Set	**Set**

Suppress Option	**Summary Only**
Advertise Map	**Route Map 1**
Attribute Map	**Attribute Set**

To configure this example via the Juniper CLI:

```
set vrouter trust-vr protocol bgp aggregate ip 192.168.0.0/16 as-set summary-only
advertise-map "Route                Map 1" attribute-map "Attribute-set"
save
```

NOTE

You must enable route aggregation before you enable the BGP instance.

Configuring Route Reflectors

When you need to run iBGP and cannot implement a full-mesh environment, route reflectors can simulate the effect of having a full-mesh environment within BGP. In this example, we will configure route reflection and set a neighbor to be a route reflector client. Note that you must disable the BGP instance before you can configure route reflection.

To configure this example in the Juniper WebUI:

1. Select **Network | Routing | Virtual Routers** and click **Edit** next to the VR you would like to configure route reflection on.
2. Scroll to the bottom of the screen and click **Edit BGP Instance**.
3. Make sure **Route Reflector** is set to enable, and that the **Cluster ID** is set to a value.
4. Click **Apply**.
5. Go to the **Neighbor** hyperlink at the top of the screen.
6. Click the **Configure** hyperlink next to the neighbor you would like to set as a route reflector client.
7. Select **Reflector Client**, and then click **OK**.

In this example, we are using the following settings:

Virtual Router	**VR-1025**
Route Reflector	**Enabled**
Cluster ID	**10**

Neighbor	**192.168.99.60**
Route Reflector Client	**Enabled**

To configure this example in the Juniper CLI:

```
set vrouter "VR-1025" protocol bgp reflector
set vrouter "VR-1025" protocol bgp cluster-id 10
set vrouter trust-vr protocol bgp neighbor 192.168.99.60 reflector-client
save
```

Configuring a Confederation

Route reflectors are not the only mechanism for supporting an iBGP environment without truly having a full-mesh network. Confederations also solve this problem. By configuring BGP with sub-ASs, you can minimize the need for a complete full mesh-network and design it around smaller AS networks. In this example, we will configure a confederation which has a sub-AS as a member. Note that you must have BGP disabled before you can configure the confederation.

To configure this example in the Juniper WebUI:

1. Select **Network | Routing | Virtual Routers** and click **Edit** next to the VR you would like to configure a confederation on.
2. Scroll to the bottom of the screen and click **Edit BGP Instance**.
3. At the top of the screen, click the **Confederation** hyperlink.
4. Click the **Enable** option.
5. Specify the main AS as the **ID**.
6. Leave the compatibility set to **RFC-1965**.
7. Specify the **Peer Member Area ID** and click **Add**.
8. Click **OK**.

In this example, we used the following configuration:

Virtual Router	**VR-1025**
Confederation	**Enabled**
ID	**65512**
Peer Member Area ID	**65514**

To configure this example in the Juniper CLI:

```
set vrouter "VR-1025" protocol bgp confederation id 65512
set vrouter "VR-1025" protocol bgp confederation peer 65514
save
```

BGP Informational Commands

BGP is definitely not considered a lightweight protocol, so you must have a lot of capabilities to gather information, and troubleshoot issues. In this section, we will review several essential BGP commands which gather information from the firewall.

Summarizing BGP State

When you want an overview of the BGP configuration on a particular VR, the best command to run is *get vrouter <vrouternname> protocol bgp*. This command shows you information such as whether BGP is enabled, the AS number, timing values, MED and local preference info, iBGP settings such as route reflecting and confederations, and several other valuable pieces of information. We have included the output of running this command on our trust-vr, which is shown next:

```
ns5gt-adsl-wlan-> get vrouter trust-vr protocol bgp
Admin State:           enable
Local Router ID:       192.168.45.1
Local AS number:       65512
Hold time:             180
Keepalive interval:    60 = 1/3 hold time, default
Retry time:            120
Local MED is:          0
Always compare MED:    disable
Local preference:      100
Route Flap Damping:    disable
IGP synchronization:   disable
Route reflector:       disable
Cluster ID:            not set (ID = 0)
Confederation based on RFC 1965
Confederation:         disable (confederation ID = 0)
Member AS:             none
Origin default route:  disable
Ignore default route:  disable
```

Viewing the BGP Configuration

The configuration file on a firewall can be pretty lengthy. When you want to find a specific piece of information relating to the BGP configuration, issue the *get vrouter <vrouternname> protocol bgp config* command. This will output the BGP-related configuration statements right to the command line. See the following example:

```
ns5gt-adsl-wlan-> get vrouter trust-vr protocol bgp config
set protocol bgp 65512
```

```
set enable
unset synchronization
set as-path-access-list 1 permit "130"
set neighbor peer-group "BGP1"
set neighbor 192.168.99.60 remote-as 65513 local-ip 192.168.45.1/24 outgoing-
interface ethernet2
set neighbor 192.168.99.60 enable
set neighbor 192.168.99.60 reject-default-route
set neighbor 192.168.99.60 force-reconnect
set community-list 1 permit 42926093
set community-list 2 permit none
set community-list 3 permit 4293525482
set community-list 4 permit 4293525481
exit
set interface ethernet2 protocol bgp
```

Viewing BGP Neighbors

When you want to view the connection state between your firewall and your BGP neighbors, you can issue the *get vrouter <vroutername> protocol bgp neighbor* command, as follows:

```
ns5gt-adsl-wlan-> get vrouter trust-vr protocol bgp neighbor
Peer AS     Remote IP       Local IP          Wt Status         State     ConnID
Up/Down
--------------------------------------------------------------------------------
  65513     192.168.99.60   0.0.0.0          100 Enabled  ACTIVE         0
00:35:00

total 1 BGP peers shown
```

Viewing BGP Flapping Information

When you have BGP flapping enabled, and things just aren't quite working right, it might be time to issue the *get vrouter <vroutername> protocol bgp flap-damping* command to check if a BGP route is unstable. This command can help reveal some valuable information about your peers if they are advertising and withdrawing routes constantly. If all is well, you should see very little output since there won't be many flapping events to take note of, as shown next:

```
ns5gt-adsl-wlan-> get vrouter trust-vr protocol bgp flap-damping
Route Flap Damping             :    enabled
Reuse                                    :    1024
Suppress Limit                      :    2048
Max Penalty                          :    4096
Reachable Half Life             :    5 minutes 0 seconds
```

```
Unreachable Half Life          :    15 minutes 0 seconds
Maximum Reachable  Hold Time    :    15 minutes 0 seconds
Maximum Unreachable Hold Time  :    30 minutes 0 seconds
Decay factor                          :    4

--------------------------------------------------------------------------------
total: 0 flapping, 0 suppressed(*).
histAge/lastEvnt/TTL: seconds
--------------------------------------------------------------------------------
event prefix          peer            penalty flap histAge lastEvnt TTL
--------------------------------------------------------------------------------
```

Displaying the BGP Routing Table

The Internet is a big place, and the routing protocol that drives it is no less extensive. When running BGP, you will undoubtedly want to verify the routes your firewall has knowledge of in a protocol-specific context. Although you can issue the *get route* command to display the entire routing table that your firewall has knowledge of, you might want to consider requesting only the BGP routing table specific to a particular VR. This can be done by specifying the *get vrouter <vrouternname> route protocol BGP* command. Don't forget that you can have the output filtered by placing a pipe at the end of the command, followed by the route you want to search for.

For example, say you want to search your entire BGP routing table on the trust-vr router for a route to 192.168.52.x to see what the subnet boundary is for the route. You could simply issue the *get vrouter trust-vr route protocol bgp | 192.168.52* and the firewall would only display any route matching 192.168.52.x.

Another way to glean even more information from the BGP routing table in a more structured format is to issue the *get vrouter <vrouternname> route protocol bgp prefix <IP Address/Netmask>*. This will give an output defining the route in a table format which you may find more readable. We have included an example searching the trust-vr for the 192.168.52.0/24 route:

```
ns5gt-adsl-wlan-> get vrouter trust-vr route protocol bgp prefix 192.168.52.0/24
H: Host C: Connected S: Static A: Auto-Exported
I: Imported R: RIP P: Permanent D: Auto-Discovered
iB: IBGP eB: EBGP O: OSPF E1: OSPF external type 1
E2: OSPF external type 2

Total 6/max entries

   ID          IP-Prefix         Interface         Gateway        P Pref    Mtr
Vsys
    1    192.168.52.0/24    Ethernet2   192.168.45.254   250       700      Root
```

Route Redistribution

You might think that one routing protocol is more than enough for your network, but some administrators find themselves in a situation where they have to implement more than one routing protocol in a network. This often occurs when BGP is needed, when proprietary routing protocols are used such as EIGRP, when legacy routing support must be enabled, or when organizations using different routing protocols merge. By default, different routing protocols do not share their routing information with each other, even in the same VR. This is because they do not understand each other, and also because the metrics are often incompatible. If full connectivity is needed across the network, the routes from one protocol must be redistributed into another routing protocol so they can be advertised by the other protocol.

In this section, we will cover how route redistribution works, and how it enables you to handle multiple routing protocols in a network. We will give examples of how redistribution can be performed between different protocols, and some best practices to help strengthen your knowledge of multiprotocol routing.

Redistributing Routes in the Juniper Firewall

Route redistribution essentially works by pulling routes out of the routing table from one protocol, and then advertising them out another protocol. The Juniper firewalls use route maps to select which routes and which protocols to pull from the routing table, as well as which routes to advertise. As mentioned earlier in this chapter, route maps can match routes both by the route prefix itself, or even by properties of the route, such as metrics, and protocol-specific attributes. Of course, route maps are configured within VRs, so you need to make sure you create the route map in the appropriate VR.

We have already discussed some of the motivations for using route redistribution, but there are also several caveats and important points you should be aware of before implementing route redistribution.

Important Points to Consider

Before you design or implement route redistribution, you should take into consideration potential issues you may introduce into your network if you don't take the necessary precautions.

- Route redistribution can cause routing loops if routes are not carefully filtered. Since you are bypassing many of the processes that routing protocols take to ensure loop-free environments, you are subject to human error.
- Routes may take undesirable paths that are not optimum. This occurs because routes that are injected might have metrics which do not accurately represent the metric scheme of the protocol receiving the routes.
- Building on the last point, routing metrics may not be equivalent between two different protocols. For instance, RIP's metric is based upon the number of routers

(hops) that traffic may pass through up to 15. On the other hand, OSPF's metric is traditionally based upon the inverse of the link speed. So if you directly inputted a RIP route into OSPF, you might not evaluate the metrics properly. This would especially be true since RIP views a 56k link the same as a T3.

- You may advertise routes which you do not want other entities to be aware of, or advertise private addressing to a trading partner. If you are not careful to filter what routes you want to advertise, you may inadvertently advertise routes to traffic you don't want to advertise to others. This is particularly common when routing private IP address ranges when you have either a trading partner, or some other party that you might announce some routes to. If you advertise the same range as someone else, you might cause a loss of connectivity, depending on whether the metric for the route in your network is more attractive than your trading partners.

Redistributing Routes between Routing Protocols

Now that we have pointed out a couple of caveats, we will cover how to redistribute routes between different routing protocols on the Juniper firewall. We will begin with redistributing routes into RIP from other protocols, followed by OSPF and BGP.

Redistributing Routes to RIP

RIP is somewhat of a legacy routing protocol, but it is certainly still implemented in production environments. Sometimes you cannot simply remove RIP because a particular system needs it and doesn't support other routing protocols. In that case, you may have to redistribute other routes from other protocols into RIP so the systems that rely on RIP for routing information can receive it.

Before you can redistribute routes, you will need to identify what routes you want to match. This is commonly done with a route map. Of course, route maps can match certain attributes of routes, and actually use route maps to match specific IP prefixes. You must configure a route map before you can redistribute routes in RIP. This is done by creating the access list, if necessary, and then creating the route map. After the route map is created, you can apply it to the RIP instance for redistribution. Finally, you must configure the RIP instance to define what protocol you would like to redistribute from. Your choices for redistributing routes into RIP are BGP, OSPF, Connected, Imported, and Static.

Redistributing Static Routes into RIP

In this example, we will configure the firewall to redistribute Static routes into RIP which match 192.168.0.0/16 in the VR-1025.

To configure this example in the WebUI:

1. We begin by creating an access list. Go to **Network | Routing | Virtual Routers** and click the **number** hyperlink in the Access List column and row for your VR.
2. Click **New** in the upper right-hand corner.
3. Define a **Name**, a **Sequence No**, an **IP Address / Netmask**, and an Action before clicking **OK**.
4. Create the route map which references the access list just created. Go back to the **Network | Routing | Virtual Routers** page and click the **number** under the Route Map column, and on the row of your VR.
5. Click the **New** button in the upper right-hand corner of the screen.
6. Define a **Name**, **Sequence Number**, **Action**, and for our example, define only the Access-List with an Access List value. Click **OK**.
7. Apply the route map to the RIP instance by going to **Network | Routing | Virtual** Routers, and then clicking **Edit** next to the VR you would like to edit.
8. Scroll to the bottom of the screen and click the **Edit RIP Instance** hyperlink.
9. At the top of the screen, click the **Redistribute Rules** hyperlink.
10. In the **Route Map** drop-down menu, select the route map you just configured for the purpose of selecting the routes to import.
11. In the Protocol field, choose the **Static** option, and then click **Add**.

In this example, we are using the following settings:

Virtual Router	**VR-1025**
Access-List	**50**
Sequence No.	**1**
IP / Netmask	**192.168.0.0/16**
Action	**Permit**
Route Map	**RIP-Match**
Sequence No.	**1**
Action	**Permit**
Access-List	**Checked**
Access-List (ID Chosen)	**50**
Redistribute	**Static into RIP**

To configure this example via the Juniper CLI, perform the following steps:

```
set vrouter VR-1025 access-list 50
```

set vrouter VR-1025 access-list 50 permit ip 192.168.0.0/16 1

```
set vrouter VR-1025
set vrouter VR-1025 route-map name "RIP-Match" permit 1
set vrouter VR-1025 route-map name "RIP-Match" 1
set match ip 50
exit
exit
set vrouter VR-1025 protocol rip redistribute route-map "RIP-Match" protocol
static
save
```

Redistributing Other Protocols into RIP

There isn't actually much to be said in this section that wasn't covered in the previous example. Redistributing other protocols into RIP follows all of the same steps except the very last. For other protocols, when you are in the Redistribute Rules section, simply choose the protocol you want to redistribute into RIP. In the last example, we used Static, but we could have selected OSPF, BGP, Imported, or Connected instead. That's all there is to it.

NOTE

By default, RIP will give any imported metric a value of 10 (hops). If you would like to change this, you must either do this in the Set Attributes of the route map, which you use to import routes, or by altering the Metric for Redistributed Routes.

Redistributing Routes into OSPF

For many of the same reasons you might need to redistribute routes into RIP, you might also want to import routes into OSPF. For instance, say you had to support some legacy equipment which could only use RIP, but RIP wasn't scalable enough for your entire organization's routing. You could implement OSPF, and redistribute routes from RIP into OSPF so machines on your OSPF-supported networks would have knowledge of the RIP routes.

Redistributing routes into OSPF is essentially the same as doing so with RIP. You need to create a route map, which usually references an access list, and then you must configure the OSPF instance to redistribute routes matching the route map from a certain routing protocol (BGP, RIP, Connected, Imported, or Static) into OSPF.

Redistributing RIP Routes into OSPF

In this example, we will redistribute routes in RIP which match a particular route map, into the OSPF protocol. The OSPF instance we are redistributing routes in is the in VR-1025 virtual router. We will borrow some of the configuration from our previous example for creating route maps.

To configure this example in the Juniper WebUI:

1. Begin by creating an access list. Go to **Network | Routing | Virtual Routers** and click the number hyperlink in the **Access List** column and row for your VR.
2. Click **New** in the upper right-hand corner.
3. Define a **Name**, a **Sequence No**, an **IP Address/Netmask**, and an **Action** before clicking **OK**.
4. Create the route map which references the access list just created. Go back to the **Network | Routing | Virtual Routers** page and click the number under the Route Map column, and on the row of your VR.
5. Click the **New** button in the upper right-hand corner of the screen.
6. Define a **Name**, **Sequence Number**, **Action**, and for this example, also define the **Access-List** with only an **Access List** value. Click **OK**.
7. Apply the route map to the RIP instance by going to **Network | Routing | Virtual Routers**. Click **Edit** next to the VR you would like to edit.
8. Scroll to the bottom of the screen and click the **Edit OSPF Instance** hyperlink.
9. At the top of the screen, click the **Redistribute Rules** hyperlink.
10. In the **Route Map** drop-down menu, select the route map you just configured for the purpose of selecting the routes to import.
11. In the **Protocol** field, choose the protocol you want to redistribute from. In our example, we will use **RIP**. Afterward, click **Add**.

In this example, we used the following settings:

Virtual Router	**VR-1025**
Access-List	**50**
Sequence No.	**1**
IP / Netmask	**192.168.0.0/16**
Action	**Permit**
Route Map	**RIP-Match**
Sequence No.	**1**
Action	**Permit**

Access-List	**Checked**
Access-List (ID Chosen)	**50**
Redistribute	**RIP into OSPF**

To configure this example via the Juniper CLI, perform the following steps:

```
set vrouter VR-1025 access-list 50
set vrouter VR-1025 access-list 50 permit ip 192.168.0.0/16 1
set vrouter VR-1025
set vrouter VR-1025 route-map name "RIP-Match" permit 1
set vrouter VR-1025 route-map name "RIP-Match" 1
set match ip 50
exit
exit
set vrouter VR-1025 protocol rip redistribute route-map "RIP-Match" protocol rip
save
```

Redistributing Routes into BGP

If you are running BGP in a multihomed environment which uses iBGP, you will most likely find yourself in need of route redistribution. This is because you may need to announce BGP routes within your autonomous system, as well as advertise routes from within your autonomous system to other systems. BGP is a particularly good example of route redistribution, because you usually need to filter what routes are being advertised and/or imported. You probably wouldn't want to announce the entire Internet BGP routing table into your internal routing structure, nor would you want to announce certain routes out to the Internet via BGP, such as private networks.

Redistributing routes into BGP isn't really any different than doing so with any other protocol. The main difference is the care you take regarding which routes will be imported, and to make sure you don't overwhelm your internal routing infrastructure. In this section, we will cover an example of redistributing OSPF and static routes into BGP, setting attributes in those routes to properly meet the necessary attributes of the route.

Redistributing OSPF into BGP

In this example, we will redistribute OSPF routes into BGP. We will filter out the private IP address ranges, and set the metric, the next hop, and the AS path.

To configure this example via the Juniper WebUI:

1. Creating an access list just like in previous examples. This access list will deny private ranges, but allow anything else. Go to **Network | Routing | Virtual Routers** and click the number hyperlink in the Access List column and row for your VR.

2. Click **New** in the upper right-hand corner.
3. Define a **Name**, a **Sequence No**, an **IP Address/Netmask**, and an **Action**. Do this four times in the example, once for each of the three private address ranges you will deny, and then a fourth time to allow any other range.
4. Create the route map which references the access list just created. Go back to the **Network | Routing | Virtual Routers** page and click the number under the Route Map column, and on the row of your VR.
5. Click the **New** button in the upper right-hand corner of the screen.
6. As in other examples, define a **Name**, **Sequence Number**, and **Action**. Match the routes based upon an **Access-List** and the **Access List ID**. Define set properties to set on any route that matches the map. This will include setting the metric, the AS path, and Next Hop for the route.
7. Apply the route map to the RIP instance by going to **Network | Routing | Virtual Routers**, and then clicking **Edit** next to the VR you would like to edit.
8. Scroll to the bottom of the screen and click the **Edit OSPF Instance** hyperlink.
9. At the top of the screen, click the **Redistribute Rules** hyperlink.
10. In the **Route Map** drop-down menu, select the route map you just configured for the purpose of selecting the routes to import.
11. In the **Protocol** field, choose the protocol you want to redistribute from. In this example, use **OSPF** and **Static**. This will require two different statements.
12. Click **Add** to put this statement into the configuration.

In this example, we configured the following properties:

Virtual Router	**VR-1025**
Access-List	**60**
Sequence No.	**1**
IP / Netmask	**192.168.0.0/16**
Action	**Deny**
Sequence No.	**2**
IP / Netmask	**172.16.0.0/12**
Action	**Deny**
Sequence No.	**3**
IP / Netmask	**10.0.0.0/8**
Action	**Deny**

Sequence No.	**4**
IP / Netmask	**0.0.0.0/0**
Action	**Permit**
Route Map	**BGP-Match**
Sequence No.	**1**
Action	**Permit**
Access-List	**Checked**
Access-List (ID Chosen)	**60**
Set Metric	**500**
AS Path	**1**
Next-Hop	**1.1.1.1**
Origin	**IGP**
Redistribute	**OSPF to BGP**
Redistribute	**Static to BGP**

To configure this example via the command line, input the following:

```
set vrouter VR-1025 access-list 60
set vrouter VR-1025 access-list 60 deny ip 192.168.0.0/16 1
set vrouter VR-1025 access-list 60 deny ip 172.16.0.0/12 2
set vrouter VR-1025 access-list 60 deny ip 10.0.0.0/8 3
set vrouter VR-1025 access-list 60 permit ip 0.0.0.0/0 4
set vrouter VR-1025 as-path-access-list 1 permit "$655"
set vrouter VR-1025
set route-map name "BGP-Match" permit 1
set route-map name "BGP-Match" 1
set match ip 60
set metric 500
set next-hop 1.1.1.1
set as-path 1
set origin igp
exit
exit
set vrouter VR-1025 protocol bgp redistribute route-map "BGP-Match" protocol ospf
set vrouter VR-1025 protocol bgp redistribute route-map "BGP-Match" protocol
static
save
```

Policy-Based Routing

Just when you thought that everything you ever wanted to do could be achieved with RIP, OSPF, or BGP, along comes *policy-based routing* (*PBR*). Traditional routing only applies to layer 3 of the OSI stack. But what if you wanted to make routing decisions based upon, say, whether traffic was HTTP? This may sound odd, but there are actually several real-world situations when this functionality is needed.

Policy-based routing was introduced in Screen OS version 5.4. As mentioned earlier, you can use policy-based routing to make routing decisions based upon layer 4 protocol, source IP address, destination address, source port, destination port, and TOS bits.

Policy-based routing is composed of several different components. At first, this may seem a bit overwhelming, but really, it is a longsighted design. By splitting up different aspects of policy-based routing into different components, you can granularly create different policies which reuse other parts. This technique can save you both time and effort when you need to create policy-based routing on a larger scale.

We will begin this chapter with an overview of the components, and their properties. We will then follow with some examples of how policy-based routing and its components are incorporated together to create customized routing policies.

Components of PBR

PBR is composed of several building blocks which allow you to create different routing policies without having to duplicate many of the steps repeatedly. To build a PBR, start by creating an *access list* to match the routes. You then create a *match group* to aggregate one or more access lists. Next, you create an *action group* that defines a routing decision to take. A *policy* is then created to define the match group (what to match), and an action group (what action to take), and then you put it into a single container. Finally, you apply this policy to either a VR, a zone, and/or an interface. This may sound like a lot of work, but it's actually very simple, and the granular components allow you to put these to good use.

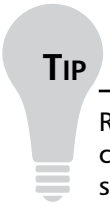
Extended Access Lists

The access list is the core component which actually matches the traffic based upon the settings you configure. The access list used for PBR is called an *extended access list* because you can match upon more than just an IP address—for instance, several other attributes, and in any combination, as mentioned earlier. Remember that, like other types of access lists, these access lists are evaluated from the top down, until a match is found.

Extended Access List Properties

The following list describes the different options you can configure to match traffic within the extended access list.

- **Virtual Router** Access lists are configured at the VR level or below, so you must define what VR it will be a part of.
- **Sequence Number** This allows you to define the placement of this statement within the access list you are creating. Since the access lists are evaluated top down, order is important.
- **Source IP Address / Netmask** If you wish to match traffic on the source IP address, you can configure this component to match a host, or an entire subnet, depending on the subnet mask you specify.
- **Source Port** This allows you to define a specific source port or range to match the traffic on.
- **Destination IP Address / Netmask** This option allows you to match traffic based upon the destination IP address of the traffic. Depending on the subnet mask defined, you can match on a single host, or an entire subnet.
- **Destination Port** You can configure a single destination port, or a range of destination ports to match upon.
- **Protocol** This will match upon the actual protocol of the traffic since, technically, a port does not actually define which protocol travels over it. Without specifying which protocol you want to match, port 80 of TCP would be the same as port 80 UDP. For protocol, you can select **TCP**, **UDP**, **ICMP**, **Any**, or **NULL**.
- **IP-TOS** You can define the traffic to match upon a predetermined IP-TOS value (1 to 255) of the traffic. The access list will check the packet to see if the bits are set, and then act accordingly.



TIP

Remember that you do not have to use all of the attributes in an access list. You can select to match on some attributes of a packet, and not on others. Also, since order is important within an access list, it is usually best to go from most specific to most general when ordering the statements.

Configuring an Extended Access List

Before you can begin to do policy-based routing, you must first configure the Extended Access List which will be used to match the traffic. In this example, we will match traffic that is from the source subnet 192.168.1.0/24, a destination subnet of 10.0.0.0/8, and a destination port 80 protocol TCP.

To configure this example via the Juniper WebUI:

1. Select **Network | Routing | PBR | Access List Ext**.
2. In the upper right-hand corner, select the **VR** you want to create this Access list for and click **New**.
3. Define an **Extended ACL ID** to reference the extended access list by.
4. Each statement in the access list must have a **Sequence No.** it can be identified by, which you must define. This sequence number is important because the access list gets evaluated from the top down.
5. If you want to match traffic based upon the source IP address, you can define the **Source IP Address/Netmask**.
6. You can match a single **Source Port** or a range by entering the values into the fields. If you would like to match a single port, enter it in both fields. If you would like to match a range, enter the low port in the first field, and the high port in the second field.
7. If you would like to match a destination host or subnet, you can do so by specifying the **Destination IP Address/Netmask**.
8. Just like the source port, you can match a single **Destination Port**, or a range or destination ports. Enter the same number in both fields to match a single port, or a low port number in the first field, and a high number in the second to match a range.
9. If you would like to specify what **Protocol** to match the port on, you may specify either **TCP**, **UDP**, **ICMP**, or **Any**. If you do not match the port based upon protocol, you can leave it at **Null**.
10. If you would like to specify an **IP-TOS** value to match on packets, you can do that by defining the appropriate value.
11. Click **OK**.

In this example, we are using the following values:

Virtual Router	**Trust-VR**
Extended ACL ID	**1**
Sequence No.	**1**
Source IP Address / Netmask	**192.168.1.0/24**
Source Port	**None Defined**
Destination IP Address / Netmask	**10.0.0.0/8**
Destination Port	**80**

Protocol	**TCP**
IP-TOS	**None Defined**

To configure this example via the Juniper CLI:

```
set vrouter trust-vr access-list extended 1 src-ip 192.168.1.0/24 dst-ip
10.0.0.0/8 dst-port 80-80 protocol                  tcp entry 1
save
```

To add an additional statement via the WebUI, go into the Extended Access List and click the **Add Seq No.** option. To add an additional statement via the CLI, all you have to do is define the next position with the ***entry <position>*** command at the end of the access-list statement.

Match Groups

A match group allows you to take one or more access lists and group them together to help match traffic. You need to configure a match group, even if only one access list is in it. This is because a match group, not an access list, is used within a *policy* to form the appropriate PBR.

Match Group Properties

There isn't much to a match group, but you should remember that the order which you add access lists in can be important to the decision-making process of the PBR. The properties of a match group are described next:

- **Virtual Router** This is the VR the match group is created within.
- **Match Group Name** This is the name of the match group that's used when you reference it in a policy.
- **Sequence Number** You can have multiple entries in a match group, each of which can match a single access list. Therefore, order is important, as well as why you would define the sequence number of the match group statement.
- **Extended ACL** This is the extended access list that you select for this statement of the match group.

Configuring a Match Group

In this example, we will configure a match group which will apply two different access lists. We will apply this in the Trust-VR.

To configure this example in the Juniper WebUI:

1. Go to **Network | Routing | PBR | Match Group**.
2. In the upper right-hand corner, select the appropriate **VR** from the drop-down menu and click **New**.

3. You must define the **Match Group Name**, which is just a name used to reference the group elsewhere.
4. Define the **Sequence No.** for the match group statement.
5. Specify which **Extended ACL** (access list) you would like to select for this statement.
6. Click **OK**.
7. Add another statement to this match group by clicking the **Add Seq No.** in the match group.
8. Just like the other match group, you must define the **Match Group Name**, **Sequence No.**, and **Extended ACL**. Of course, order is important when you define the sequence number for the match group.
9. Click **OK**.

In this example, we are using the following values:

Virtual Router	**Trust-VR**
Match Group Name	**MatchGroup**
Sequence No.	**1**
Extended ACL	**1**
Sequence No.	**2**
Extended ACL	**2**

To configure this example in the Juniper CLI:

```
set vrouter trust-vr match-group name MatchGroup
set vrouter trust-vr match-group MatchGroup ext-acl 1 match-entry 1
set vrouter trust-vr match-group MatchGroup ext-acl 2 match-entry 2
save
```

Action Groups

When you have determined what traffic you would like to match as part of a policy, you must then decide what to do with this traffic. That's where action groups come in. You use these groups to define how traffic should be routed. Just like match groups, action groups have a sequence to them, and are created within a VR.

Action Group Properties

You have the following options at your disposal for setting a routing decision (or action) for the traffic that's matched.

- **Virtual Router** Since action groups are created within a VR, you must define what VR this action group will be a part of.
- **Action Group Name** This is the name you would like to configure for the access group. The firewall will use this name to reference the object in the policy you create.
- **Sequence Number** You can define multiple sequences for an action group statement. Order is important, so you must define it here.
- **Route To** This is where you can decide on how to route the traffic. You can use one or both of these options to determine how to forward traffic.
- **Next Hop** You would define an IP address for the next hop that the firewall should forward traffic to.
- **Interface** Traffic will be forwarded out this interface.

WARNING

Be careful how you define the *Route To* option so you don't have any unintended consequences. If you only configure the interface forwarding, the traffic will be forwarded only if the interface you define is up. If you configure the forwarding to use a next hop, then that next hop must be reachable through the Destination-Based routing table within that VR. If you choose to use both interface and next hop, then the next hop must be reachable.

Configuring an Action Group

We will configure an action group to perform routing decisions for policy-based routing in this example. We will create this action group in the Trust-VR.

To configure this example in the Juniper WebUI:

1. Go to **Network | Routing | PBR | Action Group**.
2. In the upper right-hand corner, select the appropriate **VR** from the drop-down menu and click **New**.
3. You must define an **Action Group Name** that the action group can be referenced by elsewhere.
4. Define the **Sequence No.** for each statement in the action group.
5. Either define a **Next Hop**, an **Interface**, or both.
6. Click **OK**.

In this example, we are using the following values:

Virtual Router	**Trust-VR**
Action Group Name	**ActionGroup**
Sequence No.	**1**
Next Hop	**Not Defined**
Interface	**Ethernet3**

To configure this example in the Juniper CLI:

```
set vrouter trust-vr action-group name ActionGroup
set vrouter trust-vr action-group ActionGroup next-interface ethernet3 action-
entry 1
save
```

Policies

Once you have built your match groups and action groups, you can then associate them with a policy. A policy essentially puts it all together in one place. It's actually pretty simple, the match group determines if the traffic matches the appropriate policy, and the action group performs the appropriate routing decision.

Policy Properties

Just like action groups and match groups, policies can have multiple statements, and order is important. Be sure you are aware of the underlying configuration of the components of the policy so you can be certain you will not make any adverse decisions. The following list contains the properties of a policy.

- **Virtual Router** This is the VR that the policy is created in. You cannot apply policies created in one VR to another VR, so placement is important.
- **Policy Name** This is the policy name that will be referenced elsewhere in the configuration as an object.
- **Sequence Number** Each statement within a policy must have a sequence number to help determine the order of the policy.
- **Match Group** This is the match group which will evaluate the traffic. If the traffic is matched, then the action group in this policy will be applied to this traffic.
- **Action Group** Once the match has taken place, the action group is used to determine how the traffic is routed on the firewall.

Configuring a Policy

In this example, we will tie a match and action group together to form a policy. Since this policy is composed of match and action groups from a certain VR, this policy group must also be configured in that VR. For this example, we will use the Trust-VR to configure our policy in.

To configure this example in the Juniper WebUI:

1. Select **Network | Routing | PBR | Policy**.
2. Select the **VR** within which you would like to configure the policy. Then, click **New**.
3. Define a **Policy Name** for use in referencing it elsewhere in the configuration.
4. Define the **Sequence No.** which will be used to order the statements in the policy.
5. For each statement, select a **Match Group** from the drop-down menu, as well as an **Action Group** from the respective drop-down menu.
6. Click **OK** to create the policy.

In this example, we are using the following properties:

Virtual Router	**Trust-VR**
Policy Name	**PBR_Policy**
Sequence No.	**1**
Match Group	**MatchGroup**
Action Group	**ActionGroup**

To configure this example in the Juniper CLI:

```
set vrouter trust-vr pbr policy name PBR_Policy
set vrouter trust-vr pbr policy PBR_Policy match-group MatchGroup action-group
ActionGroup 1
save
```

Policy Binding

You probably thought that you were all set once you defined the policy, but there is one last task for you to perform. You must apply the policy to the appropriate place in your configuration. You have three choices: you can apply the policy at the VR level for the VR that it was created under. You can apply the policy at a zone level under the VR that it was created on. And, lastly, you can apply it at the interface level under the VR you created the policy in. Traffic will be matched from a policy configured on the interface first. If it does not match there, it will try to match on the zone level. Lastly, it will try to match on the VR.

Binding a Policy to a VR

If you would like a policy to be evaluated on all traffic routed for a specific VR, you can apply it at the VR level in the configuration. In this example, we will apply the policy we created in the last example (PBR_Policy) to the Trust-VR.

To configure this example via the WebUI:

1. Go to **Network | Routing | PBR | Policy Binding**.
2. If you have not bound any policy to the VR level, you should see an N/A next to the virtual router you would like to apply the VR to. If you have already applied a policy, you will see the policy name specified that you have previously applied. Simply click the **N/A** or policy name to bring up the pop-up window where you select the policy you would like to apply to the VR level.
3. Click **OK**.

In this example, we are using the following settings:

Virtual Router	**Trust-VR**
Policy-Level	**VR Level**
Policy	**PBR_Policy**

To configure this example in the Juniper CLI:

```
set vrouter trust-vr pbr PBR_Policy
save
```

Binding a Policy to a Zone

In this example, we will bind a policy at the zone level, so that any routing decision made on traffic within, or passing through, a zone will be evaluated against the policy. We will apply this policy to the Trust zone within the Trust-VR.

To configure this example via the WebUI:

1. Go to **Network | Routing | PBR | Policy Binding**.
2. If you have not bound any policy to the Zone level, then you should see a N/A next to the **virtual router** which you would like to apply the VR to. You can simply click the **N/A** or the policy name to bring up the popup window which you select the policy that you would like to apply to the VR level.
3. Click **OK**.

In this example, we used the following settings:

Virtual Router	**Trust-VR**
Policy-Level	**Zone Level**

Zone	**Trust**
Policy	**PBR_Policy**

To configure this example in the Juniper CLI:

```
set zone trust pbr PBR_Policy
save
```

Binding a Policy to an Interface

The most specific level you can apply a policy to is the interface level. We will cover binding the policy PBR_Policy to the ethernet1 interface.

To configure this example via the WebUI:

1. Go to **Network | Routing | PBR | Policy Binding**.
2. Select the interface you want to apply the policy to by clicking the **N/A** hyperlink (if you haven't configured a policy on this interface already) or by clicking the hyperlink that is the name of the policy (if you have already configured a policy).
3. In the pop-up window, select the **Policy** from the drop-down menu that you would like applied to the interface.
4. Make sure the **Enable** checkbox is marked.
5. Click **OK**.

In this example, we are using the following configuration:

Virtual Router	**Trust-VR**
Policy-Level	**Interface**
Interface	**Ethernet1**
State	**Enabled**
Policy	**PBR_Policy**

To configure this example via the Juniper CLI:

```
set interface ethernet1 pbr PBR_Policy
    save
```

Summary

Routing is a very powerful and necessary tool in modern networking. No stranger to routing, Juniper has extended their legendary routing support for enterprise and carrier class routing into their firewall platform. In this chapter, we covered the various aspects of routing in the Juniper firewall platform. Juniper has implemented a very powerful routing design which follows the theme of permitting granularity in their firewalls. It all starts with the capability of using virtual routers to segment the routing configuration within the Juniper routers. From there, you are capable of extending routing with routing protocols such as RIP, OSPF, BGP, and static routing.

In this chapter, we focused not only upon how to implement the protocols such as RIP, OSPF, and BGP, but also the differences between them, as well as their strengths and weaknesses. While RIP might be a good choice for a small network supporting legacy equipment, it would hardly scale for large networks like OSPF. Of course, you must use BGP if you would like to participate in Internet routing between other autonomous systems, but there are many things you should consider before doing so. We also focused on securing your routing infrastructure in this chapter. It seems appropriate to do so in a security device like a firewall, but routing security is something that is often overlooked even in very security-conscious organizations.

Configuring routing protocols in a network can be a difficult task. Juniper recognizes this and has given a lot of thought to providing you, the administrator, with simple yet effective mechanisms to administer your firewall. These include thorough support for different routing protocols, logical routing filters such as route maps and access lists, as well as route redistribution between protocols. Of course, configuring routing protocols might be the easy part, so Juniper has also provided you with an extensive array of commands to get accurate and concise information from the firewall.

Juniper firewalls have been designed to be feature rich, and highly scalable in order to meet the challenges of modern day networking and security. You will find that Juniper has not left much to be desired in their products. Whether it is support for source-based or policy-based routing, multicast, BGP, route redistribution or stateful packet inspection, Juniper has managed to fit all of this functionality into a single device.

Solutions Fast Track

Virtual Routers

- ☑ Juniper firewalls support a concept called virtual routers. Each VR is its own independent router, with its own routing table and configuration.
- ☑ By default, the Juniper firewalls come predefined with a Trust-VR and an Untrust-VR, which, though you can edit their properties, you cannot delete.

- ☑ You can create a custom VR which can have the same capabilities as a predefined one. This is often useful to help segment your internal routing domain so routes are not leaked to other routers.
- ☑ In addition to destination-based routing, Juniper firewalls support source-based routing, source interface–based routing, and multicast.

Static Routing

- ☑ Static routes allow you to manually define routes within a VR.
- ☑ You may redistribute static routes into other routing protocols, but not vice versa.

Routing Information Protocol

- ☑ RIP is a distance vector routing protocol which shares routing information between its neighbors to help build the network topology table.
- ☑ There are currently two IPv4 RIP versions: Version 1 and Version 2. The main difference between v1 and v2 is that v2 supports subnet masks and authentication.
- ☑ RIP uses a metric called hops to determine the cost of a route. A hop is viewed as a router which the traffic must pass through. If there are three routers that the traffic must pass through, then you would have a route cost of three hops. The maximum number of hops RIP will support is 15.
- ☑ RIP is susceptible to routing loops, and uses mechanisms such as count to infinity, split horizon, and poison reverse to prevent routing loops from forming

Open Shortest Path First

- ☑ OSPF is a link state routing protocol, capable of efficiently segmenting the routing domain into separate areas to help reduce the load on network routers.
- ☑ OSPF has different types of areas based upon the topology that the area supports. The different areas are known as backbone, stub, not-so-stubby area, and totally-stubby area.
- ☑ All areas must be connected to the backbone via an ABR, the one caveat is that you can create a virtual link to create a virtual connection from an area that doesn't directly connect to the backbone.
- ☑ OSPF can be redistributed into other protocols, and vice versa.

Border Gateway Protocol

- ☑ BGPv4 is the routing protocol that is supported across the Internet. It allows organizations known as autonomous systems to advertise and route traffic between each other.
- ☑ BGP is a path vector protocol, which offers several attributes to a route to help administrators provide additional information that can be used to route traffic.
- ☑ When you have an environment that must route and advertise BGP traffic through it, you may want to implement iBGP. All routers in an iBGP domain must either be fully meshed, or support router reflectors or confederations to simulate the fully meshed environment.
- ☑ BGP can be redistributed into other protocols, and vice versa, but particular attention should be paid regarding what routes are being advertised into other protocols so as to not overwhelm your routing infrastructure.

Route Redistribution

- ☑ Route redistribution allows routes from one routing protocol to be advertised in another routing protocol.
- ☑ Route redistribution is common in environments that support legacy equipment, proprietary protocols, or that have merged with other networks using a different routing protocol.
- ☑ You must be careful when redistributing routes into another protocol because you may inadvertently create routing loops or suboptimal routes. Additionally, different routing protocols use different metrics, so you must take this into account when advertising one protocol into another.
- ☑ Route redistribution is configured on the router that would like to take routes out of its routing table, as advertised from one protocol, and advertise them in another protocol.

Policy-Based Routing

- ☑ Policy-based routing is a new feature in version 5.4 of the Screen OS.
- ☑ PBR enables you to make routing decisions based upon specific information such as ports, protocols, and TOS values. This is outside of the traditional routing mechanisms, which route on the destination of traffic.

- ☑ PBRs use a combination of extended access lists, match groups, and action groups to form a policy.
- ☑ PBR policies can be applied at the VR, Zone, and Interface level.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: When would I want to use source-based routing (SBIR) over destination-based routing?

A: Destination-based routing is definitely the most widely implemented form or routing. Many administrators feel it is a more logical choice than routing based upon the source/interface because you are trying to forward traffic based upon where it is going, not on where it came from. But this isn't always the case, and so Juniper has provided support for source-based and source interface–based routing should you need to make routing decisions with these methods.

Q: When should I use a dynamic routing protocol instead of just using static routing?

A: Networks that are very small have a limited number of networks, and have only single paths between networks that are good candidates for static routing. For example, if you have a central site with three branch offices forming a hub-and-spoke configuration, it would make sense to just point each branch to the central site statically, which could route the traffic to the appropriate branch. In such a network, where there is no redundancy, you don't have to worry about being able to fail traffic over to a separate link should one link fail. Those are common reasons for going with static routing instead of dynamic.

Q: RIP seems pretty limited. When would I want to use it over another routing protocol?

A: All things being equal, OSPF is usually a better choice for dynamic routing than RIP. However, sometimes things are *not* equal and you may need to implement RIP onto your network. A good example of this is when you have to support legacy equipment that isn't capable of supporting OSPF.

Q: Does Juniper firewalls support EIGRP, and if not, how can I fit the firewall into my EIGRP network?

A: Juniper does not support EIGRP because it is a Cisco proprietary protocol. If you are running EIGRP in your network, and you cannot change to an open protocol like OSPF, your best bet is to do redistribution. The redistribution wouldn't take place on your Juniper firewall, but rather, you would need to redistribute EIGRP routes on the Cisco device into a protocol such as OSPF, which would be understood by the Juniper firewall.

Q: Some of these commands are very long to type on the CLI. Is there any shortcut?

A: In many situations on the CLI, you can save keystrokes by "entering" parts of the configuration. For instance, you can enter a VR by typing *set vrouter <vrouternam*e>. The CLI prompt will change to *<hostname>(<vrouternam*e>)->. From that point, you will not have to type the *set vrouter <vrouternam*e> before every command. The best way to discover where you can enter a mode is by using the context-sensitive help with the "?" after statements in a command. If you see a *<return>* at a point in a command that has a lot of other options, it might be a candidate for such a short cut. You can back out of a configuration mode by typing the *exit* command for each level you want to back out.

Q: Can I just use policy-based routing instead of configuring static routes?

A: This practice is not recommended. You should reserve policy-based routing for making routing decisions on nontraditional parts of TCP/IP traffic such as ports, TOS, and protocols, rather than comprehensive routing for your device.

Q: When would I want to use ECMP?

A: Equal Cost Load Balancing is most commonly used to accommodate environments where you might want to load-balance over multiple links.

Q: Why can't I filter routes advertised by OSPF to other OSPF routers, like I can with RIP and BGP (neighbors)?

A: Since OSPF is a link state protocol, every router announces the state of each link between every device. The routers take all of this information and use it to build the routing table, so routes themselves are not advertised to neighbors, but rather the state of links are. If you did not announce the state of links it would have other implications for the overall network topology map OSPF needs to build its link state database. Therefore, you must announce the link states, and you cannot filter what routes you want to advertise via OSPF.

Q: Why would I want to reject learning a default route from a routing protocol, and why would I not want to advertise it?

A: Depending on the architecture of your network, you may not want to import a default route learned by a routing protocol, because it may cause traffic to be routed in a manner which you did not expect. For instance, perhaps you don't want a default route at all, and you have not defined one statically since you only want to route to destinations within your network, then importing a default route from a routing protocol would break this functionality. On the other hand, you may not want to advertise a default route, because you don't want a downstream router to use your router as a default router. This could be because you don't want the extra traffic load, or even because you do not connect to all other networks, so the traffic would be dropped by your router anyways.

Chapter 8

Address Translation

Solutions in this chapter:

- **Overview of Address Translation**
- **Juniper NAT Overview**
- **Juniper Packet Flow**
- **Source NAT**
- **Destination NAT**

- ☑ **Summary**
- ☑ **Solutions Fast Track**
- ☑ **Frequently Asked Questions**

Introduction

With the explosive growth of the Internet in the last decade, the number of available IP addresses has become scarce. To help ease the burden for unique IP addresses, three network ranges were created to be used as private addresses not routable on the Internet. In order to support these nonroutable addresses that were not globally unique, they have to be translated into addresses that *are* globally unique and routable. This technique is called Network Address Translation, or NAT for short.

Juniper has engineered several methods of performing NAT on their firewalls. They have created mechanisms for performing NAT in policies as well as on the interface level. You will no doubt find a good solution to implementing NAT in your network.

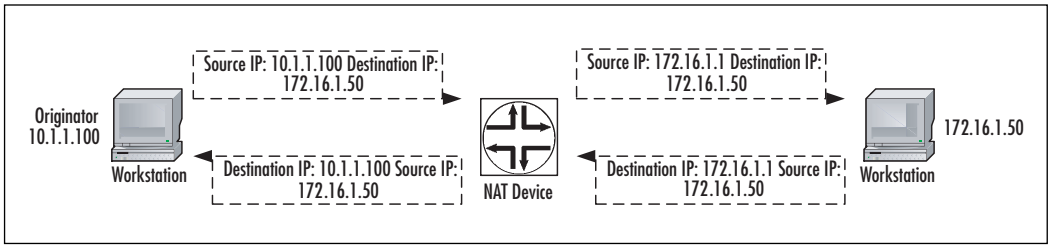
In this chapter, we will begin with an overview of NAT itself. We will then discuss the various concepts and terminology of NAT on the Juniper firewalls, reinforcing those concepts with real-world examples to help get you on your way with configuring NAT on your firewall.

Overview of Address Translation

NAT has provided a way to virtually expand the number of hosts that can connect to the Internet. Since IPv4 has a limited number of IP addresses it can globally support (2^32,) NAT has allowed network administrators to buy some time with IPv4. Essentially, NAT allows you to masquerade one IPv4 address with another. The address translation does not have to be strictly one-to-one; it can also be many-to-one, one-to-many, or many-to-many on the Juniper firewall. The method you can choose depends on the type of NAT that you have to perform. For instance, you can hide all of your internal (private) hosts behind a single address with many-to-one NAT. You could also hide a single private IP address behind a single public IP address. In the former case, the ability to hide many IP addresses behind a single address has allowed many more devices to connect to the Internet than would be possible without NAT. Since IP address space also costs money to obtain (and is pretty limited with IPv4), performing NAT can save you money (as well as headaches) trying to acquire additional IP addresses.

As mentioned earlier in this chapter, NAT replaces one IP address with another. This functionality is completely transparent to users and applications alike. For example, Figure 8.1 shows a host on network 10.1.1.*x*/24 traversing through a NAT device. The NAT device then translates the source packet coming from host 10.1.1.100 and going to address 172.16.1.1, which then communicates with host 72.16.1.50. This method is called *source NAT*.

Figure 8.1 All Egress Traffic from 10.1.1.*x* Network NATs from Source 172.16.1.1

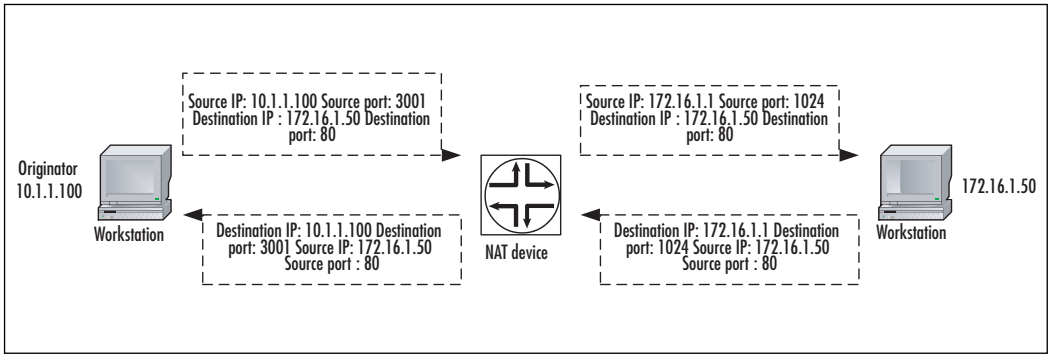


Port Address Translation

In order to accommodate translating multiple addresses into a single address, the firewall must be able to track what traffic came from which host, since once the packet gets translated the IP address looks like all other traffic which has been translated. This technique is called Port Address Translation, or PAT for short. PAT can run over 64,000 sessions off of one IP address. Source PAT starts at port 1024 and above; therefore, it is possible to scale up to 64,512 ports (65,535 is the max number that a TCP/UDP port can reach – 1023 = 64,512) that can be allocated for one NAT'd IP address. The reason that PAT ports start at 1024 and above is because ports 0 through 1023 are reserved and primarily used for well-known services (for example, TCP port 23 is for Telnet, TCP port 22 is for Secure Shell [SSH], and TCP port 80 is for Hypertext Transfer Protocol [HTTP]). (See RFC 3022 for more information on PAT.)

An example of source PAT is illustrated in Figure 8.2. The image shows the NAT device performing a source NAT and a source PAT from the originator (10.1.1.100). Besides translating the source IP, such as that shown in Figure 8.1, the NAT device will also translate the original source port to a random source port, which usually starts at 1024 and above. Notice that the return packet response from 172.16.1.50 is translated back to port 3001.

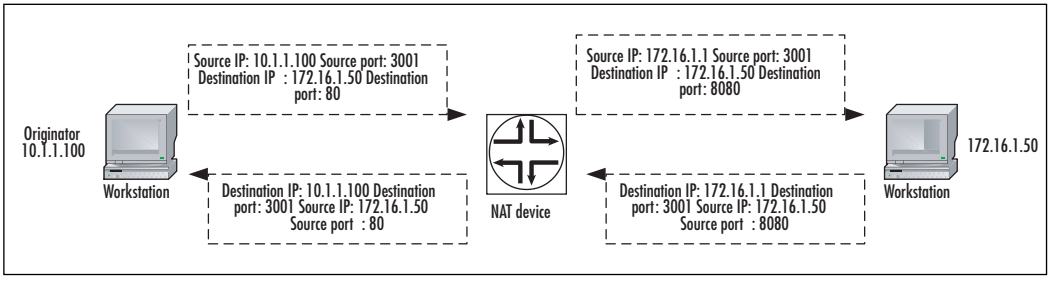
Figure 8.2 All Egress Traffic from the 10.1.1.*x* Network NATs from 172.16.1.1 and PAT with a Random Source Port



The reason why you need to perform PAT to be able to hide multiple addresses is because you need to have some variable for the NAT device to be able to map sessions with. Since the NAT'd address is always going to be the same (in Figure 8.2 it would be 172.16.1.1), and the destination IP address and destination port cannot change, source port makes the right choice for PAT. This is because most applications don't care about what the source port is of a connection that gets assigned dynamically by the host. Therefore, the NAT device can keep a table for all of the sessions by NATing the source address, as well as the port, and keeping track of it. So when the traffic returns from the destination with the translated port as the destination port (1024 in Figure 8.2) the NAT device looks up that mapping in the table, and knows that the new destination must be 10.1.1.100, port 3001, and NATs it again accordingly.

The opposite of source PAT is destination PAT. This technique is used to change the destination port address to another port address, and possibly host when it passes through a NAT device. Figure 8.3 shows an example of a destination PAT function. Traffic destined for port 80 from the originator would be translated to a different port. Notice that the return packet response from 172.16.1.50 is translated back to port 80. The NAT device uses the same session matching technique as it does with the source PAT, but rather, it uses the destination port instead to map the session.

Figure 8.3 All Egress Traffic from 10.1.1.x Network NATs from 172.16.1.1 and Destination PAT from Port 80 to Port 8080



Advantages of Address Translation

Because of the tremendous growth of the Internet in the past decade, there were not enough IPv4 addresses. NAT was developed to provide an immediate solution to this depletion. Request for Comment (RFC) 1631 was written in 1994 as the short-term solution to address the problem—the long-term solution was IPv6.

Other ways that NAT is useful is

- **Private Address Usage on a Routable Network** A NAT device can translate an existing nonpublic routable subnet to a public routable address(es). Most companies use RFC 1918 addresses for their corporate networks because it helps con-

serve their routable Internet Assigned Numbers Authority (IANA) public addresses. RFC 1918 addresses are

```
10.0.0.0 to 10.255.255.255 (10/8 prefix)
172.16.0.0 to 172.31.255.255 (172.16/12 prefix)
192.168.0.0 to 192.168.255.255 (192.168/16 prefix)
```

- **Cost** By hiding many IP addresses behind one or a few, you do not need to pay for as many external IP addresses, thus providing cost savings.
- **Addresses Overlapping Networks** NAT can provide a masquerade of different networks when two duplicate networks must be merged.
- **Helps Maintain a Cohesive Network** Provides a method of maintaining one cohesive network when needed to communicate with different extranets.

Both the source and destination packets can be translated using Juniper's NAT functionality. Additionally, you can translate traffic based upon the direction that it is headed, either inbound or outbound. Translation can take place on the IP addresses and Port numbers of IP protocols, including TCP, UDP, and ICMP.

Disadvantages of Address Translation

When using address translation, certain scenarios come with certain concerns. The following is a list of the most common issues found when using NAT:

- **Secure Internet Protocol (IPSec) usage through a NAT device** See Chapter 12 for more information on why NAT causes IPSec to break. This is because the NAT operation alters the contents of the packet (for example, source port for source PAT). Since a one-way hash is taken of the packet before it leaves and is appended to the packet, when NAT is performed on the packet it will alter the contents of the payload. Thus, the one-way hash performed on the destination end will differ from the one created in the payload. Two workarounds to this exist in a NAT environment: (1)create a one-to-one NAT and disable PAT, and (2) use NAT Traversal.
- **Protocols that perform their own dynamic port allocation** For example: passive File Transfer Protocol (FTP), Sun Remote Procedure Call (RPC), MS-RPC, and so on. The problem is that when one server instructs the client to connect over a different port, it will fail, because that new session has not been created in the firewall. Workarounds are available, however. Most firewalls implement a feature called Application Level Gateway (ALG) to address applications that require dynamic port openings. Essentially, the firewall will use ALG to read the control traffic and learn what port the server is requesting the client to connect through,

and then it will dynamically open that port for the client when it sends the traffic through. Juniper supports ALG for several protocols, as discussed in Chapter 2.

- **Legacy application or custom application requires that the original packet information be maintained** This varies from requiring the network address to the port to remain the same. This is usually because some of the packet information is actually stored inside the payload, besides being in the layer 3 and layer 4 levels. Therefore, the application will decapsulate the packet, and see that its addresses and port values do not match that which is contained in the payload. Some firewalls can use ALG to alter the internal contents of the payload so that it will match the NAT'd information. Sometimes, disabling NAT, PAT, or both will address this issue. It is generally recommended to disable PAT first, because the majority of these applications relate to restrictive ports.

Juniper NAT Overview

Juniper provides you with the capability of performing source and destination NAT, as well as source and destination PAT in several different ways. In this section, we will discuss the different NAT mechanisms available in ScreenOS 5.4. It is important to note that some of the older features such as Mapped IP (MIP) and Virtual IP (VIP) can also be performed in the newer ScreenOS releases. The following NAT features are covered in detail in subsequent sections.

- **Source NAT** Provides address translation on the source IP address. Source PAT is another functionality that may be performed along with source NAT. Source PAT provides address translation on the source port. Juniper supports several methods of Source NAT, such as:
  - **Interface-Based Source NAT** This feature provides the capability to NAT traffic as it passes through a defined interface. The traffic gets the source IP address of the egress interface of which it leaves. This mechanism is on by default on many interfaces on the Juniper firewall. It can often be disabled by putting the interface into "route mode" instead of "NAT mode."
  - **MIP** This NAT technique provides a static one-to-one address translation, meaning that a certain destination or source address will be NAT'd to a specific address which is not shared by other hosts. This technique does not alter the source or destination ports, because it doesn't have to since only one machine is operating with this NAT. As mentioned already, MIPs can be used for either source or destination NAT.
  - **Policy-Based Source NAT** The technique utilized here is similar to interface-based NAT, but is configured on a per-policy basis. This provides you with much greater granularity since you can configure NAT to match certain

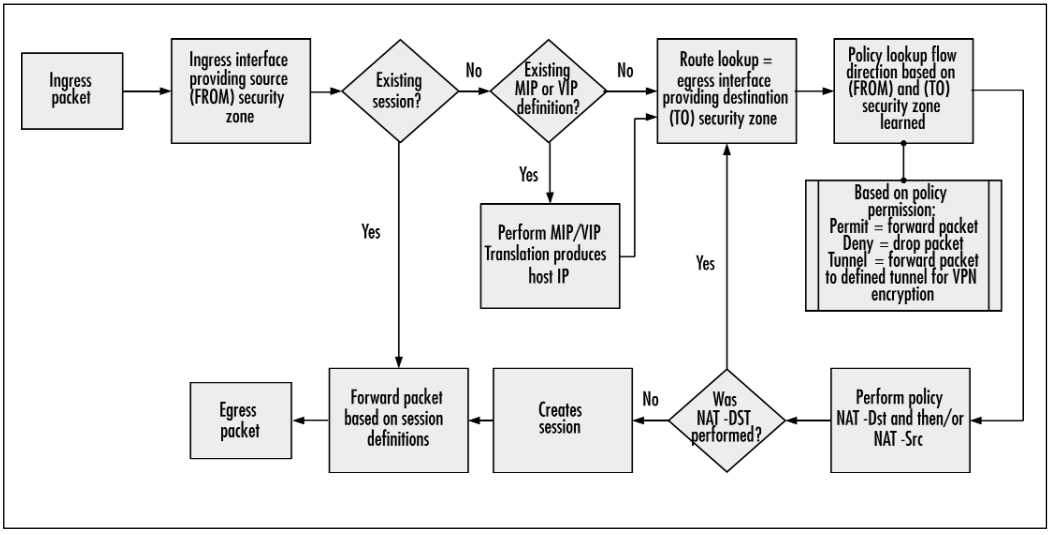
conditions met in the policy (source IP address, destination IP address, and service). You can also specify different NAT addresses to be used in different policies.

- **Destination NAT** Juniper provides you with a very similar set of NAT capabilities for the destination side, while it does for the source side. The following NAT techniques allow you to perform various NAT and PAT translations:
  - **VIP** A VIP allows you to translate both the destination IP address and destination port address of traffic. VIPs are created at the interface level, and also must be allowed by policy (although they differ from policy-based destination NAT).
  - **MIP** As mentioned before, MIPs can be used for both source- and destination-based one-to-one address translation.
  - **Policy-Based Destination NAT** This is similar in functionality to policy-based source NAT except that it performs address translation on the destination IP or destination port on a per-firewall rule definition.

Juniper Packet Flow

This section highlights the address translation portion of the Juniper packet flow. Understanding how Juniper handles a packet flow provides a good base to understanding how address translations are triggered and also makes troubleshooting and debugging a problem much easier. Figure 8.4 shows a high-level overview of how a Juniper firewall handles packets flowing into their devices.

Figure 8.4 NetScreen Packet Flow



The process steps are as follows:

1. Based on the arriving ingress packet, the Juniper device notes the incoming interface and the security zone bound to that interface. (For the purposes of this book, the ingress security zone is considered the FROM zone.) The interface can be a physical interface, such as an Ethernet interface, or a virtual interface, such as a sub-interface, a VPN tunnel interface, or a VPN tunnel zone. At this point, the packet screening functions are performed. The screening function detects any anomalous traffic behavior such as Denial-of-Service (DoS) attacks. The screen options are configurable at the security zone level.
2. The firewall then checks to see if the session exists. If it does, then the firewall will forward the packet based on the session definition. If the session does not exist, then the firewall will check to see whether a MIP or VIP entry exists. If one does exist, it will then perform a MIP or VIP translation.
3. The route lookup is performed next. Based on the destination packet IP address, the route lookup determines which egress interface the packet will eventually leave from. When you know the egress interface, you will also know the egress security zone (remember, the interfaces are bound to a security zone). For the purposes of this book, the egress security zone is considered the TO zone.
4. Now that you know the FROM and TO security zones, you can apply them to a policy lookup. At a minimum, a policy will permit, deny, or push the packet through a VPN tunnel. Other miscellaneous operations can also be performed, such as traffic shaping, deep inspection, authentication, logging, counters, antivirus, and threshold alarms. If address translations are defined for either the source (NAT-Src) or the destination (NAT-Dst), those functions are performed here. If NAT-Dst is performed, another route lookup is required since the destination IP address has changed and may require a different route than the original traffic.
5. Once the packet has been processed by a policy, a session is created and the packet is forwarded to the egress interface based upon routing.

Source NAT

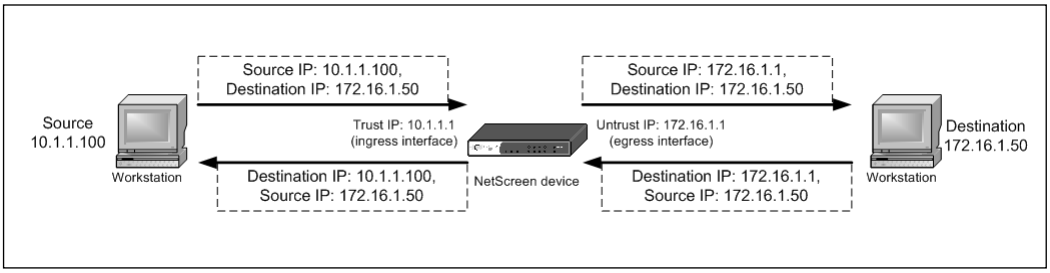
Source NAT is the most widely deployed method of address translation provided by vendors. It offers the ability to translate a source IP address to another IP address (as illustrated in Figure 8.1). The Juniper firewalls enable source NAT by default on the interfaces in the Trust security zone (see Figure 8.5, the *Interface Mode* button). This functionality can be changed by putting the interface in *Route Mode*, which does not perform any NAT on the traffic as it leaves the egress interface, but rather just routes it.

Figure 8.5 Web UI Screen Shot of Interface-Based Source NAT

Interface-Based Source Translation

Interface-based source translation provides the ability to launch NAT off of a physical interface or a logical interface (for example, a sub-interface). The interface with NAT mode enabled will perform NAT with port translation by default for the traffic passing through it. The source IP address used for translation will be the egress interface. In other words, whichever interface the packet exits from is the interface IP address that will be used as the translated source IP address. Figure 8.6 shows what address translation looks like for a packet sourcing from a host behind Ethernet1 on the Trust zone with NAT enabled.

Figure 8.6 Source 10.1.1.100 Sources NAT from the Egress Interface Untrust



The egress interface used is determined via the Juniper packet flow (see Figure 8.4). The key area is the route table lookup for the destination.

Interface-Based NAT Properties

Interface-based NAT has very few settings, unlike other NAT types. Essentially, all you need to do is set the **Interface Mode** to **NAT** instead of **Route** on the ingress interface. When traffic passes out the egress interface, it will get NAT'd to the IP address of that interface. In this respect, you do not have any options for what the IP address will be NAT'd to. If you need to have it NAT'd to a different IP address, you will have to set it to use a different form of NAT.

Example of Interface-Based NAT

In this example, we will configure NAT on the Ethernet2 interface, and Route Mode on the Ethernet4 Interface, so that any traffic passing from the Ethernet2 to Ethernet4 interface will be NAT'd to the untrust IP address. This is a common scenario when you have internal/DMZ machines that need to be hidden behind a public IP address that is routable on the Internet. We will assume that you already have the IP addressing configured on this interface, as described in previous chapters.

To configure this example via the Juniper WebUI:

1. Go to **Network | Interface** and click the **Edit** hyperlink next to the ingress interface.
2. Make sure the **Interface Mode** is set to NAT.
3. Click **OK**.
4. Go to **Network | Interface** and click the **Edit** hyperlink next to the egress interface.
5. Make sure this **Interface Mode** is set to **Route**.
6. Click **OK**.

In this example, we will use the following settings:

Ingress Interface	
Interface	**Ethernet2**
Interface Mode	**NAT**
Egress Interface	
Interface	**Ethernet4**
Interface Mode	**Route**

To configure this example in the Juniper CLI:

```
set interface ethernet2 nat
```

```
set interface ethernet4 nat
save
```

NOTE

Prior to ScreenOS 5.0, communication from a host on the Untrust zone to a host on the Trust zone with the interface set for NAT was not possible unless a MIP or VIP was defined. With the current ScreenOS release, this communication is possible as long as there is a firewall rule defined to allow it.

TIP

The following are some rules and limitations on interface-based NAT:

1. The egress interface *must* be bound to the Untrust zone.
2. When a user-defined zone is bound to the ingress interface with NAT enabled, that user-defined security zone must be defined on a different virtual router than the Untrust zone.
3. Interface-based NAT will not work between the Trust zone and a user-defined defined zone.
4. Interface-based NAT does not work on an interface bonded to the Untrust zone, even though it can be enabled.

MIP

MIP provides the ability to perform a one-to-one mapping translation, which is referred to as *static NAT*. This setting ensures that a host gets the same NAT every time traffic traverses the firewall, whether it's ingress or egress traffic. A MIP definition only performs NAT (no PAT), thus the IP address changes but the protocol ports remain the same. MIP(s) are defined on an interface that can be the physical Ethernet interface, a sub-interface, a tunnel interface, or a loop-back interface. Once a MIP is defined, a firewall rule is needed to allow access to the MIP.

Besides one-to-one mapping, a MIP can also be created to a subnet. In a MIP-to-subnet definition, it is important to define the subnet mask properly. The host range used for the MIP should not be used elsewhere on the Juniper device.

All MIP definitions are placed within a global zone no matter which security zone originally defined the MIP. Once a MIP has been defined, a firewall rule must be set up to allow for traffic destined for the MIP address. Within the firewall rule creation, the destination MIP selection can be from a global zone or the zone the MIP address was originally defined in.

WARNING

You must create a MIP address that is within the subnet of the interface which the MIP is applied to. For instance, if you have an interface with an IP address 10.1.1.0/24, a valid MIP would be 10.1.1.50, while 10.1.2.50 would not be valid. This is because traffic must be routed to this interface, and the firewall must perform a proxy-arp for the MIP address on its interface. This is only possible when the MIP is in the same subnet as the interface itself.

MIP Properties

When configuring a MIP, the following properties must be configured to create it.

- **Mapped IP** This is the external IP address that is applied on the interface.
- **Host IP** This is the internal IP address of the host.
- **Netmask** You can configure MIPs to either NAT one IP address to another, or create a one-to-one mapping for a whole subnet of external IP addresses to internal IP addresses. You define this with the netmask. For instance, if you define a netmask of 255.255.255.255 with a mapped IP 192.168.1.50, host IP 10.1.1.50, then the mapped IP will NAT to the host IP. If you defined a netmask of 255.255.255.0, with a mapped IP of 192.168.1.0 and host IP of 10.1.1.0, the mapped IP would create a one-to-one mapping for all of those IP addresses in the subnet so 192.168.1.1 -> 10.1.1.1, 192.168.1.2 -> 10.1.1.2…192.168.1.253->10.1.1.253, 192.168.1.254->10.1.1.254.
- **Host Virtual Router Name** You must define what VR the host IP address belongs to. In other words, you need to see what interface the host is located through, what zone that interface is bound to, and then determine what the VR is for that host.

MIP Limitations

Only a limited number of MIPs can be performed on a NetScreen firewall. The matrix in Table 8.1 shows the MIP capacity, as of ScreenOS 5.4. These numbers are the same for both basic and advanced license models, and are presented according to the datasheets published by Juniper:

Table 8.1 MIP Capacity Matrix

Product Name	MIP Capacity
NetScreen HSC	5
NetScreen 5XT	32
NetScreen 5GT	300
Juniper SSG 5/20	Not Published
NetScreen 25	500
NetScreen 50	500
Juniper SSG 140	Not Published
NetScreen 204/208	4000
NetScreen 500	4096
Juniper SSG 520	1500
Juniper SSG 550	6000
NetScreen ISG 1000	4096
NetScreen ISG 2000	8192
NetScreen 5200/5400(Gen 1 and 2)	10,000

MIP Scenarios

This section covers some real-world scenarios where MIP is useful. Each scenario contains the steps needed to configure the example in the Web UI, example values and configurations, commands to perform the example in the CLI, and a diagram representing the physical equipment and NAT performed. It is assumed that you have the necessary basic settings predefined on your firewall, such as security zone definitions on interfaces, IP address definitions on interfaces, route definitions, and so on.

Scenario 1

The following example shows a typical MIP scenario. MIP is defined for you to access a Web server located on your private network from the Internet.

The following Web UI configuration example is illustrated in Figure 8.7:

1. To define a MIP, go to **Network | Interface** and click the **Edit** hyperlink next to the interface which you would like to edit. Select **MIP** at the top of the screen. Then click **New** in the upper right-hand corner.
2. Define a **Mapped IP** which represents the public IP address that will be applied to this interface.

3. Define the internal **Host IP** address for the host on the internal interface of the firewall.
4. Define the **Netmask** according to whether you are configuring this as a single host translation, or configuring this for multiple IP addresses.
5. The last step you need to perform in the MIP configuration is to specify what **VR** the host machine is routed through on the internal side.
6. Click **OK**.
7. Define a policy which enforces the MIP. To do so, go to **Policies** and select the appropriate **FROM** and **TO Zones**. Click **New** in the upper right-hand corner of the screen to create a new rule in the rulebase according to the zones selected.
8. Assuming that this example is for incoming NAT from the public Internet to a machine within the network, you would define the **Source Address** either as **Any** or as specific hosts or subnets on the Internet.
9. For the destination IP address, you would select the MIP address from the drop-down menu. You can identify this address by the form **MIP (<***public address***>)**.
10. Specify the **Service** and the **Action** as Permit. You can also specify any additional policy options such as logging, deep inspection, AV, and so on.
11. Click **OK**.

In this example, we will use the following settings:

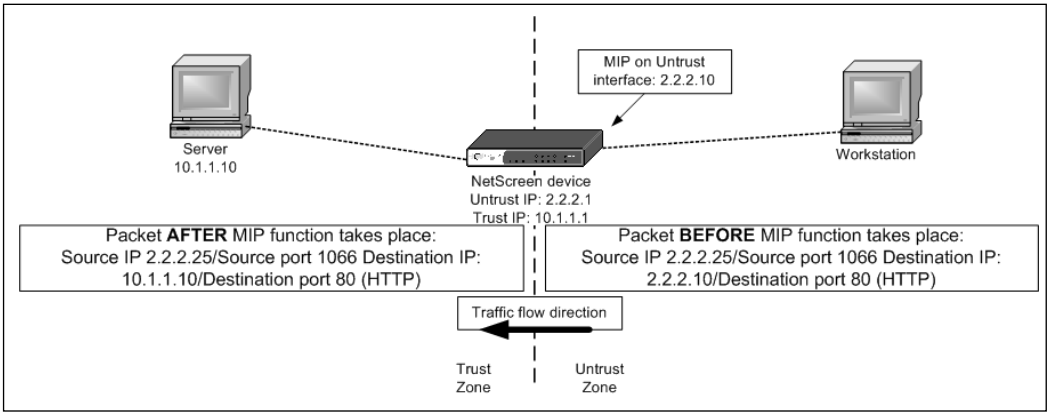
MIP	
Mapped IP:	**2.2.2.10**
Host IP:	**10.1.1.10**
Netmask	**255.255.255.255**
Host Virtual Router	**Trust-VR**

Policy	
From Zone	**Untrust**
To Zone	**Trust**
Source Address	**Any**
Destination Address	**MIP(2.2.2.10)**
Service	**HTTP**
Action	**Permit**
Logging	**Enabled**

To configure this example in the Juniper CLI:

```
set interface "untrust" mip 2.2.2.10 host 10.1.1.10 netmask 255.255.255.255 vr
"trust-vr"
set policy from Untrust to Trust any MIP(2.2.2.10) http permit log
save
```

Figure 8.7 NetScreen with MIP Definition to a Host on the RFC 1918 Network



Scenario 2

The following example shows how a host defined within a MIP definition always uses the translated MIP address when originating traffic.

The configuration for this example is not very complicated. In fact, it can be as simple as defining an outgoing "permit all" rule. Note that there is no incoming rule (Untrust to Trust) because the 10.1.1.10 server is *initiating* the traffic. Because NetScreen is a stateful firewall, the response packet back is handled based on the session that was originally created from this server (see Figure 8.4, step 2).

The following Web UI configuration example is illustrated in Figure 8.8:

1. To define the MIP, choose **Network | Interface** and click the **Edit** hyperlink next to the interface you want to apply the MIP to.
2. Click the **MIP** hyperlink at the top of the page, and then click the **New** button in the upper right-hand corner.
3. You must define the **Mapped IP,** which is the external IP address, as well as the **Netmask** to define the host/subnet mask range.
4. Specify the **Host IP** address of the internal host, as well as the **VR** that is responsible for routing traffic to/from the interface that the internal host is on.
5. Click **OK**.
6. Define a policy to allow the traffic to pass. To do so, go to **Policies**, and then define the **From** and **To** zones. Then, click **New**.
7. Define a **source IP address**. This does not have to be the specific MIP address of the internal host. It could be a network or any, so long as the source address of the host falls within the range that you define for the policy to match.

8. Define a destination IP address accordingly.
9. Specify the Service along with the optional policy choices, such as logging, antivirus, and so on.
10. Click OK.

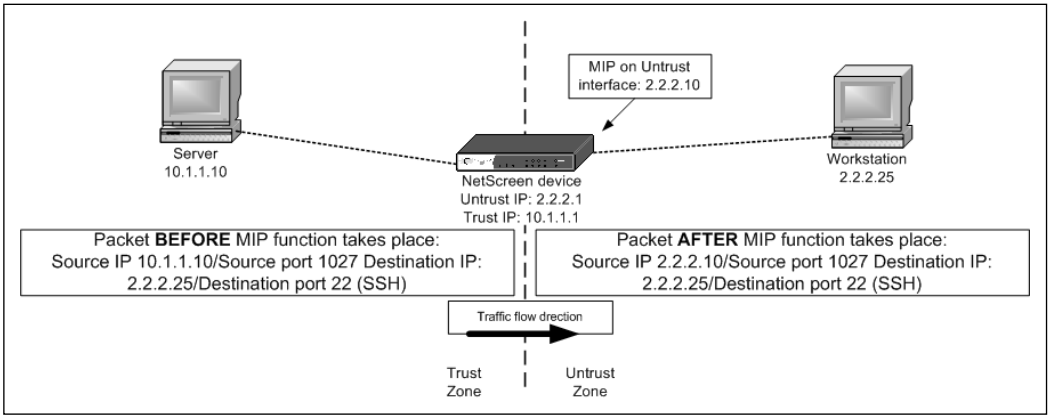
In this example, we used the following settings:

MIP	
Mapped IP	**2.2.2.10**
Netmask	**255.255.255.255**
Host IP Address	**10.1.1.10**
Host Virtual Router	**Trust-VR**
Policy	
From Zone	**Trust**
To Zone	**Untrust**
Source Address	**Any**
Destination Address	**Any**
Service	**Any**
Action	**Permit**
Logging	**Enabled**

To configure this example from the Juniper CLI:

```
set interface "untrust" mip 2.2.2.10 host 10.1.1.10 netmask 255.255.255.255 vr
"trust-vr"
set policy from Trust to Untrust any any any permit log
save
```

Figure 8.8 MIP Outbound Traffic from Internal Host



Again, you can always create a more granular rule for this scenario. Change the source address to 10.1.1.10/32 and the destination address to 2.2.2.25/32, and then change the servers to SSH. Notice that there is no need to choose the MIP (2.2.2.10) for either the source or destination, because the translation occurs automatically (see Figure 8.4, step 2).

Scenario 3

There may be certain scenarios where you have to create a route to reach the original host once a MIP translation occurs. Going back to Juniper packet flow, a MIP translation occurs before a route lookup and a policy lookup. If the MIP translation to the original host route is not defined within the routing table(s), the packet either gets dropped or is sent to a default route, assuming one exists.

The following example scenario shows the need to create a route in order for the MIP translation to work. Figure 8.9 shows a diagram of MIP host 172.16.1.10 behind a different segment from the Trust side of the Juniper firewall.

To reach the server on the 172.16.1.*x* network segment after MIP translation occurs, a static route is needed on the Juniper device. This route would consist of the following:

```
Destination: 172.16.1.0/24
Interface: Trust (Ethernet1)
Gateway: 10.1.1.254
```

To configure this example via the Juniper WebUI:

1. Configure the MIP just as we did in the previous example. Go to **Network | Interface**; click **Edit** next to the interface you would like to apply the **MIP** to.
2. Click the **MIP** hyperlink at the top of the screen.
3. Define the external **Mapped IP**, as well as the **Netmask**.
4. Define the **Host IP** address of the internal address.
5. As usual, define the **VR** which the firewall will use to route the traffic.
6. Click **OK**.
7. Define the route in the appropriate **VR**. Go to **Network | Routing | Destination**. In the upper right-hand corner, select the appropriate **VR** and click **New**.
8. Define the **IP Address** and **Netmask** of the internal IP address you wish to route to.
9. Select the next hop as a **Gateway** and define the **Interface** which would be considered the outbound interface for the route.
10. Define the **Gateway IP Address** which would be considered the next hop address for the firewall to the host.
11. You can define options such as **metric**, **preference**, **route tag**, and **permanent**.

12. Click **OK**.
13. You must also define a policy to allow the traffic through. To do so, select **Policies**, then define the **From** and **To** zones from the drop-down menus. Click **New**.
14. Specify the **Source Address**, **Destination Address**, **Service**, **Permit**, and any additional options you would like for this policy.
15. Click **OK**.

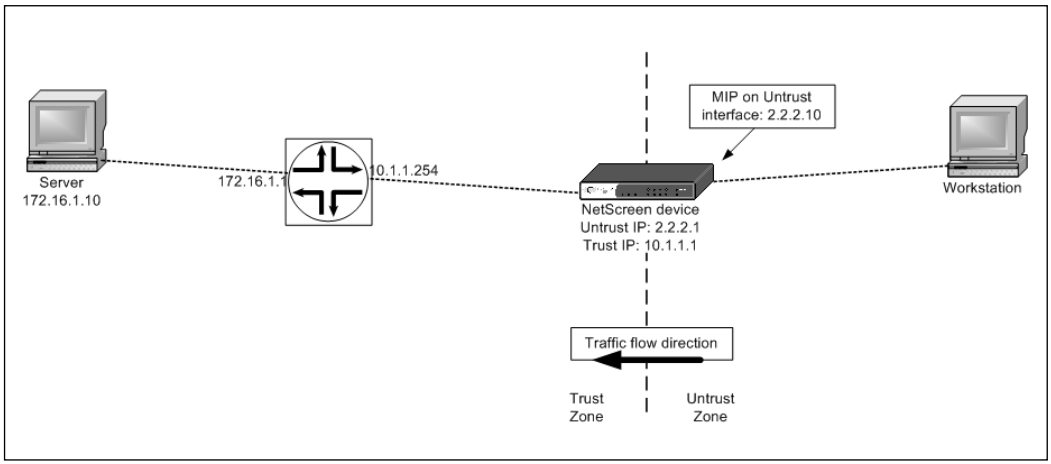
In this example, we will use the following values:

MIP	
Mapped IP	**2.2.2.10**
Netmask	**255.255.255.255**
Host IP	**172.16.1.10**
Virtual Router	**Trust-VR**
Route	
Virtual Router	**Trust-VR**
IP Address/ Netmask	**172.16.1.10/32**
Next Hop	**Gateway**
Interface	**Ethernet1**
Gateway IP Address	**10.1.1.254**
Metric	**1**
Policy	
From Zone	**Untrust**
To Zone	**Trust**
From Address	**Any**
To Address	**MIP(2.2.2.10)**
Service	**HTTP**
Action	**Permit**
Logging	**Enabled**

In this example, we will use the following configuration:

```
set interface "untrust" mip 2.2.2.10 host 172.16.1.10 netmask 255.255.255.255 vr
"trust-vr"
set vrouter trust-vr route 172.16.1.10 255.255.255.255
set policy from Untrust to Trust any MIP(2.2.2.10) http permit log
set vrouter trust-vr route 172.16.1.10 255.255.255.255 gateway 10.1.1.254 metric
1
       save
```

Figure 8.9 MIP Host Behind Another Network Segment



TIP

The MIP address needs to be on the same subnet as the interface's subnet, but cannot overlap with another IP(s) within the same subnet (in other words, you cannot use a MIP address from an IP address already defined within an existing dynamic IP [DIP] pool). However, there are two advantages if the interface is set up on the Untrust zone:

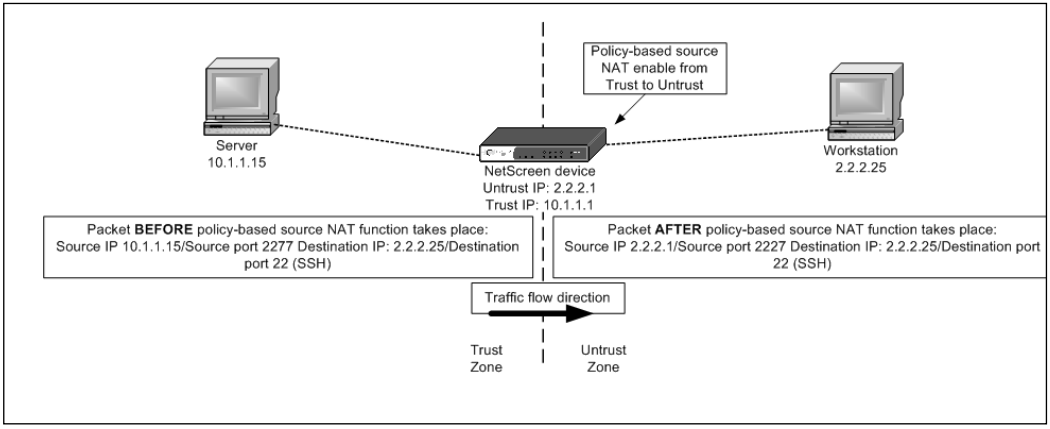
1. On smaller appliances (HSC, 5XT, 5GT) the existing interface IP that is bonded to the Untrust zone as a MIP address can be used.
2. A MIP address on a subnet different than the Untrust interface is allowed. This requires that a route be defined to indicate that traffic destined for that MIP address subnet must go through the Juniper Untrust interface.
3. Internal machines can reach a MIP address assuming that a policy allows it. Of course, you will need to ensure that a route exists to reach this MIP address elsewhere in your network.
4. MIPs can be applied to virtual interfaces such as sub-interfaces and tunnel interfaces. You essentially apply them in the same manner you would a physical interface.

Policy-Based Source NAT

Policy-based source translations are accomplished by creating a firewall rule with source NAT enabled on the policy. By default, the outbound interfaces' or egress interfaces' IP address in the destination zone is used as the newly translated source address. PAT is also

enabled by default. Figure 8.10 shows a host behind the Juniper Trust zone (10.1.1.15) sending a packet out to the Internet via the Juniper device that is acting as the default gateway. The Juniper policy will use source NAT for traffic sourcing from the Trust to Untrust zone. The source NAT address will be the egress interface IP—in this case, the Untrust Ethernet interface. The figure shows the packet before and after translation using the IP address of the egress interface on the Untrust side of 2.2.2.1.

Figure 8.10 Example of Policy-Based Source NAT



Address objects must be defined before any firewall rule is created. Two address objects are created for this example, one for the Trust zone and one for the Untrust zone.

To configure this example through the Juniper WebUI, follow these steps:

1. For the Trust zone address definition, go to **Objects** | **Addresses** | **List**, select the appropriate **Zone**, and click **New**.
2. Specify the **Address Name** of the object, and optionally, a **Comment**.
3. Define the **IP Address/Netmask** and click **OK**.
4. Define the object for the Untrust zone by going to **Objects | Addresses | List**, selecting the appropriate **Zone** for the object, and then clicking **New**.
5. Define the **Address Name**, **Comment (Optional)**, **IP Address**, and **Netmask**. Then, click **OK**.
6. Create a firewall rule by going to **Policies**, defining the **From** and **To** Zones, and then clicking **New** in the upper right-hand corner.
7. Specify the **Source** and **Destination** addresses.
8. Define the **Service**, and make sure the **Action** is set to Permit.

9. Define the NAT for this policy by clicking the **Advanced** button at the bottom of the screen. Check the **Source Translation** address box. In the drop-down menu on the right, select **None (Use Egress Interface IP)**.
10. Click **OK**.

In this example, select the following options:

Address Object 1	
Address Name	**10.1.1.15**
IP Address/Netmask	**10.1.1.15/32**
Zone	**Trust**
Address Object 2	
Address Name	**2.2.2.15**
IP Address/Netmask	**2.2.2.15/32**
Zone	**Untrust**
Policy	
From Zone	**Trust**
To Zone	**Untrust**
Source Address	**10.1.1.15**
Destination Address	**2.2.2.15**
Service	**SSH**
Action	**Permit**
Logging	**Enabled**
Source NAT	**Enabled**
DIP (ON)	**Use Egress Interface IP**

```
In this example, we will use the following settings
set address trust 10.1.1.15 10.1.1.15 255.255.255.255
set address trust 2.2.2.15 2.2.2.15 255.255.255.255
set policy from Trust to Untrust 10.1.1.15 2.2.2.15 ssh nat src permit log
save
```

Policy-based NAT can perform the same functions as a MIP; however, there are a couple of advantages to using a policy-based NAT for one-to-one mapping. Using one form of translation, such as policy-based translation over older features such as interface-based translation, MIP and VIP provide a more unified way of managing your NAT functions. Another advantage of using policy-based NAT is that the only limitation is the number of policies you can create, which far exceeds any MIP capacity. The matrix in Table 8.2 shows the policy capacity for each firewall platform, and is presented according to the datasheets published by Juniper for ScreenOS 5.4.

Table 8.2 Policy Capacity Matrix

Product Name	Policy Capacity
NetScreen HSC	50
NetScreen 5XT	100
NetScreen 5GT	100
Juniper SSG 5/20	200
NetScreen 25	500
NetScreen 50	1000
Juniper SSG 140	500
NetScreen 204	4000
NetScreen 208	4000
NetScreen 500	20,000
Juniper SSG 520	1000
Juniper SSG 550	4000
NetScreen ISG 1000	10,000
NetScreen ISG 2000	20,000 base / 30,000 adv
NetScreen 5200/5400	40,000

DIP

DIP address translation provides you with another method of performing NAT on the firewall. DIPs begin with a definition for the DIP pool, which defines the address ranges. DIP pools are created on interfaces that can be a physical interface, a sub-interface, a VPN tunnel interface, or a loop-back interface. A DIP can be defined as a single host or a range of contiguous hosts known as a pool. If it is a pool of address definitions, the pool must be in consecutive order. Therefore, it is important to note that no other IP(s) within that pool can be used anywhere else (for example, a MIP definition).

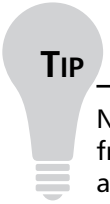
Normally, the DIP pool must be in the same subnet as the interface it is applied on. There is an exception to this when you configure it on an extended IP range. Also, you can configure DIP on as an address shift, where you create a range that is mapped to an internal range in order to provide a contiguous pool of addresses.

DIP pool definition also offers the option to disable or enable PAT. Since the DIP pool is only used in source NAT scenarios, PAT on the source ports can be utilized to increase the amount of usage for each address within the pool.

DIP Properties

Several properties can be used to configure a DIP pool. This is because the DIP pool is an extremely flexible form of NAT which can be placed in various locations. In this section, we will cover the different DIP properties.

- **ID** This value uniquely identifies the DIP pool on the device.
- **IP Address Range** You must define the contiguous range. You define the beginning and ending values of the range.
- **Port Translation** By checking this value, you can enable Port Translation for the DIP pool. This will perform a many-to-one mapping.
- **IP Shift** An IP shift defines the base IP address of the internal range, and then defines the base and ending IP addresses of the DIP range. The total number of IP addresses in the DIP range will be used to determine the ending value of the base range. This performs a one-to-one mapping.
- **In the Same Subnet as the Interface IP or Its Secondary IPs** This option specifies that the DIP pool will be located on the primary interface. This is similar to how MIPs function when they are bound to an external interface.
- **Incoming NAT** This option enables incoming NAT for inbound connections.
- **In the same subnet as the extended IP** This option can be specified if you would like to create the DIP pool on an extended IP address for the interface. This range is not on the same subnet as the primary interface. You must enter the **Extended IP** address and **Netmask** for the DIP pool.



Normally, when you configure DIP pool, it may assign different IP addresses from the DIP pool for each session. If you would like to prevent this functionality, configure Sticky DIP, which will maintain the sessions to a specific IP address. Of course, you need to make sure you will not run out of IP addresses for each machine. This configuration can only be done through the CLI with the **set dip sticky** command.

Also, DIP Pools will use PAT by default. If you want to disable this functionality, you must enable the **fix-port** option on the CLI (*set interface <interface> dip <pool values> fix-port*) or uncheck the **Port Translation** option in the WebUI.

Configuring DIP on a Policy

The following example shows policy-based source NAT using a DIP pool with PAT disabled and traffic flow from Trust to Untrust. Address objects must be defined before a firewall can be created.

To configure this example in the Juniper WebUI:

1. First create the objects. Go to **Objects | Addresses | List** and select the appropriate **VR**. Then, click **New**.
2. Create another object by performing the same steps.
3. Create the DIP object by selecting **Network | Interface** and clicking the **Edit** hyperlink. Then, click the **DIP** hyperlink at the top of the screen.
4. Define the **ID**, **IP Address Range**, **Port Translation**, and **Subnet Location**.
5. Click **OK**.
6. Configure the policy by going to **Policies** and selecting the appropriate **From** and **To** zones. Afterward, define the **Source Address**, **Destination Address**, **Service**, and **Action**.
7. Define any other additional options for the policy such as logging, scheduling, traffic shaping authentication, and so on.
8. To configure the **DIP** on the policy, click the **Advanced** button.
9. Select **Source NAT**, and then select the appropriate **DIP** pool from the **DIP On** drop-down menu.
10. Click **OK**.

In this example, we will use the following settings:

Address Object 1	
Address Name	**10.1.1.0/24**
IP Address / Netmask	**10.1.1.0/24**
Zone	**Trust**
Address Object 2	
Address Name	**2.2.2.50/32**
IP Address Netmask	**2.2.2.50/32**
Zone	**Untrust**
DIP	
Interface	**Untrust**
ID	**4**
IP Address Range	**2.2.2.2 ~ 2.2.2.10**

Port Translation	**Disabled**
In the same subnet as interface	**enabled.**
Policy	
From Zone	**Trust**
To Zone	**Untrust**
Source Address	**10.1.1.0/24**
Destination Address	**2.2.2.50**
Service	**HTTP**
Action	**Permit**
Logging	**Enabled**
Source NAT	**Enabled**
DIP (ON)	**4(2.2.2.2~2.2.2.10)**

```
In this example, we will use the following settings:
set address trust 10.1.1.0 10.1.1.0 255.255.255.0
set address trust 2.2.2.50 2.2.2.50 255.255.255.255
set interface Untrust dip 4 2.2.2.2 2.2.2.10 fix-port
set policy from Trust to Untrust 10.1.1.0/24 2.2.2.50 http nat src dip-id 4
permit log
save
```

A DIP pool can also contain one IP range. For example, when defining the single IP address 2.2.2.2 within a DIP pool, the IP address range would be 2.2.2.2 ~ 2.2.2.2. This is an alternate way of using a different IP address than what is currently defined on the egress interface. It is recommended you enable PAT within the DIP pool definition when creating for a one IP range.

NOTE

When PAT is disabled in a DIP pool, the IP pool assignment remains the same for the host for all concurrent sessions. When PAT is enabled, the IP pool assignments rotate in a round-robin fashion for each new session.

Sticky DIP

Sticky DIP provides the capability for the translated host to maintain its IP pool assignment. By default, the IP addresses within the DIP pool are rotated in a round-robin fashion for each new session. For example, when there exists a DIP pool of 2.2.2.2 to 2.2.2.10, the host (10.1.1.10) utilizing the DIP pool will have a new NAT IP pool assignment starting at

2.2.2.10 for each new session. Figure 8.11 illustrates the default scenario without Sticky DIP enabled, while Figure 8.12 shows the scenario with Sticky DIP enabled. Note that the Sticky DIP setting is a command line–only setting.

Figure 8.11 Policy-Based Source NAT Using a DIP Pool

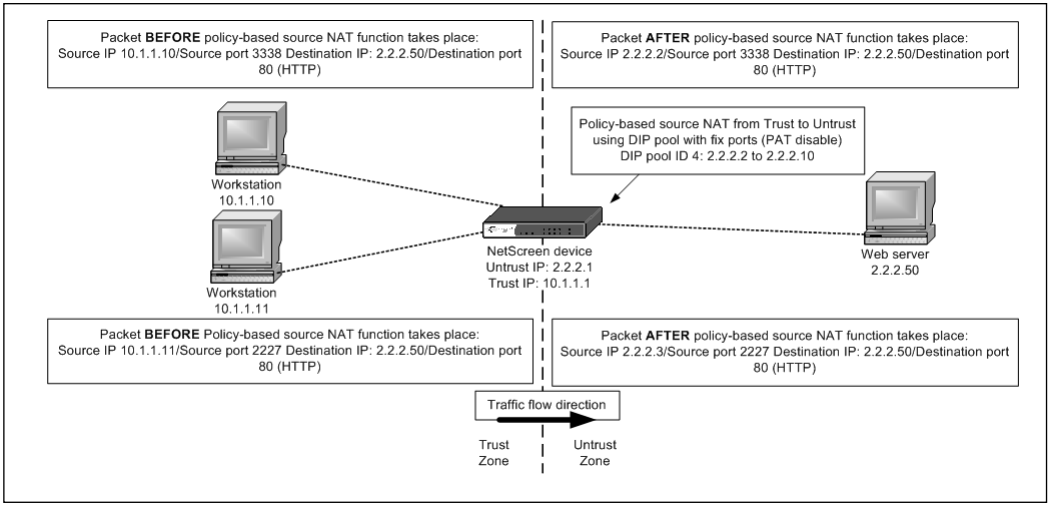
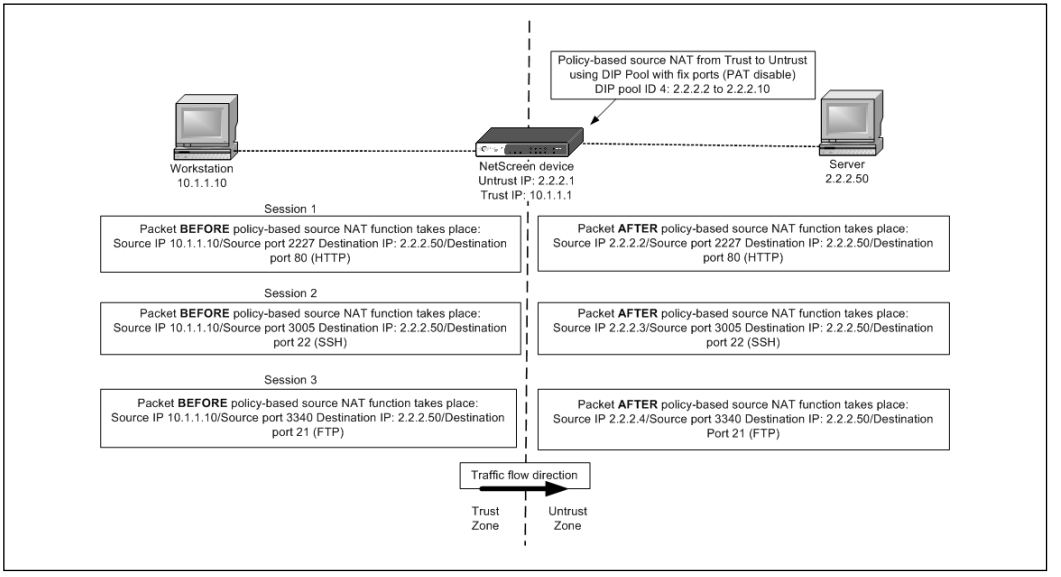


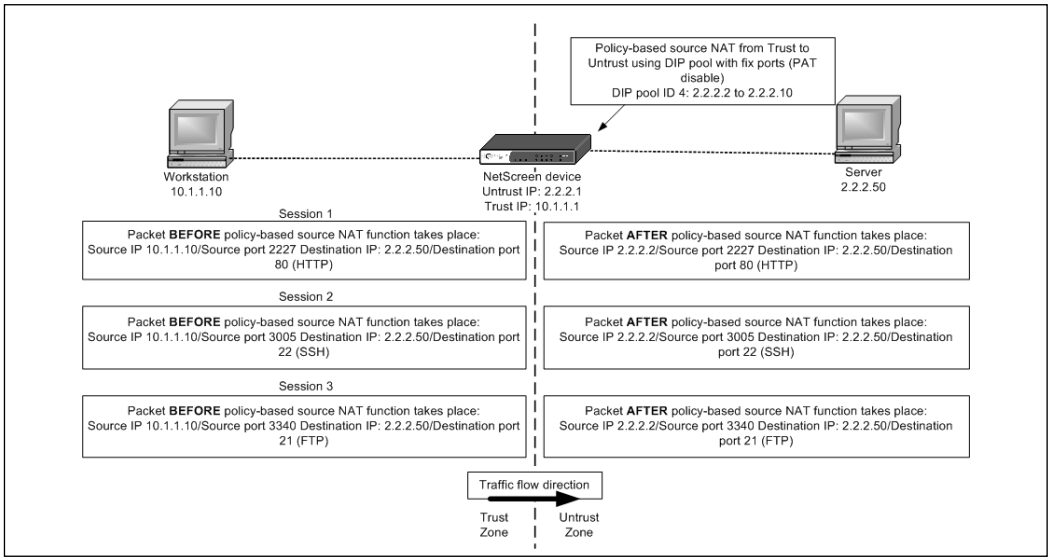
Figure 8.12 Policy-Based Source NAT DIP Pool Usages without Sticky DIP



The example in Figure 8.12 shows that the Juniper device assigns a different IP address from the pool for each new session originated from workstation 10.1.1.10. This pool assignment is done via round-robin fashion.

The example in Figure 8.13, however, shows the same DIP pool usage as shown in Figure 8.12, but with the Sticky DIP feature enabled. The Juniper device now maintains the same IP address (2.2.2.2) assignment for all sessions generated from 10.1.1.10.

Figure 8.13 Policy-Based Source NAT DIP Pool Usage with Sticky DIP



As of this writing, enabling Sticky DIP must be done via the command-line interface (CLI). The command to enable Sticky DIP is

```
set dip sticky
```

Tools & Traps…

Packet Captures

One of the most powerful tools a network administrator can use is a packet capture program. This can help shed light on the exact bit-for-bit composition of traffic. One of the most popular open-source packet sniffers is Wireshark, formally known as Ethereal. We recommend that you thoroughly explore this appli-

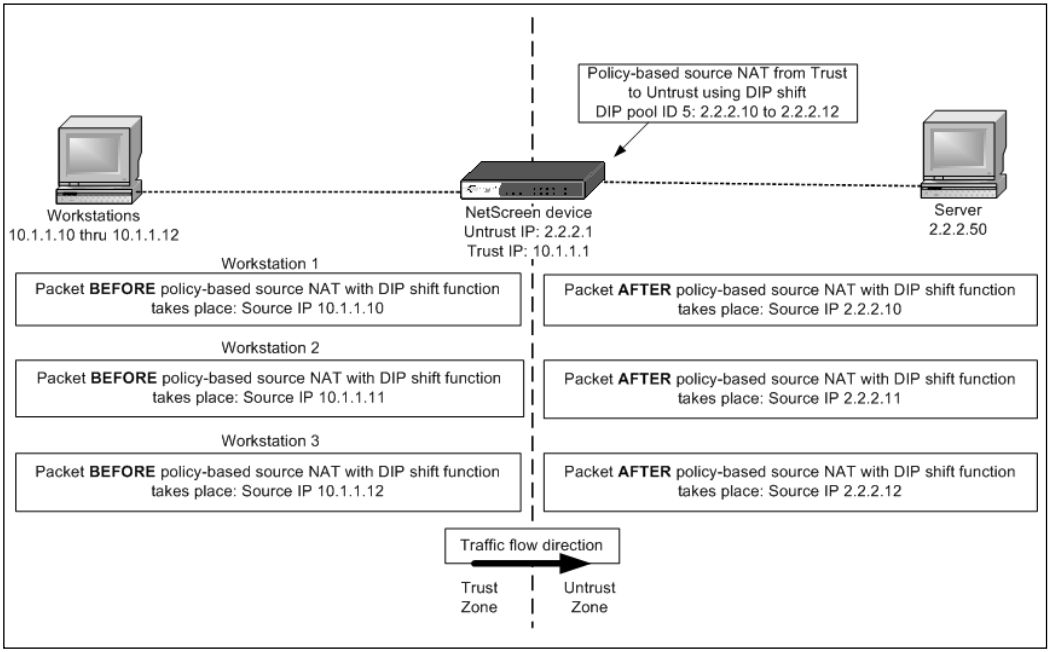
Continued

cation if you are not already intimately familiar with it. It can save you a lot of time and effort, and you will probably learn a great deal about the exact traffic that is passing through your network. We also recommend you experiment with it by sniffing traffic on the inside and outside of the firewall (perhaps with a hub) to help display the different types of traffic and the end result of NAT operations.

DIP Shift

DIP shift ensures that the translated IP pool assignment ranges are one-to-one mapping with the original host IP ranges requesting the translation. The advantage of this feature is that it provides a predictable translation. For example, if communication takes place to a remote end with another firewall, the administrator can define a more granular source IP access rather than allow a range of addresses to come in. The following example shows a range of hosts from 10.1.1.10 to 10.1.1.12 using a DIP shift definition to map its one-to-one mapping translations. The DIP shift pool will always be assigned in order from the first mapping original IP. Note that PAT is not supported in DIP address shifting. Figure 8.14 shows the one-to-one mapping translation for all traffic generated from the original host IP.

Figure 8.14 Policy-Based Source NAT Using DIP Shift



To configure this example in the Juniper WebUI:

1. Create the DIP pool on the egress interface as defined in Figure 8.14. Go to **Network** | **Interfaces**, and then click the **Edit** hyperlink next to the interface you would like to create the DIP on.
2. Click the **DIP** hyperlink at the top of the screen, and then click the **New** button in the upper right-hand corner of the screen.
3. Define the **ID** as a unique value which is not defined to other DIP pools.
4. Define the DIP pool as an **IP Shift** instead of a range. To do so, you must define the **From** address, which is the base internal address. Afterward, define the **To** range, which is the beginning of the external range, to the end of the external range. This must be contiguous. The firewall will automatically calculate the end of the internal range based upon the size of the external range.
5. For this example, select that you will use the same subnet applied to the interface.
6. Click **OK**.
7. Create a policy to allow this traffic outbound and apply the DIP pool. To do so, go to **Policies**, select the appropriate **From** and **To** zone, then click **New**.
8. Define the **Source** and **Destination addresses** to match, along with the **Service**. We will want to **Permit** this traffic.
9. Click the **Advanced** button. Check the **Source Translation** checkbox and define the **DIP** pool which we created to perform the shift.
10. Define any additional options you would like to include in the policy.
11. Click **OK**.

In this example, we will use the following values:

DIP	
Interface	**Untrust**
DIP ID	**5**
IP Shift From	**10.1.1.10**
To	**2.2.2.10~2.2.2.12**
In Same Subnet	**Yes**

Policy	
From Zone	**Trust**
To Zone	**Untrust**
Source Address	**10.1.1.0/24**
Destination Address	**2.2.2.50**
Services	**Any**
Action	**Permit**
Source Translation	**Enabled**
DIP Pool	**5(2.2.2.10~2.2.2.12)/ip-shift**
Logging	**Enabled**

```
To create this example in the Juniper CLI:
set interface Untrust dip 5 shift-from 10.1.1.10 to 2.2.2.10 2.2.2.12
set policy from "Trust" to "Untrust"  "10.1.1.0/24" "2.2.2.50/32" "ANY" nat src
dip-                   id 5 permit log
save
```

WARNING

The Juniper firewall will not prevent you from trying to configure a policy which is more restrictive than the DIP Pool. For instance, say that you have a policy which matches Source IP 10.1.1.0/24, Destination 192.168.1.0/25, Service Any, Permit, with a DIP pool with an IP Shift for 192.168.1.0/24; then Any, Any, Any Deny below it. Since the first policy will only match the /25 range, any other traffic in the upper half of the /24 range will be blocked, so you have to be careful of where the lines of the policy begin and end.

Destination NAT

Although source NAT performs address translation on the source packet to help hide the identity of the original host behind another IP address (often public), destination NAT allows you to change the destination IP address of the packet before it goes to the destination. This can be useful, or necessary, depending on the architecture of the network. Sometimes you may want to advertise access to one end system, and redirect the traffic to another. Or perhaps you have a VPN with a trading partner with the same internal IP address range as you, so you must perform a destination translation. Lastly, you may have a limited number of IP addresses, so you would like to forward traffic based upon the service of the traffic.

In this section, we will discuss the different types of destination NAT. We will begin with VIP NATs and then continue on to cover policy-based destination NAT. Each different type of destination NAT will be explained, properties will be detailed, and examples will be given to reinforce the topics covered. Please note that MIP is a form of source and destination NAT and was covered in the "Source NAT" section earlier for both examples.

Notes from the Underground…

Hackers Bypassing Firewalls

On older firewalls that simply imposed access lists that were not truly stateful, a hacker could potentially bypass some of the security rules by spoofing the source address of the packets so that it looked like they came from the internal private network, when in reality they actually came from an external machine. If administrators were not careful to block this traffic from entering their network, the firewall might just allow it in. Modern firewalls have the potential to not only prevent this attack, but can also use stateful inspection to help defend against randomly inserted traffic.

VIP

When you need to translate multiple services to (potentially) different end systems and you have a limited number of external IP addresses, a VIP may make a good solution. A VIP allows you to create a port forwarding rule which can determine where the traffic should be forwarded to based upon where the traffic arrives (external VIP interface) and the external port which the traffic is directed to.

VIPs differ from MIPs because a VIP provides a one-to-many mapping scenario, whereas a MIP provides a one-to-one mapping. The one-to-many mappings a VIP performs are more related to a combination of destination NAT and PAT.

VIP Properties

Depending on what you are trying to accomplish with a VIP translation, you can configure several different options. We will cover the different properties so you will have enough knowledge to empower your decisions when implementing VIPs.

- **Interface Applied** When you create a VIP you must apply it to a specific interface. VIP definitions are created in the global zone regardless of where they are created so you can apply them appropriately for the policy.
- **Where to Create VIP** You can either create a VIP on the **Same as the Untrusted interface IP address**, or on a **Virtual IP Address**. The Virtual IP Address option allows you to define the virtual IP address that will create the VIP on. This must be located within the same subnet as the interface that the VIP is

being applied (since the interface must proxy ARP for it). It must also be unique both on the firewall, and also from other IP addresses in the subnet.

- **VIP Services** A VIP represents an IP address which can be multiplexed into multiple different end hosts based upon the destination and port number. Each VIP can contain multiple different services which map to different machines on the internal networks.
- **Virtual IP** Within a VIP service, you must specify which VIP this service will apply to.
- **Virtual Port** This is the port number on the VIP which will be used to match traffic to this VIP. So, for instance, if you wanted to match any HTTP traffic directed to the VIP address, you would use port 80.
- **Map to Service** This is the internal service that you would like to match the service on. You will have an option to select any predefined or custom service. If the service is not already defined (such as TCP 5917) then you must create it as a custom service first before you define the VIP. The destination port of the traffic will be changed from the Virtual Port value to whatever value you have defined here when it passes through the VIP.
- **Map to IP** This is the internal IP address that the traffic should be mapped to when it passes through the VIP. This value will change from the VIP IP address to this value after NAT is performed.
- **Service Auto Detection** This option will enable the firewall to periodically check to see whether the end system is active. If the system is not active, then traffic will not be forwarded through the VIP.

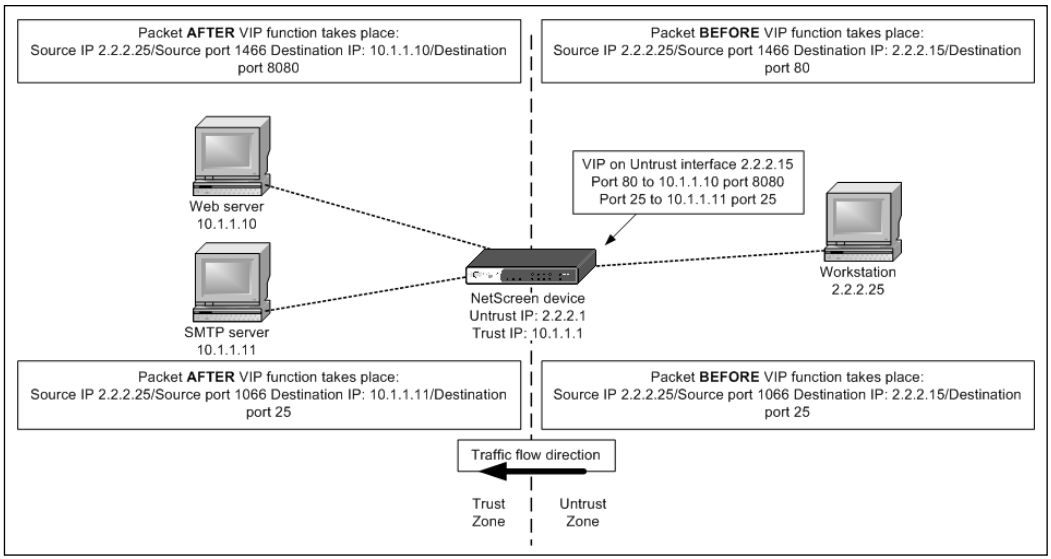
WARNING

You cannot define the Virtual Port to be the same as a service port on the firewall when it is defined on a firewall interface which accepts the control connections. For instance, if you have HTTP enabled on the untrust interface for management, you cannot define a VIP with a Virtual Port of 80 since they directly overlap. However, you have a few options to work with:

1. You can change the management ports for management services (a good idea anyway).
2. You can create the VIP on a virtual address, which will not cause an issue since there won't be service ports defined on it.
3. You could use a different Virtual Port number and still translate it to port 80 on the Mapped Port.

VIP definitions are placed into the global zone no matter which interface/security zone it was originally defined in. Once a VIP is defined, a firewall rule must be set up to allow for traffic destined for the MIP address. Within the firewall rule creation, the VIP can be selected from a global zone or the zone that the VIP address was originally defined in (see Figure 8.15).

Figure 8.15 A VIP Example



To configure this example via the Juniper WebUI:

1. Define the VIP by going to **Network** | **Interfaces** and clicking the **Edit** hyperlink next to the interface you would like to create the VIP on.
2. At the top of the screen, click the **VIP** hyperlink.
3. For this example, we will create a new virtual IP address instead of using the same IP address as the Untrusted interface. Select **Virtual IP Address**, enter the appropriate VIP value, and click **Add**.
4. Add the VIP mapping for the internal host by selecting **New VIP Services** on the top left portion of the Web UI.
5. Make sure that the appropriate VIP is selected by the drop-down menu. Next, you will need to define the **Virtual IP Port** which the traffic will be matched to.
6. Define the **Map to Service** port which will specify the port that the traffic will be translated to when it passes through the VIP.
7. Define the **Map to IP** address of the server on the internal side of the network which will accept the connections.

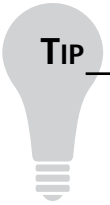
8. You may optionally define to have the firewall perform a **Server Auto Detection** check to make sure the service is active before it tries to forward traffic.
9. Click **OK**. You can perform these steps for multiple VIPs if you would like.
10. Create a policy to allow this traffic to be forwarded on the firewall. To do this, go to **Policies** and specify the appropriate **From** and **To** zones on the firewall, then click **New**.
11. Specify the appropriate **Source Address**. For the destination, you will define the **VIP**. The service must match the **Virtual Port** in the VIP. The action should be **Permit**. Define any additional policies options such as **Logging** for this policy.
12. Click **OK**.

In this example, we will use the following configuration:

VIP	
Interface	**Untrust**
VIP	**Virtual IP Address 2.2.2.5**
Virtual Port	**80**
Map to Service	**HTTP(8080)**
Map to IP	**10.1.1.10**
Server Auto Detection	**Disabled**
Virtual Port	**25**
Map to Service	**SMTP(25)**
Map to IP	**10.1.1.11**
Server Auto Detection	**Disabled**
Policy	
From Zone	**Untrust**
To Zone	**Trust**
Source Address	**Any**
Destination Address	**VIP::1**
Services	**HTTP & SMTP**
Action	**Permit**
Logging	**Enabled**

```
To configure this example via the Juniper CLI, perform the following steps:
set interface Untrust vip 2.2.2.5 80 "HTTP-ALT" 10.1.1.10 manual
set interface Untrust vip 2.2.2.5 25 "MAIL" 10.1.1.11 manual
set policy id 8 from "Untrust" to "Trust"  "Any" "VIP(2.2.2.5)" "HTTP" permit
log
set policy id 8
set service "SMTP"
```

```
exit
save
```



TIP

1. VIP only works on interfaces bonded to the Untrust zone.
2. The VIP address has to be on the same subnet as the Ethernet interface where the VIP is defined.
3. Custom ports can be created for VIP mappings.
4. You can use the existing IP assignment on the Ethernet interface. During your VIP creation select **Same as the Untrusted interface IP address**. This feature is only limited to the HSC, 5XT, 5GT, 25, and 50 models.

Policy-Based Destination NAT

Policy-based destination NAT is a configurable option that you can define within a firewall policy, as the name implies. Just like the source-based NAT configuration, there is a separate definition in place to define a destination NAT. Unlike source NAT, there are no requirements to predefine settings on the interfaces. For example, you do not need to create a DIP pool before actually creating a destination NAT firewall rule. The address schemes for the newly translated destination are all defined within the firewall rule. Besides destination NAT, a destination PAT can also be defined. The options available to perform destination NAT are as follows:

- Destination NAT to another IP
- Destination NAT to another IP with PAT to a different port
- Destination NAT to an IP range

NOTE

When using destination NAT, it is important to remember the Juniper packet flow. Route lookup on the Juniper firewall occurs *before* and *after* policy lookup. Policy lookup also entails policy-based NAT functions. Therefore, there might be a need to create a route *before* or *after* a policy-based NAT function takes place.

When to Use Policy-Based Destination NAT

Policy-based NAT can be put to good use in a few different situations. First, policy-based NAT is much more extensible than interface-based NAT since you can configure as many policy-based NAT rules as your firewall can support policies. Interface-based NAT is much more restrictive, especially on the smaller Juniper firewalls. The other place when policy-based NAT can be used in an effective manner is when you are NATing to a zone in a different subnet than your Untrust interface. For instance, if you have an external interface 1.1.1.1/24, and a DMZ with 2.2.2.1/24 (internal DMZ range 172.16.1.0/24), then you can perform a policy-based destination NAT rule from Untrust to DMZ. Then NAT the traffic to the internal DMZ range 172.16.1.1/24 all in a single policy.

When Not to Use Policy-Based Destination NAT

Policy-based destination NAT is not a good choice when you do not have a separate subnet to NAT traffic to. In the preceding example, if you did not have a separate DMZ subnet 2.2.2.0/24, but rather only the 1.1.1.0/24 subnet, policy-based destination NAT would not work. There are several reasons. First, if you tried to create a policy whose destination was in the same subnet as the Untrust interface, the firewall would not perform a gratuitous ARP for the virtual object (say 1.1.1.50). Next, even if you put the ARP entry in your external device manually, the firewall determines what the from and to zones are, based on a route lookup (where the traffic is going). Taking it a step further, you could create a static route for this destination host (say 2.2.2.50/32 to DMZ). By now, you probably feel like you would be jumping through way too many hoops. We recommend that you just go with interface-based NAT in this scenario. It's just too much trouble to try to get this to work. With interface-based NAT, the firewall handles the ARP and Routing and is tried and proven.

Policy-Based Destination NAT Properties

Destination NAT that is performed in a policy has a few simple properties that are configurable. The granularity of being able to configure NAT in a policy is a great advantage to using destination NAT on a per-policy basis—along with the fact that you can configure far more policy-based NAT entries than you can MIP, or VIPs.

- **Destination Translation** This option must be set in the policy for destination NAT to be enabled.
- **Translate to IP** This is the IP address that you would like all traffic that matches this rule to be translated to.
- **Map to Port** You can additionally perform a PAT translation from the port that matched the firewall rule, to the port that you define in this field.

- **Translate to IP Range** You can use a destination range translation which will perform an address shift from the original range to the new range similar to a DIP IP shift.

Destination NAT Scenarios

The following examples are some possible scenarios that destination NAT can accomplish when configured within a firewall policy.

One-to-One Mapping

A one-to-one mapping scenario illustrates a translation from one host to another host. One-to-one mapping is equivalent to a static NAT or the MIP feature.

The following is a Web UI configuration example for a one-to-one mapping as defined in Figure 8.16.

1. Create the firewall rule to perform the destination NAT. Go to **Policies** and select the appropriate **FROM** and **TO** zones, then click **New**.
2 Begin by filling in the parameters which will be used to match the traffic to perform destination-based NAT on. You must fill out the appropriate **Source** and **Destination Addresses**, as well as the **Services**, and **Action**.
3. Click the **Advanced** button. You will now need to define the destination-based address translation for this policy.
4. Select the **Destination Translation** option for the policy. Next, define the translation type for the policy. In this example, we will use **Translate to IP** since we only want to translate to one address. Specify the address you want the traffic to be defined as and click **OK**.
5. Create a route in the VR which routes for the untrust interface. This is needed so the firewall can determine the zone-to-zone policy lookup. Go to **Network | Routing | Destination** and click the **VR** in the upper right-hand corner and then click **New**.
6. Define the **IP Address** and **Netmask** for the route, and define the **Interface** you would like to route the internal traffic to so that the zones will match.

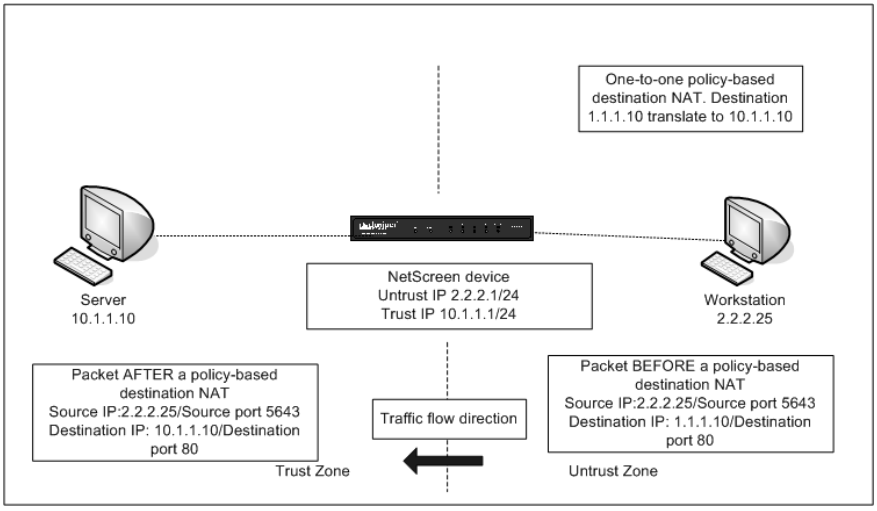
In this example, we used the following settings:

Policy	
From Zone	**Untrust**
To Zone	**Trust**
Source Address	**2.2.2.25**
Destination Address	**1.1.1.10**
Service	**HTTP**

Action	**Permit**
Destination Translation	**Enabled**
Translate to IP	**10.1.1.10**
Route	
VR	**Trust-VR**
IP Address/Netmask	**1.1.1.10 / 32**
Interface	**Trust**

```
To configure this example via the Juniper CLI
set policy from "Untrust" to "Trust"  "Any" "1.1.1.10/255.255.255.255" "HTTP"
nat                   dst ip 10.1.1.10 permit log
set vrouter trust-vr route 1.1.1.10 255.255.255.255 interface trust
save
```

Figure 8.16 One-to-One Policy-Based Destination NAT



Many-to-One Mapping

A many-to-one mapping scenario illustrates that traffic sent to several different destinations can be translated to a single host. You may want to do this to create the illusion of having multiple servers, or perhaps you are migrating from one address range to another and you need to have multiple external addresses map to the same internal host.

The following Web UI configuration example for a many-to-one mapping is defined in Figure 8.17.

1. Create the address objects of the destination hosts and put them into an address group to be used in the policy-based rule definition.
2. To create address objects, go to **Objects** | **Addresses** | **List** and click **New**. You must perform this function for as many hosts as you need to create. Then click **OK** to add these into the configuration.
3. Fill in the appropriate values for the **Address Name**, **IP/Netmask**, and **Zone**.
4. We will create a group to contain these objects by going to **Objects | Addresses Groups** and clicking the **New** button.
5. Define a **Group Name**, then add the appropriate **Members** to the group.
6. Click **OK**.
7. Create a policy for this destination NAT rule by going to **Policies** and selecting the appropriate **From** and **To** zones. Then, click **New**.
8. Select the **Source Address** for the traffic you would like to match on. Next, you will define the group that was created as the **Destination Address**.
9. Specify the **Service** and **Action**.
10. Click the **Advanced** button. Select the **Destination Translation** option and specify to **Translate to IP**, then define the address.
11. Click **OK**.
12. Create a route in the VR which routes for the untrust interface. This is needed so the firewall can determine the zone-to-zone policy lookup. Go to **Network | Routing | Destination** and select the **VR** in the upper right-hand corner. Then, click **New**.
13. Define the **IP Address** and **Netmask** for the route, then define the **Interface** you would like to route the internal traffic to so that the zones will match.

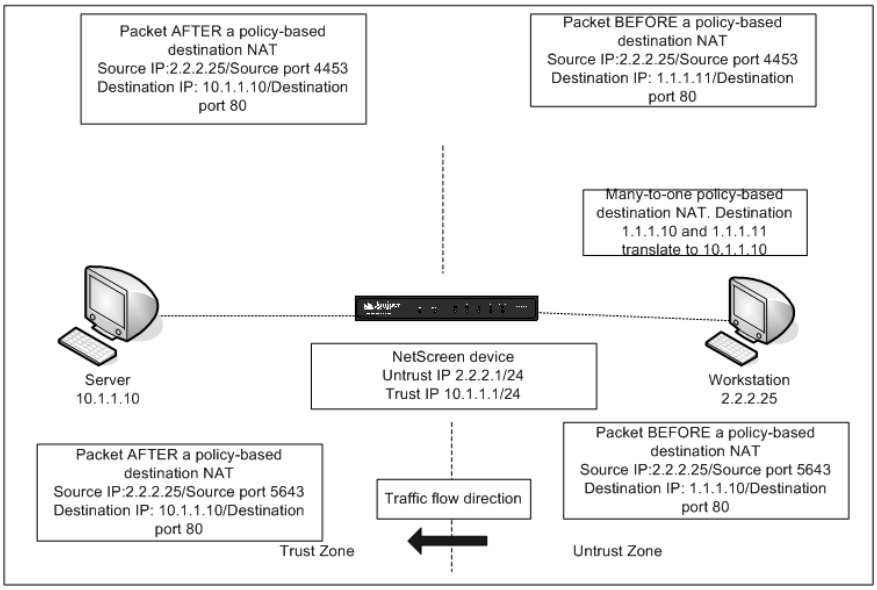
In this example, we will use the following settings:

Address Object	
Address Name	**1.1.1.10/32**
IP/Netmask	**1.1.1.10/32**
Zone	**Trust**
Address Object	
Address Name	**1.1.1.11/32**
IP/Netmask	**1.1.1.11/32**
Zone	**Trust**
Group	

Group Name	**Web_Servers**
Members	**1.1.1.10/32, 1.1.1.11/32**
Policy	
From Zone	**Untrust**
To Zone	**Trust**
Source Address	**2.2.2.25**
Destination Address	**Web_Servers**
Service	**HTTP**
Action	**Permit**
Logging	**Enabled**
Destination Translation	**Enabled**
Translate to IP	**10.1.1.10**
Route 1	
VR	**Trust-VR**
IP Address/Netmask	**1.1.1.10/32**
Interface	**Trust**
Route 2	
VR	**Trust-VR**
IP Address/Netmask	**1.1.1.11/32**
Interface	**Trust**

```
To configure this example via the CLI:
set address "Trust" "1.1.1.10/32" 1.1.1.10 255.255.255.255
set address "Trust" "1.1.1.11/32" 1.1.1.11 255.255.255.255
set group address "Trust" "Web_Servers"
set group address "Trust" "Web_Servers" add "1.1.1.10/32"
set group address "Trust" "Web_Servers" add "1.1.1.11/32"
set policy id 10 from "Untrust" to "Trust"  "2.2.2.25/32" "Web_Servers" "HTTP"
nat                    dst ip 10.1.1.10 permit log
set vrouter trust-vr route 1.1.1.10 255.255.255.255 interface trust
set vrouter trust-vr route 1.1.1.11 255.255.255.255 interface trust
save
```

Figure 8.17 Many-to-One Policy-Based Destination NAT



Many-to-Many Mapping

The many-to-many mapping scenario illustrates that traffic sent to several different destinations can be translated to several other destinations.

The following Web UI configuration example is defined in Figure 8.18.

1. We must begin by creating objects for the devices we need to reference.
2. To create address objects, go to **Objects** | **Addresses** | **List** and then click **New**. You must perform this function for as many hosts as you need to create. Then, click **OK** to add these into the configuration.
3. Fill in the appropriate values for the **Address Name**, **IP / Netmask**, and **Zone**.
4. Create a group to contain these objects by going to **Objects | Addresses Groups** and clicking the **New** button.
5. Define a **Group Name**, and then add the appropriate **Members** to the group.
6. Click **OK**.
7. Now we must create a policy for this destination NAT rule. Go to **Policies** and select the appropriate **From** and **To** zones. Then, click **New**.
8. Select the **Source Address** for the traffic that you would like to match on. Next, define the group that was created as the **Destination Address**.
9. Specify the **Service** and **Action**.

10. Click the **Advanced** button. Select the **Destination Translation** option and specify to **Translate to IP Range**. You must define the beginning and ending IP values for this range.
11. Click **OK**.
12. Create a route in the VR which routes for the untrust interface. This is needed so the firewall can determine the zone-to-zone policy lookup. Go to **Network | Routing | Destination** and select the **VR** in the upper right-hand corner. Then, click **New**.
13. Define the **IP Address** and **Netmask** for the route, and define the **Interface** that you would like to route the internal traffic to so that the zones will match.

This example will use the following parameters:

Address Object 1	
Address Name	**1.1.1.10/32**
IP/Netmask	**1.1.1.10/32**
Zone	**Trust**
Address Object 2	
Address Name	**1.1.1.11/32**
IP/Netmask	**1.1.1.11/32**
Zone	**Trust**
Address Object 3	
Address Name	**1.1.1.12/32**
IP/Netmask	**1.1.1.12/32**
Zone	**Trust**
Group	
Group Name	**Servers**
Members	**1.1.1.10/32, 1.1.1.11/32, 1.1.1.12/32**
Policy	
From Zone	**Untrust**
To Zone	**Trust**
Source Address	**2.2.2.50**
Destination Address	**Servers**
Service	**Any**
Action	**Permit**
Destination Translation	**Enabled**
Translate To IP Range	**10.1.1.10–10.1.1.12**

Route 1

VR	**Trust-VR**
IP Address/Netmask	**1.1.1.10/32**
Interface	**Trust**

Route 2

VR	**Trust-VR**
IP Address/Netmask	**1.1.1.11/32**
Interface	**Trust**

Route 3

VR	**Trust-VR**
IP Address/Netmask	**1.1.1.12/32**
Interface	**Trust**

```
To configure this example with the Juniper CLI:
set address "Trust" "1.1.1.10/32" 1.1.1.10 255.255.255.255
set address "Trust" "1.1.1.11/32" 1.1.1.11 255.255.255.255
set address "Trust" "1.1.1.12/32" 1.1.1.12 255.255.255.255
set group address "Trust" "Servers"
set group address "Trust" "Servers" add "1.1.1.10/32"
set group address "Trust" "Servers" add "1.1.1.11/32"
set group address "Trust" "Servers" add "1.1.1.12/32"
set policy from "Untrust" to "Trust"  "2.2.2.50/32" "Servers" "Any" nat
                     dst ip 10.1.1.10 permit log
set vrouter trust-vr route 1.1.1.10 255.255.255.255 interface trust
set vrouter trust-vr route 1.1.1.11 255.255.255.255 interface trust
set vrouter trust-vr route 1.1.1.12 255.255.255.255 interface trust
save
```

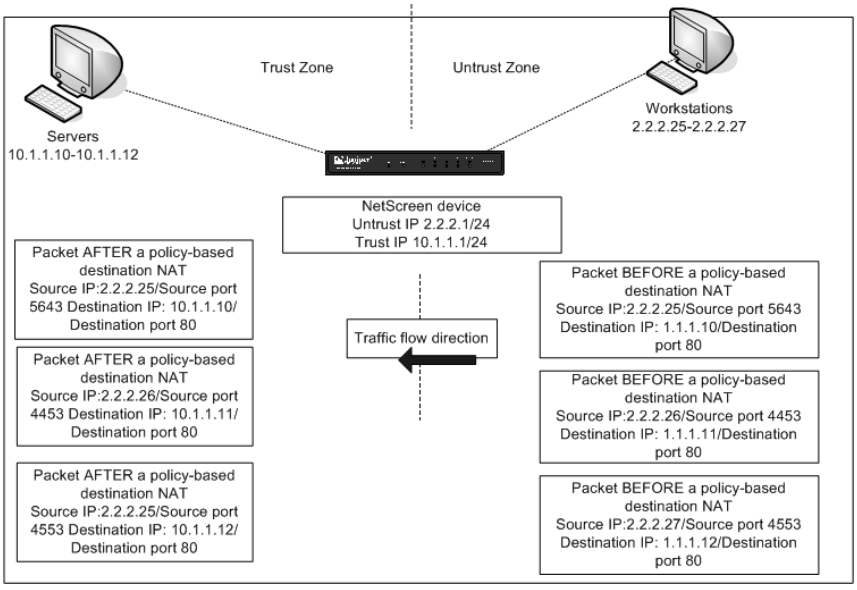
Destination PAT Scenario

Destination PAT provides an alternative destination port from what the original packet is sent to. It can also provide an extra security measure for hiding the original destination port.

The following Web UI configuration example is defined in Figure 8.19.

1. Create the firewall rule to perform the destination NAT and PAT. Go to **Policies** and select the appropriate **FROM** and **TO** zones and click **New**.

Figure 8.18 Many-to-Many Policy-Based Destination NAT



2. Define the appropriate **Source** and **Destination Addresses**. You must also specify the appropriate **Services** to match.
3. Specify the **Action** and click **Advanced**.
4. Enable the **Destination Translation** as well as the **Translate to IP** which is the IP address of the internal server.
5. Check the **Map to Port** box, and then define the destination port which the original port should be translated to.
6. Click **OK**.
7. Create a route in the VR which routes for the untrust interface. This is needed so the firewall can determine the zone-to-zone policy lookup. Go to **Network | Routing | Destination** and select the **VR** in the upper right-hand corner. Click **New**.
8. Define the **IP Address** and **Netmask** for the route, and then define the **Interface** you would like to route the internal traffic to so that the zones will match.

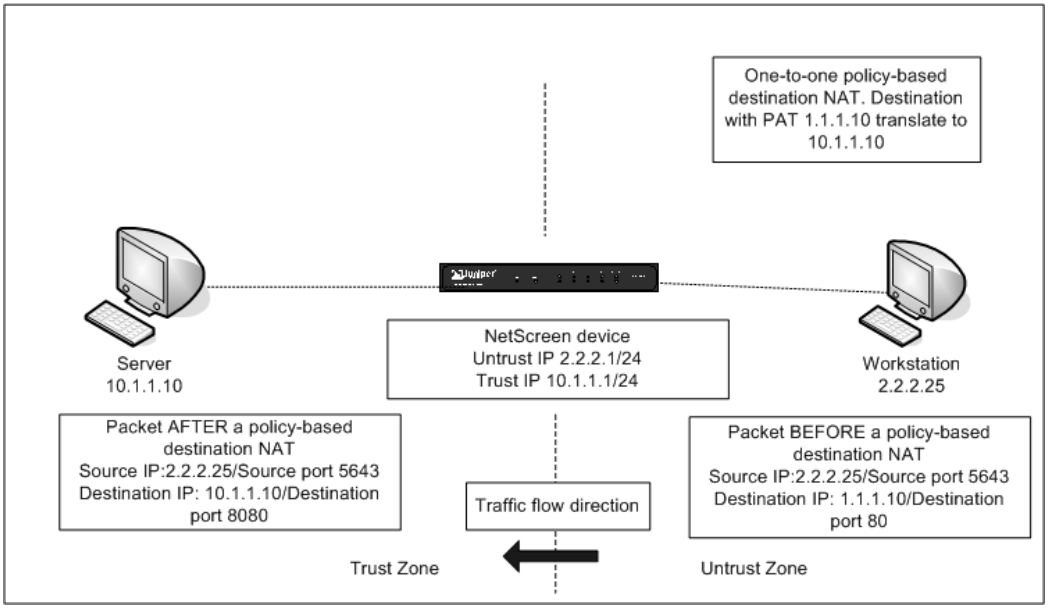
In this example, we will use the following settings:

Policy
From Zone **Untrust**

To Zone	**Trust**
Source Address	**2.2.2.25**
Destination Address	**1.1.1.10**
Service	**HTTP**
Action	**Permit**
Logging	**Enabled**
Destination Translation	**Enabled**
Translate to IP	**10.1.1.10**
Map to Port	**Enabled, 8080**
Route	
VR	**Trust-VR**
IP Address / Netmask	**1.1.1.10/32**
Interface	**Trust**

```
To configure this example on the Juniper CLI:
set policy from "Untrust" to "Trust"  "2.2.2.25/32" "2.2.2.10/32" "HTTP" nat dst
ip                    10.1.1.10 port 8080 permit log
set vrouter trust-vr route 1.1.1.10 255.255.255.255 interface trust
save
```

Figure 8.19 One-to-One Policy-Based Destination NAT with Destination PAT



Source and Destination NAT Combined

Sometimes, you might need to translate the original source IP address of the packet, along with the destination. An example of this might be if you have to communicate with a private address range that overlaps with your internal private range. You could use source and destination translation to accomplish this. Source and destination address translation can be combined together in a single firewall rule. The following example shows a source and destination NAT.

The following Web UI configuration example is defined in Figure 8.20.

1. Create the firewall rule to perform the source and destination NAT. Go to **Policies** and select the appropriate **From** and **To** zones, and click **New**.
2. Specify the appropriate **Source** and **Destination Addresses**.
3. Just like other policies, you must define the **Service** that you would like to match for this traffic.
4. Specify the **Action**, and then click **Advanced**.
5. You must enable the **Source Translation**, and make sure that the DIP is set to **Egress Interface**, or a DIP pool which represents a group of addresses that the traffic could be translated to.
6. Now enable the **Destination Translation**. If you are only going to a single IP address, you can define **Translate to IP** and specify the IP address. If you need to define a range, you can do so with the **Translate to IP Range**.
7. Click **OK**.
8. Create a route in the VR which routes for the untrust interface. This is needed so the firewall can determine the zone-to-zone policy lookup. Go to **Network | Routing | Destination** and select the **VR** in the upper right-hand corner. Click **New**.
9. Define the **IP Address** and **Netmask** for the route, and define the **Interface** that you would like to route the internal traffic to so that the zones will match.

In this example, we will configure the following parameters:

Policy	
From Zone	**Untrust**
To Zone	**Trust**
Source Address	**2.2.2.25**
Destination Address	**2.2.2.10**
Service	**HTTP**
Action	**Permit**
Logging	**Enabled**
Source Translation	**Enabled**

DIP(ON)	**Egress Interface**
Destination Translation	**Enabled**
Translate to IP	**10.1.1.10**

Route	
VR	**Trust-VR**
IP Address/Netmask	**1.1.1.10/32**
Interface	**Trust**

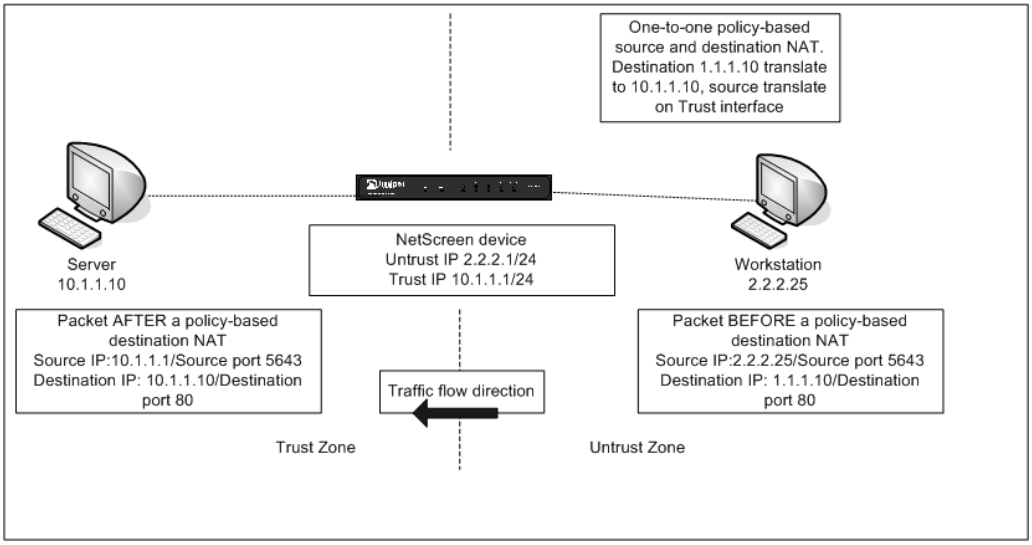
To configure this example via the CLI:

set policy from "Untrust" to "Trust" "2.2.2.25/32" "2.2.2.10/32" "HTTP" nat src dst ip 10.1.1.10 permit log

set vrouter trust-vr route 1.1.1.10 255.255.255.255 interface trust

save

Figure 8.20 Policy-Based Source and Destination NAT



Summary

Network Address Translation is one of the most widely implemented techniques you will find across most networks today. It is found on both large and small networks, and is likely to remain popular until there are enough IP addresses available. NAT provides the ability to hide original source and destination IP addresses behind a different translated IP address. NAT provides a short-term solution to the depleting IPv4 addresses on the Internet. NAT provides the ability to utilize one IP for several thousand devices, thus conserving non-RFC 1918 IP addresses. With the cost of NAT devices going down each year, and the increase in Internet usage, it is not surprising that NAT is a widely used feature.

The NAT and PAT features of the Juniper products are covered in detail with example scenarios and their respective configurations steps.

One of the original methods used for NAT was the interface-based NAT mode, which is enabled by default on the Ethernet interface bonded to the Trust security zone. It is recommended that the interface-based NAT mode setting be disabled and set to Route mode all the time, thus using policy-based NAT instead. Policy-based NAT provides a more efficient and scalable method than interface-based NAT. As seen with MIP and VIP, there are capacity limitations that restrict the use of these NAPT methods. Policy-based translation can perform the same functions as a MIP or a VIP and also has a much larger capacity support.

It is good to note the tips provided throughout this chapter. The goal of these tips is to provide an understanding of the limitations and capabilities of the Juniper firewall address translation features. Knowing how the firewall handles a packet is a key essential for troubleshooting NAT issues.

Links to Sites

- NAT RFC 1631: http://www.faqs.org/rfcs/rfc1631.html
- NAPT RFC 3022: http://www.faqs.org/rfcs/rfc3022.html

Solutions Fast Track

Overview of Address Translation

- ☑ Address translation conserves IP addresses, thus providing the capability to use nonroutable addresses from the RFC 1918 space.
- ☑ Address translation provides a hidden identity for host(s), addresses overlapping subnets, and maintains, a cohesive network.
- ☑ Address translation breaks IPSec traffic; use NAT traversal as a workaround.

- ☑ Address translation breaks applications that require dynamic port allocation; most firewall vendors have ALG, which addresses this issue.
- ☑ Address translation lacks compatibility with legacy-based applications.

Juniper Packet Flow

- ☑ Understanding how Juniper handles packets is very important.
- ☑ MIP and VIP translation occurs before a route and policy lookup.
- ☑ A route lookup occurs *before* and *after* a policy-based NAT function.

Source NAT

- ☑ Prior to ScreenOS 5.0 communication, translating from a host on the Untrust zone to a host on the Trust zone with the interface set for NAT was not possible unless a MIP or VIP was defined. With current ScreenOS releases, this is possible as long as there is a firewall rule in place to allow it.
- ☑ The egress interface has to be bonded to the Untrust zone.
- ☑ When a user-defined zone is bonded to the ingress interface with NAT enabled, that user-defined security zone must be defined on a different virtual router than the Untrust zone.
- ☑ Interface-based NAT will not work between the Trust zone and a user-defined zone.
- ☑ Interface-based NAT does not work on interfaces bonded to the Untrust zone even though it can be enabled.
- ☑ Only limited amounts of MIP can be created. To address the scalability of one-to-one mapping, use policy-based NAT.
- ☑ The MIP address needs to be on the same subnet as the interface's subnet but cannot overlap with another IP(s) within the same subnet (in other words, you cannot use a MIP address from an IP address already defined within an existing DIP pool). However, there are two advantages if the interface is set up on the Untrust zone: First, on the smaller appliances (HSC, 5XT, 5GT), the existing interface IP that is bonded to the Untrust zone as a MIP address can be used. Second, a MIP address on a different subnet than the Untrust interface is allowed. This requires that a route be defined to indicate that traffic destined for that MIP address subnet must go through the NetScreen Untrust interface.

- ☑ When PAT is disabled in a DIP pool, the IP pool assignment remains the same for the host for all concurrent sessions. When PAT is enabled, the IP pool assignments rotate in a round-robin fashion for each new session.
- ☑ No options exist for PAT in a DIP shift configuration.

Destination NAT

- ☑ VIP only works on interfaces bonded to the Untrust zone.
- ☑ The VIP address has to be on the same subnet as the Ethernet interface where the VIP is defined.
- ☑ Custom ports can be created for VIP mappings.
- ☑ The existing IP assignment on the Ethernet interface can be used. During VIP creation, select **Same as the Untrusted interface IP address**. This feature is limited to only the HSC, 5XT, 5GT, 25, and 50 models.
- ☑ When you areusing policy-based destination NAT, it is important to understand the Juniper packet flow—specifically that a route lookup is performed *before* and *after* a policy-based destination NAT.
- ☑ PAT is not used on the source port during a policy-based destination NAT. PAT can be used for the destination.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: What are the advantages of using NAT?

A: NAT conserves IP addresses, provides a hidden identity for host(s), has the ability to use nonroutable addresses from the RFC 1918 space, addresses overlapping subnets, and maintains a cohesive network.

Q: What is the difference between a MIP and a VIP?

A: MIP provides a one-to-one static NAT function whereas VIP provides a one-to-many NAT function.

Q: What are the advantages of using policy-based NAT over interface-based NAT?

A: The number one reason to choose policy-based NAT over interface-based NAT is the scalability. With interface-based NAT you are limited to only performing address translation in one flow direction, only the source address can be translated, you cannot turn off PAT, and it requires all ingress traffic to be translated. With policy-based NAT you can uniquely define address translation on a per-firewall rule definition, giving you the ability to control address translation flows and to perform source and/or destination translation, as well as the ability to turn PAT on/off.

Q: Can interface-based NAT and policy-based NAT configuration coexist?

A: Yes. Interface-based NAT and policy-based NAT can coexist together. It is recommended that interface-based NAT be disabled (set to Route mode) and to utilize policy-based NAT for your address translation needs.

Q: What is a DIP?

A: A DIP is used for policy-based source NAT functionality. A DIP can consist of one-to-many IP ranges.

Q: MIP and VIP methods scale better then policy-based NAT. True or False?

A: False. There are software-set limitations for MIP and VIP on all Juniper security appliance and system products. This is the same with policy-based NAT, but these limitations

far exceed the capacity for MIP and VIP. Instead, the limitation is based on the number of firewall policies that can be created, which usually ranges from to 50 to 40,000.

Q: If NAT hides the identity of a system, can it substitute for a firewall rule?

A: No, NAT should never be thought of as a replacement for a good security policy. NAT is not really intended to be a security enhancement, but rather just provide address translation (although there are some inherent security benefits). You would be better off investing your time in a solid firewall rulebase that uses rules to handle the security, utilizing NAT only where needed for proper functionality.

Q: Is there a command that can display all of the software and hardware limits of my firewall?

A: Yes, there is a hidden command, *get sys-cfg*, which will display all of the limits of your firewall. This can be a very important command to make sure you are not approaching the limits of your firewall platform. The following is a sample output from issuing this command on the CLI:

```
ns5gt-adsl-wlan-> get sys-cfg
acl rule mem size number: 16384
ADSL Sub-if limit number: 10
alarm glog number: 128
arp-size number: 1024
AntiSpam Black/White List Size number: 500
AntiSpam SBL Request Queue Size number: 1000
Asic based forwarding supported number: 0
b_list number: 1024
max bgp ext change number: 50
max bgp int change number: 50
max bgp purge rib max number: 50
max routes redistributed into bgp at a time. if there are more they are handled
in next iteration number: 55
max bgp fdb change update number: 50
5xt combined mode number: 1
config glog number: 32
default period of timeout in cryptlib number: 200
default h323 call num number: 8
dbuf number: 32768
default mgcp call num number: 8
default PPTP call num number: 8
default root security zone number: 9
default rtsp connections number: 8
```

```
default sccp call num number: 8
default sip call num number: 8
default syn-ack-ack threshold number: 512
anti virus req queue size number: 400
def apppry conn number: 2000
def icap conn number: 256
def icap default max connections allowed number: 32
def icap server groups number: 2
def icap servers number: 2
dlog buf max chunk number: 256
dlog queue max chunk number: 4
dlog session log pool number: 256
DRP enable number: 1
The maximum number of dynamic allocate net pak number: 24117248
emav def decompress layer number: 2
emav def max concurrent msgs number: 16
emav def max content size number: 10000
emav max decompress layer number: 4
emav max max content size number: 10000
emav max max concurrent msgs number: 16
emav max pattern size number: 10000000
emav max queue size number: 16
emav max resource size number: 32000
encap number: 256
esp alg ib hash array size number: 16
esp alg ob hash array size number: 16
esp alg pending hash array size number: 16
event alarm entry number: 2048
event log entry number: 1024
5gt extended mode number: 1
extra dialup vpn number: 0
Extra VR number: 1
extra zones number: 0
fcb hash size number: 4096
frag control block number: 4096
Frame-Relay Sub-if limit number: 0
GTP enable number: 0
HA Mode number: 1
IDP Database update number: 1
IDP Enable number: 1
idp memory pool chunk number number: 1280
ike alg hash array size number: 16
```

```
ike alg self hash array size number: 16
info glog number: 64
internal glog number: 128
ipsec alg free list size number: 200
L2 A-A enable number: 0
mac table size number: 1024
total max address book entries number: 512
total max addr group allowed number: 32
max admin users number: 20
total aggregate interface number: 0
max attack entries allowed per group number: 32
max attack groups per policy number: 64
max authentication servers number: 4
max BGP instances number: 3
max BGP peers number: 10
max br node number: 4096
max binary br entry number: 4096
maximum conncurrent active auth users number: 4096
total dialer interface number: 0
max dns cache size number: 256
max dns proxy concurrent client sessions number: 32
max dns proxy sel table size number: 32
max number of eap session number: 255
max entries allowed per group number: 32
max group expressions number: 5
max h323 call num number: 16
total hardware interface number: 32
max idp ttys number: 0
max IGMP groups per unit number: 130
max IKE peer gateways number: 1024
shared interface number: 10
max mal url length number: 64
max mgcp call num number: 16
max root mip number: 301
max vsys mip number: 0
total multilink interface number: 0
max multicast policy num per unit number: 10
max Multicast routing table owners number: 8
max num of NAS objects number: 1024
session number: 4064
max number memory threshold for nslib memory allocation number: 0
max mal url entries number: 48
```

```
net user number: 0
max output interfaces in multicast route number: 2
max OSPF instances number: 3
maxium packets per interrupt number: 128
maximum number of PBR policies number: 16
physical ports number: 16
max PIM instances number: 2
max falsh storage available to PKI number: 139712
max crl size allowed to save to flash number: 105600
max crl size allowed to save in ram number: 10485760
policy number: 100
max port node number: 5120
max pport dst number: 1
max pport number: 4048
max PPTP call num number: 8
max precalc pki prime number: 3
total redundant interface number: 0
maximum number of redistribution routes into OSPF, BGP number: 512
max RIP instances number: 3
max rm resource client items number: 6
max rm resource group items number: 32
max rm resource items number: 80
Maximum number of route maps in the system number: 16
max rtsp connections number: 8
sbuf number: 8192
max sccp call num number: 16
max secondary ip number: 4
max security zone number: 16
max service groups number: 32
max custom service number: 128
max sip call num number: 16
stats number: 1024
max system zone id number: 99
config size number: 524288
ooo segments number: 32
sockets number: 64
maximum tftp buffer size number: 9437184
max number of objects to add token per tic number: 32
max ttys number: 6
tunnel interface number: 10
The maximum number of URLs/IP addresses in a category number: 20
The maximum number of categories in a URL Filtering profile number: 15
```

```
The maximum number of URL filtering profiles number: 10
The maximum number of user-defined categories number: 30
url blocking cache number: 0
url blocking servers number: 1
max users number: 100
max vip number: 4
max vlan number: 0
vpn number: 256
policy entries per vsys number: 0
service entries per vsys number: 0
vpn entries per vsys number: 0
max vsys number: 1
max users per vsys number: 0
max X509 object number: 142
max zone number: 128
mgcp transaction hash size number: 64
mip configurable on tagged sub-interface number: 0
nat arp size number: 32
max nat cookie number: 128
max nat gate number: 256
TCP reassembly buffer number: 100
nd6-size number: 1024
nbuf list size number: 1024
Non-tagged Sub-if limit number: 2
4.0 config size number: 262144
max OSPF interfaces number: 32
max OSPF interfaces per one area number: 16
max OSPF routes processed per timer tic number: 5
max OSPF nodes processed in SPF per timer tic number: 5
max address book entries for other zone number: 512
max addr group allowed for other zone number: 32
packet log entry number: 512
policy counter number: 125
rip max interface number: 16
route entry number: 1024
Number of Entries in global RPC Mapping Table Pool number: 512
Session soft limit number: 4064
max multicast routes number: 1024
maximum RIP neighbors per interface number: 16
sys_res glog number: 32
traffic alarm entry number: 4096
traffic log entry number: 4096
```

```
traffic glog number: 512
URL Filtering Cache Size(K) for SurfControl number: 500
URL Filtering Cache Size Limit(K) for SurfControl number: 1000
URL Filtering Request Queue Size for SurfControl number: 512
Url Filtering SC cpa enable number: 1
url query queue number: 40
User soft limit number: 4294967295
VLAN Soft limit number: 10
vpn soft limit number: 25
VSYS soft limit number: 0
root well-known zone max address book entries number: 512
root well-known zone max addr group allowed number: 32
```

Chapter 9

Transparent Mode

Solutions in this chapter:

- Interface Modes
- Understanding How Transparent Mode Works
- Configuring a Device to Use Transparent Mode
- Transparent Mode Deployment Options

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

As some point in their careers, network administrators will need to reassess their current network deployments and determine if they are designed to meet the needs of their growing companies. Perhaps, when the company was a small startup, it was convenient to have the database, the Web server, and the user community on the same network. But as a company grows and services are added, and additional resources continually exposed to various parts of the internal and external infrastructure, the administrator will probably wonder if this environment is really benefiting the company or just creating unnecessary risk to corporate information assets. Once administrators make this decision, they are confronted with the possibility of added complexity as well as the cost of making the recommended changes.

One solution that can help address these possible issues is the *transparent* mode capability of a Juniper firewall. Transparent mode provides the capability to convert a Juniper firewall from a layer 3 device to a layer 2 device. Rather than requiring the administrator to redesign the entire network for physical and network changes to servers and devices, he or she has the option to implement a flexible alternative that can help to simplify deployment efforts and reduce the costs.

Interface Modes

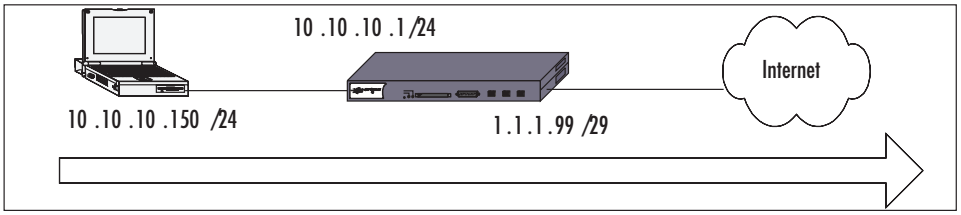
Before we talk about transparent mode, let's quickly review the other interface modes on a Juniper firewall. The interfaces on a Juniper firewall can operate in three different modes: NAT (Network Address Translation), route, and transparent. The following is a review of NAT and route modes.

NAT Mode

As described in Chapter 8, NAT can be configured at the interface or through policy. When an interface is placed in NAT mode, the Juniper device replaces the private, unroutable IP (Internet Protocol) address of the host with the IP address of the interface in the *Untrust* zone. Additionally, the source port number is replaced by a random port generated on the firewall. By doing so, NAT provides an additional layer of security by never directly exposing a resource on a trusted zone to one in an untrusted zone.

Figure 9.1 shows an example of NAT mode. In this sample, the packet originating from 10.10.10.150 is translated to the Untrust IP address of 1.1.1.99. All Internet traffic will appear as if it is coming from the translated address.

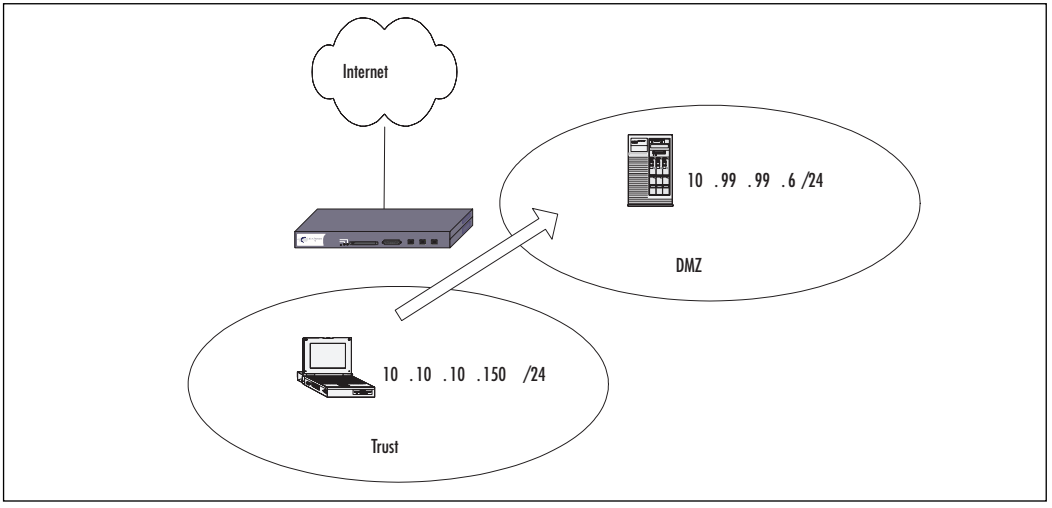
Figure 9.1 Traffic from Private IP to Internet



Route Mode

With route mode, the device passes traffic from one zone to another without performing NAT translation. This means that the source IP address and port remain unchanged as a packet passes through one zone to another, as shown in Figure 9.2. In this example, the traffic passing from the Trust zone to the DMZ (demilitarized zone) is not translated and the DMZ servers can see the IP address 10.10.10.150 as the source.

Figure 9.2 Traffic from Trust Zone to DMZ Zone

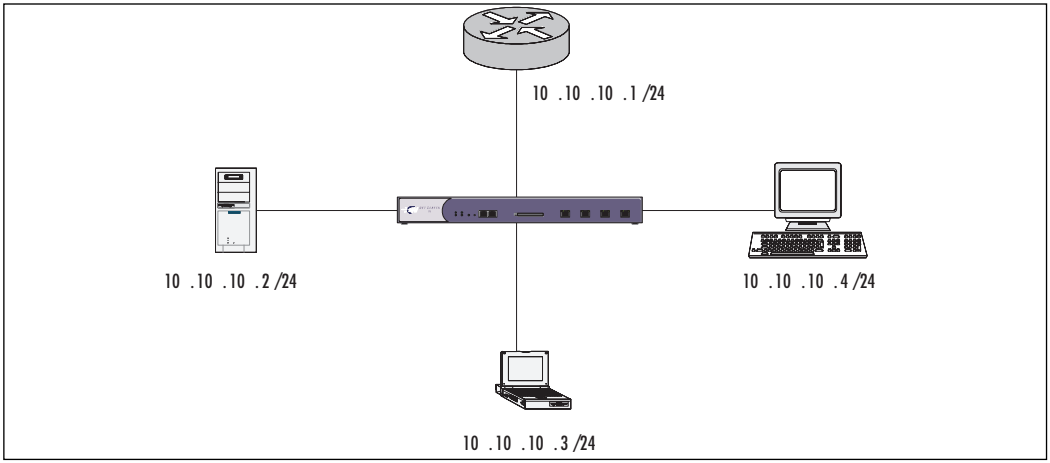


Understanding How Transparent Mode Works

With transparent mode, the Juniper firewall is converted from a layer 3 device to one that operates at layer 2, essentially becoming a layer 2 bridge. By doing so, the device can be deployed into existing infrastructures without requiring the readdressing that would be required for a routed solution. The IP addresses of the physical interfaces are set to 0.0.0.0/0

and truly make the deployment invisible to the user. Figure 9.3 provides an example for segmenting internal resources using transparent mode.

Figure 9.3 Juniper Device in Transparent Mode



In this example, notice that all of the networked devices are part of the same subnet. By converting the device to transparent mode, an existing subnet can be segmented, providing the ability to secure resources with firewall policy. If the firewall was to remain in layer 3 mode, it would have to route the traffic, which would require subnet and network changes. Transparent mode is the easiest method to use since it requires the fewest infrastructure changes.

How Transparent Mode Works

Transparent mode converts the firewall device from its default layer 3 route mode to what is essentially a layer 2 bridge. A Juniper firewall comes predefined with three layer 2 zones, which are applied to the physical interfaces. In addition, there is a single VLAN (virtual local area network) zone that hosts a virtual interface for management and VPN (virtual private network) termination. Once all interfaces have been converted to the layer 2 zones, the device is considered to be in transparent mode. It is important to note that at this time, a Juniper device cannot operate in a mixed configuration of transparent and route modes.

Layer 2 Zones

The three predefined zones included with a Juniper firewall are V1-Trust, V1-Untrust, and V1-DMZ, and are all part of the same broadcast domain. Also, interfaces assigned to these zones must be a part of the same subnet. It is these zones that are used when defining policy between network resources.

VLAN Zone

A VLAN zone is a predefined zone that hosts the virtual interface named VLAN1. Like the layer 2 zones, the VLAN1 interface is part of the same subnet, but unlike the layer 2 zones, it can be assigned an address for VPN tunnel termination as well as another IP address for management. In addition, the VLAN zone can be used in policy to protect the interface.

Broadcast Methods

It is important to understand how a Juniper firewall forwards traffic when it is acting as a layer 2 bridge. Without policy, no traffic can pass through the device. Once a policy is applied, it permits traffic to flow based on the permitted services, as well as allowing ARP (Address Resolution Protocol) and non-IP-based layer 2 traffic, such as spanning tree. However, IP-based layer 2 traffic and IPSec are not permitted by default. Managing these services will be discussed later.

When a network host does not know the MAC (Media Access Control) address associated with a particular IP, the host will find the MAC by performing an ARP query. The requesting host will flood all networked devices with the ARP query. The host that owns the MAC address will respond to the requestor with an ARP reply; all other network hosts will drop the packet. Once the requestor receives the reply, it then adds that information to its ARP cache. Juniper firewalls will also learn which interface is associated with the MAC address based on which interface receives the ARP reply. Juniper firewalls can use one of two methods for building its ARP table:

- **Flood Method** When a Juniper device receives an Ethernet frame with a destination MAC address that is not in the MAC table, the device will flood the packet out all interfaces, similar to the way most switches work. When a response is found, the Juniper device learns which interface is attached to the responding MAC address and adds the MAC and interface to its forwarding table.
- **ARP/Traceroute** When a Juniper device receives an Ethernet frame with a MAC address that is not in the MAC table, the device performs a series of actions. First, it records the MAC in the initial packet and then drops that packet. Next, the Juniper device generates two packets, one for an ARP query and another for a traceroute. The ARP query replaces the source MAC address from the initial packet and instead uses the MAC address of the VLAN1 interface. The traceroute packet is an ICMP echo request with a Time To Live (TTL) of 1. The two generated packets are then flooded out to all interfaces except for the one that received the initial packet. If the ARP reply is found on a device in the same subnet, the Juniper device learns that MAC and forwards traffic to the appropriate interface. If the IP address of the packet exists in a different subnet, the traceroute packet returns with the IP address and MAC of the router that the packet must pass through. The Juniper device then learns the router's MAC and forwards traffic accordingly.

ARP/traceroute is considered the most secure broadcast method since it does not flood the initial packet out on all interfaces; rather, the Juniper device floods the ARP queries and traceroute packets. It should also be noted that the traceroute packet is optional. If it is not used, then the Juniper will only be able to learn the destination MAC if it is in the same subnet as the received packet. The traceroute option is turned on by default.

Configuring a Device to Use Transparent Mode

By default, a Juniper firewall is configured as a layer 3 device. Switching to layer 2 is a simple matter of moving the interfaces into the layer 2 zones. However, before the interfaces are moved, it is important to configure the VLAN1 interface. This will ensure you are able to manage the device across the network once it has been converted.

VLAN1 Interface

As mentioned, all Juniper devices have a VLAN1 interface. Using the *get int* command will show a list of all physical and virtual interfaces, including the VLAN1 interface (see Figure 9.4).

Figure 9.4 Interfaces on a Juniper 204

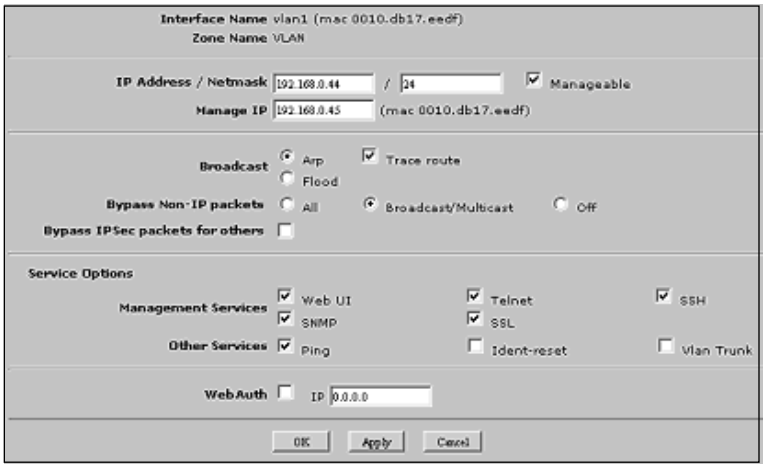
```
ns204-> get int

A - Active, I - Inactive, U - Up, D - Down, R - Ready

Interfaces in vsys Root:
Name            IP Address          Zone         MAC              VLAN State VSD
eth1            192.168.1.1/24      Trust        0010.db32.3500     -   D   -
eth2            0.0.0.0/0           DMZ          0010.db32.3505     -   D   -
eth3            0.0.0.0/0           Untrust      0010.db32.3506     -   D   -
eth4            0.0.0.0/0           HA           0010.db32.3507     -   D   -
vlan1           0.0.0.0/0           VLAN         0010.db32.350f     1   D   -
```

Figure 9.4 displays the default configuration of a Juniper 204 firewall. As shown, the VLAN1 interface is listed along with its MAC address. Configuring an IP address on VLAN1 is the same as for any other interface. If you are using the command line, enter **set int vlan1 ip 192.168.0.44/24**. If you are using the WebUI, access **Network** | **Interfaces** | **VLAN1**, then enter the IP address. Figure 9.5 displays the available VLAN1 options from the WebUI.

The following discusses the options that are unique to the VLAN1 interface:

- **Broadcast** As previously discussed, this is where the administrator must decide what broadcast method is going to be used. The default method is Flood.

Figure 9.5 WebUI Configuration for VLAN1 Interface



From the CLI:

```
set int vlan1 broadcast arp
unset int vlan1 broadcast arp trace-route
set int vlan1 broadcast flood
save
```

- **Bypass Non-IP packets** *All* will permit all non-IP traffic, like IPX, to pass while in transparent mode. *Broadcast/Multicast* will only allow broadcast and multicast to pass, while *off* disables this feature altogether. By default, *Broadcast/Multicast* is selected.

  From the?CLI:

```
set int vlan1 bypass-non-ip
set int vlan1 bypass-non-ip-all
save
```

- **Bypass IPSec packet for others** By enabling this option, the Juniper device will allow IPSec traffic to pass. This would allow a VPN device sitting behind the Juniper to terminate the traffic.

  From the?CLI:

```
set int vlan1 bypass-others-ipsec
save
```

- **Vlan Trunk** All other service options are the same as a physical interface, with the exception of Vlan Trunk. By default, if a packet is received with an 802.1q

VLAN tag, the Juniper will drop the traffic. This option must be enabled in order to allow the traffic to pass on a trunk port. It is important to note that a Juniper cannot currently perform 802.1q tagging while in transparent mode, the tagging can only be passed. If 802.1q tagging is required, the device must be deployed in route mode.

By default, the VLAN1 interface is only accessible from the V1-Trust zone. In order to manage the device from a zone other than the V1-Trust, management must be enabled on the layer 2 interface of that zone. This can only be accomplished from the command line by using the following:

```
set int ip V1-Untrust manage
save
```

Converting an Interface to Transparent Mode

Once the VLAN1 interface is configured, the device is ready to be deployed in transparent mode. This is done by moving all interfaces from layer 3 zones to the layer 2 zones. Before an interface can be moved to the new zone, the IP address on the interface needs to be set to 0.0.0.0/0. Once completed, the interface can then be moved to the layer 2 zone. This will have to be completed for all interfaces that will participate in the subnet. This excludes interfaces dedicated for high availability.

From the WebUI:

1. Go to Network | Interfaces | ethernet1.
2. Select Zone Name and enter V1-Trust.
3. In the IP Address/Netmask option, fill in 0.0.0.0/0.
4. Go to Manage IP and fill in <*Blank*>.

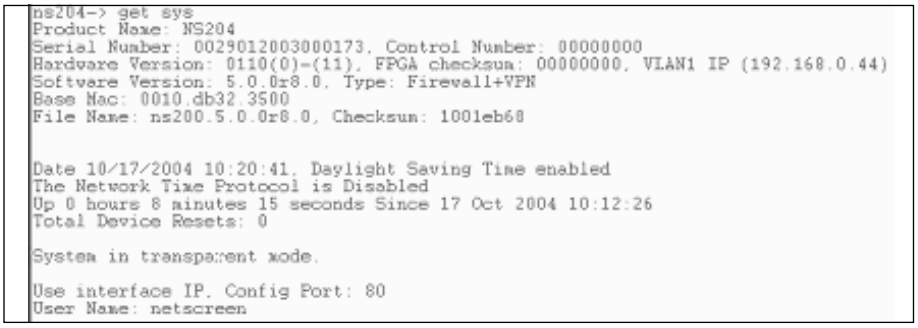
From the CLI:

```
unset int eth1ip
set int eth1 zone V1-Trust
save
```

Once all participating interfaces are moved to the layer 2 zones, confirm the device is in transparent mode by entering **get sys** from the command line. A sample output of this command is shown in Figure 9.6.

From the WebUI, transparent mode can be confirmed by verifying participating interfaces are in layer 2 zones (see Figure 9.7).

Figure 9.6 System Information Indicating a Juniper 204 Is in Transparent Mode



```
ns204-> get sys
Product Name: NS204
Serial Number: 0029012003000173, Control Number: 00000000
Hardware Version: 0110(0)-(11), FPGA checksum: 00000000, VLAN1 IP (192.168.0.44)
Software Version: 5.0.0r8.0, Type: Firewall+VPN
Base Mac: 0010.db32.3500
File Name: ns200.5.0.0r8.0, Checksum: 1001eb60


Date 10/17/2004 10:20:41, Daylight Saving Time enabled
The Network Time Protocol is Disabled
Up 0 hours 8 minutes 15 seconds Since 17 Oct 2004 10:12:26
Total Device Resets: 0

System in transparent mode.

Use interface IP, Config Port: 80
User Name: netscreen
```

Figure 9.7 Interface Screen from WebUI Indicating the Interfaces Are in Layer 2

Name	IP/Netmask	Zone	Type	Link	Configure	
ethernet1	0.0.0.0/0	V1-Trust	Layer2	up	Edit	
ethernet2	0.0.0.0/0	V1-DMZ	Layer2	down	Edit	
ethernet3	0.0.0.0/0	V1-Untrust	Layer2	down	Edit	
ethernet4	0.0.0.0/0	HA	Layer3	down	Edit	
vlan1	192.168.0.44/24	VLAN	Layer3	up	Edit	

Creating a Custom Layer 2 Zone and Network Object

Creating a custom layer 2 zone is just like creating a custom layer 3 zone. When naming the zone, the name must be prefaced with **L2-**.

From the WebUI:

1. Go to **Network | Zones** and press **New**.
2. Select **Zone Name** and fill in **L2-Test**.
3. Go to **Zone Type** and fill in **Select Layer 2**.

From the CLI:

```
set zone name L2-Test L2 1
save
```

From the WebUI:

1. Go to **Objects | Addresses | List | Select Zone** and press **New**.
2. In the **Address Name** option, fill in **TestObject**.
3. Got to **IP/Netmask** and type **10.10.10.40/24**.

4. Select **Zone**, and fill in **L2-Test**.

From the CLI:

```
set address L2-Test TestObject 10.10.10.40/24

save
```

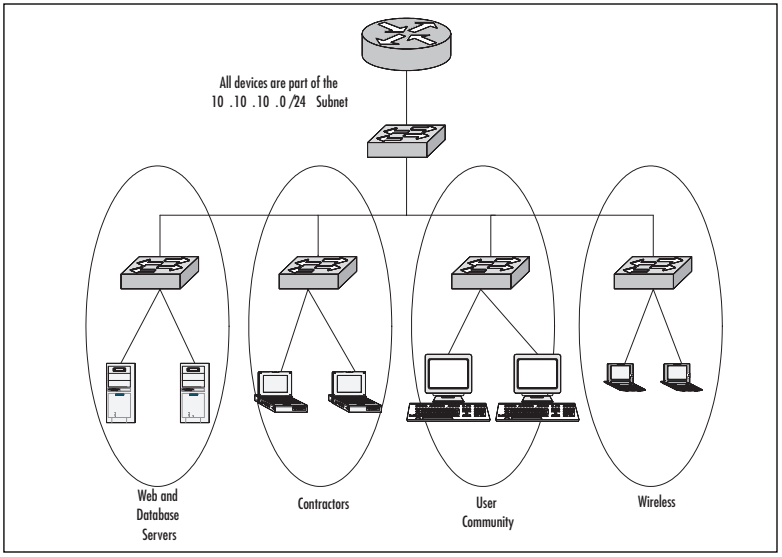
Transparent Mode Deployment Options

Transparent mode provides a great deal of flexibility by providing additional ways to protect the network with a firewall. The following are examples of several types of deployments along with the steps for configuration.

Network Segmentation

In this first example, let's refer to the example used in the beginning of the chapter. A small company has grown and network resources are exposed to unnecessary risk. Cost and time are factors, so keeping the design simple and effective is critical. Figure 9.8 provides an example of what this network could look like.

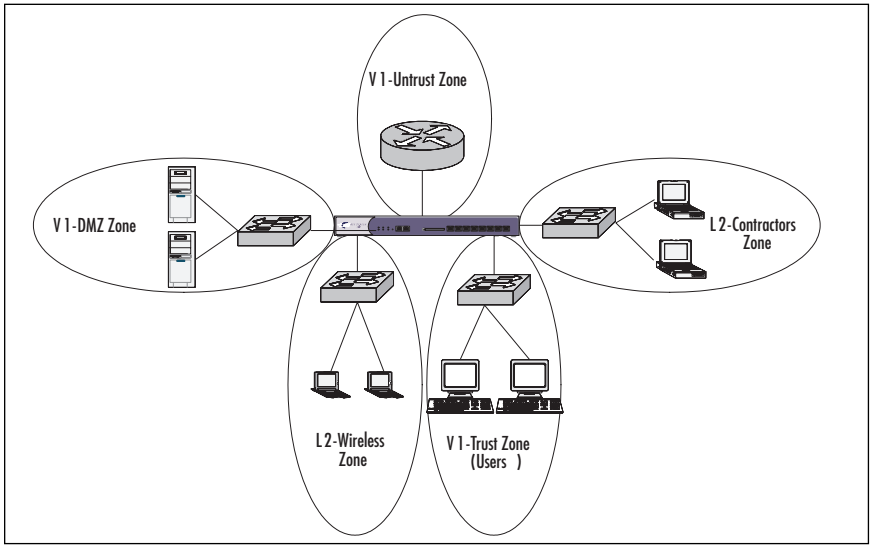
Figure 9.8 A Single Subnet without Segmentation



As noted in Figure 9.8, all devices exist in the same subnet, 10.10.10.0/24. There is no policy protecting the Web and database servers from the contractors, wireless, or general user community. The contractors are mobile laptops and provide no way for the administrator to guarantee their security. Wireless devices have their own security issues and allowing these devices open access to the network creates a great risk for company assets. Even though the

general user community could have antivirus software running, it does not make them 100-percent secure from a worm outbreak. Inserting a Juniper firewall in transparent mode into the mix could help to mitigate the risk, while keeping the required network changes minimal and invisible to the users (Figure 9.9).

Figure 9.9 The Same Subnet with Transparent Mode Segmentation



This example makes use of a Juniper-208 to achieve segmentation. Let's walk through the steps that would be required for this type of configuration.

First configure the VLAN1 interface with an IP address of 10.10.10.2 and select the management services for WebUI, Telnet, SSH, and Ping (see Figure 9.10).

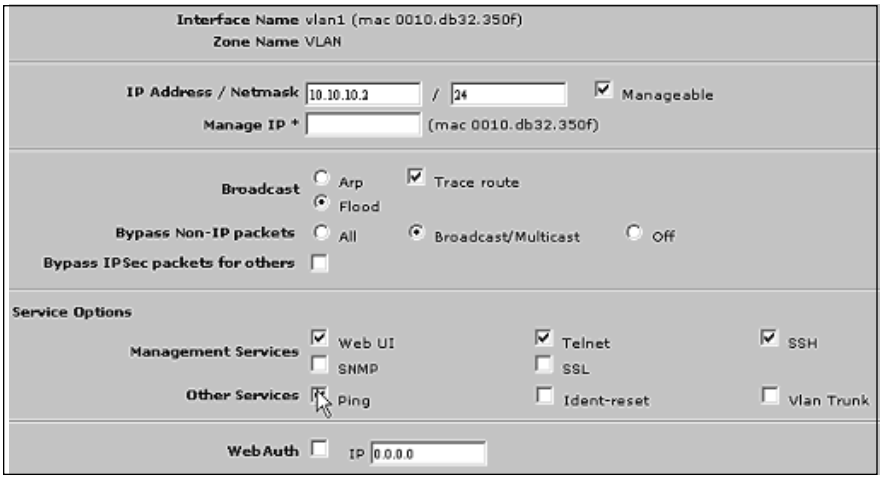
From the WebUI:

1. Go to **Network | Interfaces**. Select **Edit** for **VLAN1**.
2. In **IP Address/Netmask**, type **10.10.10.2/24**.
3. Go to **Service Options** and select **WebUI, Telnet, and SSH**.
4. In the **Other Services** section, select **Ping**.

From the CLI, type in:

```
set int vlan1ip 10.10.10.2/24
set int vlan1manage web
set int vlan1manage telnet
set int vlan1manage ssh
set int vlan1manage ping
save
```

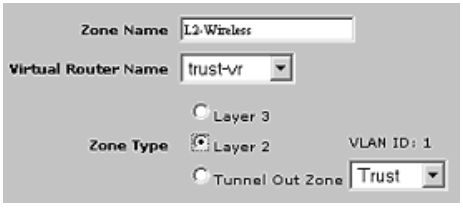
Figure 9.10 VLAN1 Interface



Now create custom zones for Wireless and Contractors.
From the WebUI:

1. Go to **Network | Zones | New**.
2. Select **Zone Name** and fill in **L2-Wireless** (see Figure 9.11).
3. Select **Zone Type** and type **Layer 2**.

Figure 9.11 L2-Wireless Zone



4. Go to **Network | Zones | New**.
5. Select **Zone Name** and fill in **L2-Contractors**.
6. Select **Zone Type** and type **Layer 2**.

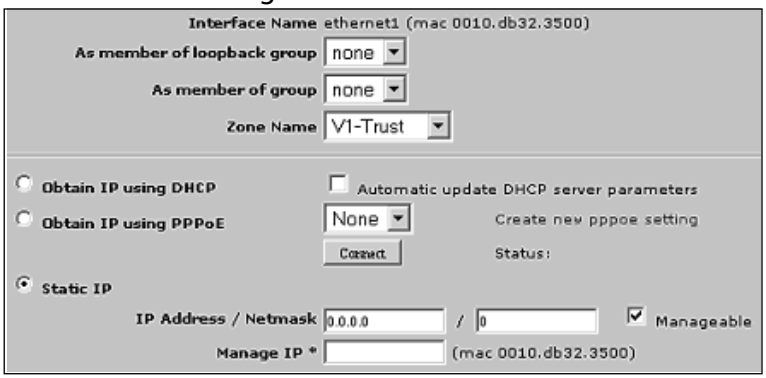
From the CLI, type in:

```
set zone name L2-Wireless L2 1
set zone name L2-Contractors L2 1
save
```

The next step is to move interfaces into Layer 2 zones.
From the WebUI:

1. Go to **Network | Interfaces | Ethernet 1**.
2. Select **Zone Name** and fill in **V1-Trust**.
3. In the **IP Address/Netmask** option, type **0.0.0.0/0** (see Figure 9.12).

Figure 9.12 Ethernet 1 Configured for V1-Trust



4. Go to **Network | Interfaces and select Ethernet 2**.
5. Go to **Zone Name and fill in V1-DMZ**.
6. Go to **IP Address/Netmask and fill in 0.0.0.0/0**.
7. Go to **Network | Interfaces | Ethernet 3**.
8. Go to **Zone Name: V1-Untrust**.
9. Go to **IP Address/Netmask: 0.0.0.0/0**.
10. Go to **Network | Interfaces | Ethernet 4**.
11. Go to **Zone Name: L2-Wireless**.
12. Go to **IP Address/Netmask: 0.0.0.0/0**.
13. Go to **Network | Interfaces | Ethernet 5**.
14. Go to **Zone Name: L2-Contractors**.
15. Go to **IP Address/Netmask: 0.0.0.0/0**.

From the CLI:

```
unset int eth1 ip
unset int eth2 ip
unset int eth3 ip
unset int eth4 ip
unset int eth5 ip
set int eth1 zone V1-Trust
set int eth2 zone V1-DMZ
set int eth3 zone V1-Untrust
```

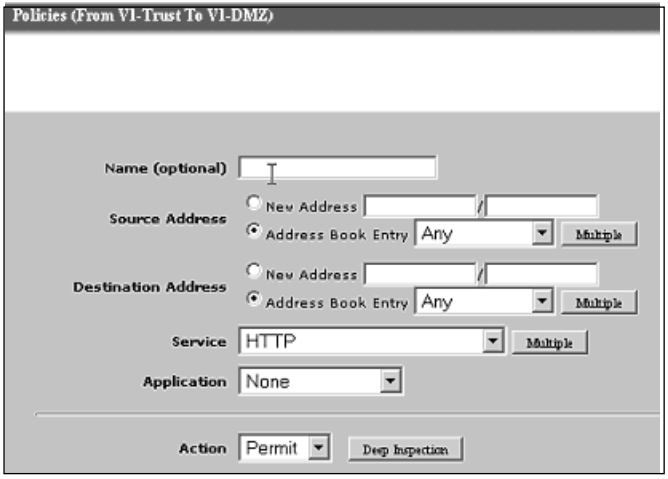
```
set int eth4 zone L2-Wireless
set int eth5 zone L2-Contractors
save
```

Now configure policies to allow V1-Trust, L2-Wireless, and L2-Contractors HTTP access to the L2-DMZ zone.

Steps from the WebUI:

1. Go to **Policies | From: V1-Trust, To: V1-DMZ**. Select **New**.
2. Select **Service** and type **HTTP** (see Figure 9.13).

Figure 9.13 Policy for V1-Trust to V1-DMZ for HTTP



3. Go to **Policies | From: L2-Wireless, To: V1-DMZ**. Select **New**.
4. Select **Service** and type **HTTP**.
5. Go to **Policies | From: L2-Contractors, To: V1-DMZ**. Select **New**.
6. Select **Service** and fill in **HTTP**.

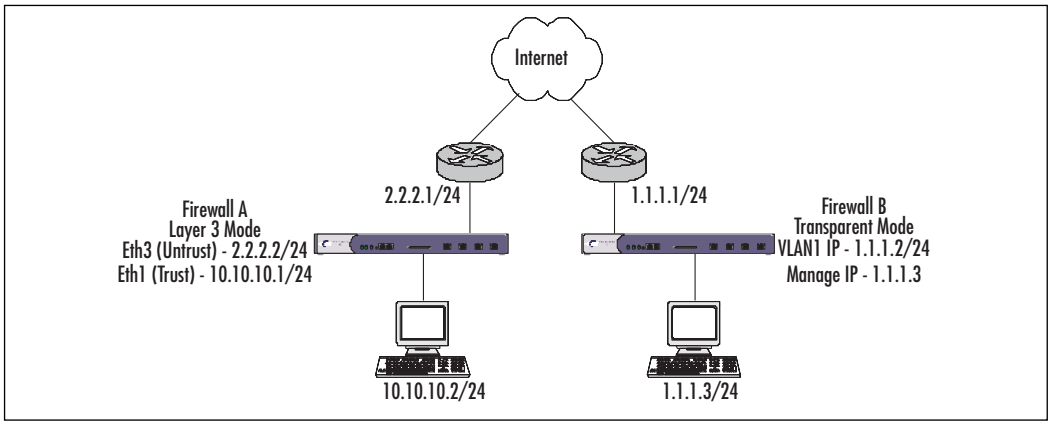
Steps from the CLI:

```
set pol from V1-Trust to V1-DMZ any any http permit
set pol from L2-Wireless to V1-DMZ any any http permit
set pol from L2-Contractors to V1-DMZ any any http permit
save
```

VPNs with Transparent Mode

Even in transparent mode, the Juniper device is still able to terminate policy-based VPNs. In Figure 9.14, there are two Juniper 25s establishing a VPN tunnel across the Internet. Firewall A is running in Layer 3 route mode, while firewall B is running layer 2 transparent mode. The following details the steps involved with configuring this scenario.

Figure 9.14 VPN Deployment with a Juniper Device in Transparent Mode



First configure interface IP addresses and default route.

From Firewall A:

Steps from the WebUI:

1. Go to **Network | Interfaces | Ethernet 1**. In the **Static IP** option, fill in **10.10.10.1/24**.
2. Go to **Network | Interfaces | Ethernet 3**. Again, select **Static IP** and type **2.2.2.2/24**.
3. Go to **Network | Routing | Routing Entries | Trust-VR** and press **New**. Fill in:

   Network Address/Netmask: **0.0.0.0/0**

   Select **Gateway**

   Interface: **ethernet3**

   Gateway IP Address: **2.2.2.1**

Steps from the CLI:

```
set int eth1 ip 10.10.10.1/24
set int eth3 ip 2.2.2.2/24
set route 0.0.0.0/0 int eth3 gate 2.2.2.1
save
```

From Firewall B:

Steps from the WebUI:

1. Go to **Network | Interfaces | VLAN1**.
2. Select **IP Address/Netmask**. Type **1.1.1.2/24**.
3. Select **Manage IP**. Type **1.1.1.3**.

4. Go to **Network | Interfaces | Ethernet 1**.
5. Select **Zone Name**. Type **V1-Trust**.
6. Select **IP Address/Netmask**. Type **0.0.0.0/0**.
7. Go to **Network | Interfaces | Ethernet 2**.
8. Select **Zone Name**. Type **V1-DMZ**.
9. Select **IP Address/Netmask**. Type **0.0.0.0/0**.
10. Go to **Network | Interfaces | Ethernet 3**.
11. Select **Zone Name**. Type **V1-Untrust**.
12. Select **IP Address/Netmask**. Type **0.0.0.0/0**.
13. Go to **Network | Routing | Routing Entries | Trust-VR** and press **New**.

Fill in:

Network Address/Netmask: 0.0.0.0/0

Select Gateway

Interface: VLAN1

Gateway IP Address: 1.1.1.1

Steps from the CLI:

```
set int vlan1 ip 1.1.1.2/24
set int vlan1 manage-ip 1.1.1.3
unset int eth1 ip
unset int eth2 ip
unset int eth3 ip
set int eth1 zone V1-Trust
set int eth2 zone V1-DMZ
set int eth3 zone V1-Untrust
set route 0.0.0.0/0 int vlan1 gate 1.1.1.1
save
```

Now configure objects and a VPN for both firewalls.

From Firewall A:

From the WebUI:

Go to **Objects | Addresses | List | Select Trust** and select **New**.

Fill in:

Address Name: **FirewallA_Local**

IP/Netmask: **10.10.10.2/32**

Go to **Objects | Addresses | List | Untrust** and choose **New**.

Fill in:

Address Name: **FirewallB_Remote**

IP/Netmask: **1.1.1.2/32**

Go to **VPNs | AutoKey Advanced | Gateway** and select **New**.
Fill in:

Gateway Name: **p1-VPN**

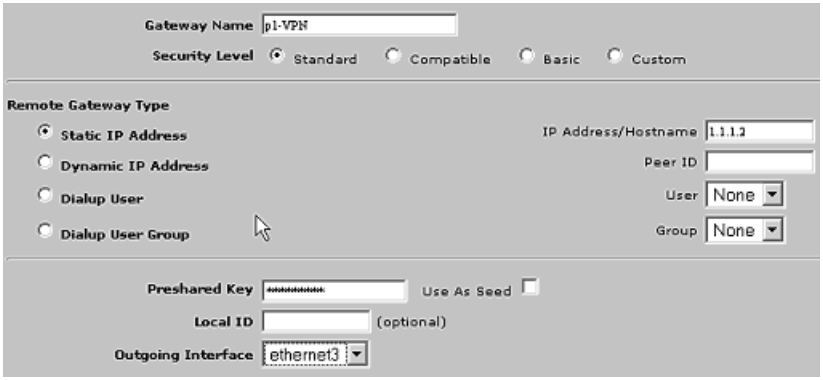
Security Level: **Standard**

Static IP Address: **1.1.1.2**

Preshared Key: **Juniper**

Outgoing Interface: **ethernet3** (see Figure 9.15)

Figure 9.15 VPN Configuration for Gateway (Phase 1)



Go to **VPNs | AutoKey IKE** and select **New**.
Fill in:

VPN Name: **p2-VPN**

Security Level: **Standard**

Predefined: **p1-VPN** (see Figure 9.16)

From the CLI:

```
set address trust FirewallA_Local 10.10.10.2/32
set address untrust FirewallB_Remote 1.1.1.2/32
set ike gateway p1-vpn address 1.1.1.2 main outgoing-interface ethernet3
preshare Juniper sec-level standard
set vpn p2-vpn gateway p1-vpn tunnel sec-level standard
save
```

Figure 9.16 VPN Configuration for AutoKey IKE (Phase 2)

From Firewall B
From the?WebUI:

Go to **Objects | Addresses | List**, select **V1-Trust** and then **New**.

Fill in:

Address Name**: FirewallB_Local**

IP/Netmask: **1.1.1.3/32**

Go to **Objects | Addresses |List**, select **V1-Untrust** and then **New**.
Fill in:

Address Name**: FirewallA_Remote**

IP/Netmask: **10.10.10.2/32**

Go to **VPNs | AutoKey Advanced | Gateway**, and then select **New**.
Fill in:

Gateway Name: **p1-VPN**

Security Level: **Standard**

Static IP Address: **2.2.2.2**

Preshared Key: **Juniper**

Outgoing Interface: **V1-Untrust**

Go to **VPNs | AutoKey IKE** and select **New**.
Fill in:

VPN Name: **p2-VPN**

Security Level: **Standard**

Predefined: **p1-VPN**

From the CLI:

```
set address V1-Trustrust FirewallB_Local 1.1.1.3/32
set address V1-Untrust FirewallA_Remote 10.10.10.2/32
set ike gateway p1-vpn address 2.2.2.2 main outgoing-interface V1-Untrust
preshare Juniper sec-level standard
set vpn p2-vpn gateway p1-vpn tunnel sec-level standard
save
```

Now configure bidirectional VPN policies for both firewalls.

From Firewall A:
From the WebUI:
Go to **Policies | From Trust to Untrust**, and then select **New**.
Fill in:

Source Address Book Entry: **FirewallA_Local**

Destination Address Book Entry: **FirewallB_Remote**

Service: **Any**

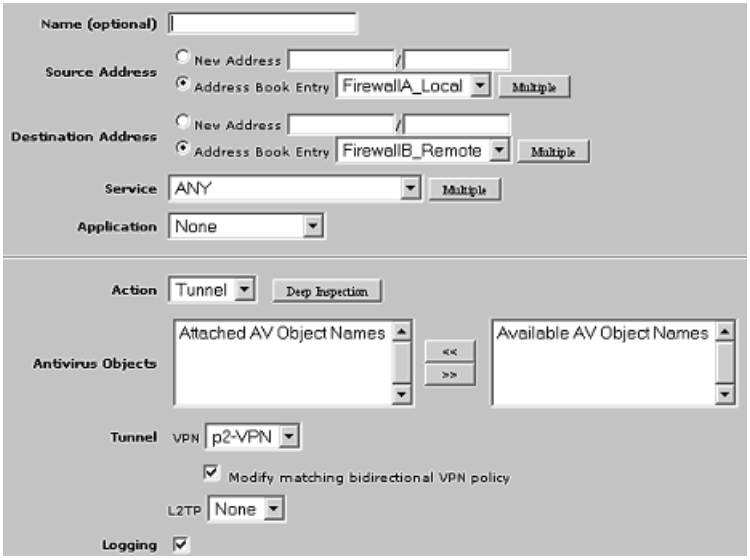
Action: **Tunnel**

Tunnel VPN: **p2-VPN**

Modify matching bidirectional policy: Checked

Logging: Checked (see Figure 9.17)

Figure 9.17 Policy Configuration for VPN



From the CLI:

```
set policy from trust to untrust FirewallA_Local FirewallB_Remote any tunnel vpn
p2-VPN log
set policy from untrust to trust FirewallB_Remote FirewallA_Local any tunnel vpn
p2-VPN log
save
```

From Firewall B
From the WebUI:

Go to **Policies | From V1-Trust to V1-Untrust**, and select **New**.
Fill in:

Source Address Book Entry: **FirewallB_Local**

Destination Address Book Entry: **FirewallA_Remote**

Service: **Any**

Action: **Tunnel**

Tunnel VPN: **p2-VPN**

Modify matching bidirectional policy: Checked

Logging: Checked

From the CLI:

```
set policy from V1-Trust to V1-Untrust FirewallB_Local FirewallA_Remote any
tunnel vpn p2-VPN log
set policy from V1-Untrust to V1-Trust FirewallA_Remote FirewallB_Local any
tunnel vpn p2-VPN log
save
```

Summary

As demonstrated in this chapter, transparent mode provides a viable and cost-effective method to provide the segmentation that is required in today's networks. Transparent mode can be used to quickly create a DMZ environment and easily secure Web servers and resources, provide internal segmentation, and also a method to terminate VPN tunnels. From a user perspective, the "invisible hop" removes the pain that can be associated with dramatic changes to the network.

While the overall look and feel of a Juniper in transparent mode is not significantly different from a device operating in layer 3, it is important to understand what differences exist. NAT and routing are no longer the concern since the device operates within the same subnet as the other existing devices. The zones that are assigned to the interfaces are specifically designed to be layer 2 zones, which are then used in developing policy. The firewall is managed by making use of the virtual interface, VLAN1.

The final section of this chapter provided two examples that show off these capabilities. Referring to these examples while considering the direction of your network can provide additional alternatives that may not have been previously considered. Segmentation is extremely important and any advantage that is provided to administrators is greatly welcomed.

Solutions Fast Track

Interface Modes

- ☑ The interfaces on a Juniper firewall can operate in three different modes: NAT (Network Address Translation), route, and transparent.
- ☑ When an interface is placed in NAT mode, the Juniper device replaces the private, unroutable IP (Internet Protocol) address of the host with the IP address of the interface in the *Untrust* zone.
- ☑ With route mode, the device passes traffic from one zone to another without performing NAT translation.

Understanding How Transparent Mode Works

- ☑ The Juniper firewall operates at layer 2.
- ☑ Two methods exist for performing ARP: flood and ARP/traceroute.
- ☑ The VLAN zone hosts the VLAN1 interface.

Configuring a Device to Use Transparent Mode

- ☑ The VLAN1 interface is used for managing the device and terminating VPNs.
- ☑ Interfaces are assigned to layer 2 zones.
- ☑ Three layer 2 zones are included by default and additional ones can be created.

Transparent Mode Deployment Options

- ☑ Segment internal subnets with minimal network configuration.
- ☑ Often used to provide firewall protection for Internet-based resources.
- ☑ Can be used as a termination point for VPNs.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: What interface modes can a Juniper firewall be assigned?

A: NAT, route, and transparent modes.

Q: How is a Juniper device converted to transparent mode?

A: The interfaces are placed in layer 2 zones. By default, the layer 2 zones included with a Juniper device are V1-Trust, V1-Untrust, and V1-DMZ.

Q: How is ARP managed?

A: Juniper firewalls have two methods for supporting ARP: flood and ARP/traceroute. Simply put, flood sends the ARP query out all interfaces and learns the destination based on which interface receives the reply. The ARP/traceroute method searches for the destination by using the MAC address of the VLAN1 interface, while the traceroute provides the router destination if the packet is not part of the local subnet.

Q: What is the VLAN1 interface?

A: This is the virtual interface used to manage the Juniper firewall and terminate VPNs when in transparent mode. Management services are turned off and on just like on an interface that is running in layer 3. Like a physical interface, a management IP address can also be assigned.

Q: With transparent mode, how is policy developed?

A: Rather than using layer 3 zones, policy is simply applied by using the layer 2 zones.

FAQ

Chapter 10

Attack Detection and Defense

Solutions in this chapter:

- Understanding Attacks
- Understanding the Anatomy of an Attack
- Configuring SCREEN Settings
- Applying Deep Inspection
- Setting Up Content Filtering
- Understanding Application Layer Gateways
- Applying Best Practices

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

This chapter covers the nuts and bolts of the security features in Juniper Networks' NetScreen firewall products. As you've no doubt already discovered, these devices are packed with features that make life easier for administrators—easy-to-configure VPNs (virtual private networks), built-in DHCP (Dynamic Host Control Protocol) servers, advanced Network Address Translation (NAT) functionality, support for a wide range of routing protocols, and much more. But a firewall's primary responsibility has always been security—keeping the bad bits out, and letting the good bits in.

In addition to the strong feature set used for network administration is an equally strong set of protective tools. NetScreen firewalls have always protected owners from classic attacks such as Land, Teardrop, and other network layer-based attacks. These defensive SCREEN features allow for zone-specific settings based upon the risk factor of the facing network segment.

And while protecting at the network layer is both important and efficient, in today's world of application layer-specific attacks, it's not sufficient security coverage all by itself. Starting with tentative steps for application layer coverage in ScreenOS 4.0 with the Malicious URL feature, NetScreen firewalls now have full application layer coverage for typical Internet-facing protocols with Deep Inspection (DI), found in ScreenOS version 5.0 and later.

Combine the Application Layer Gateway features with the advanced filtering features and antivirus (AV) protection, and a complete coverage picture emerges. But what are we protecting ourselves from?

Understanding Attacks

A network can be attacked in many different ways, and each day we learn of new attacks. The kinds of attacks a system may encounter include the following:

- **Virus** Small program that piggybacks on real programs.
- **E-mail virus** Virus that moves by e-mail macros.
- **Worm** Small program that uses a network security hole to replicate.
- **Trojan horse** Program that claims one thing, and does another.
- **Spyware** Specific kind of Trojan horse that collects confidential information.
- **Phishing** E-mail and Web site that claims one thing, yet does another.

All of these different attack types are called threats. Threats are the bad things that are out there.

Threats take advantage of weaknesses or openings in our network, usually in our software. These openings are called vulnerabilities, and the portion of software that attacks the vulnerability is called an exploit.

A quick list of 10 notable exploits includes the following:

- Brain virus, 1986
- Morris worm, 1988
- Panix SYN flood, 1996
- Melissa e-mail virus, 1999
- ILOVEYOU e-mail virus, 2000
- Code Red worm, 2001
- Nimda e-mail virus and worm, 2001
- Slammer worm, 2003
- Blaster worm, 2003
- Mydoom worm, 2004

Let's take a closer look at a couple of these.

Brain Virus, 1986

The Brain virus was one of the first widespread IBM PC viruses (earlier viruses had hit mainframes). Similar to a biological virus, a computer "virus" (a term coined by Fred Cohen in 1983) needs to replicate and spread. The Brain virus was a boot sector virus, and infected IBM PC 360KB diskettes by doing the following:

- Its input vector exploited a trust vulnerability in the boot sector file input.
- It became resident in memory.
- It spread by automatically writing to other 360KB diskette boot sector files.

The cleanup response was to shut down and reboot with a clean 360KB boot diskette. Unfortunately, the same kind of piggybacking onto executable files lies behind many virus attacks today.

Morris Worm, 1988

The Morris worm was one of the first Internet worms. Unlike a virus, a worm exploits vulnerabilities in network programs in order to execute and spread. Richard Morris released the worm as an experiment to measure the size of the Internet, whereupon his creation infected DEC VAX and Sun 3 hosts running Unix, and exploited various security holes in sendmail (back door debug build), finger (buffer overflow), and rsh/rexec (password dictionary attack), doing the following:

- It became resident in memory.
- It spread automatically through local directory and network searches.

The cleanup response was to patch the sendmail program and harden the host by disabling or removing finger and rsh/rexec programs.

The same kind of buffer overflow vulnerability in network programs is behind many worm attacks today.

Panix SYN Flood, 1996

The Panix SYN flood was one of the first Internet Denial-of-Service (DOS) floods. Unlike a virus or worm, which executes code, a flood consumes resources. Servers for the New York ISP Panix were flooded, with the following results:

- The input vector exploited the resource management weakness in the TCP stack.
- It consumed buffer resources.
- It spread only by imitation and exploit sharing.

The cleanup response was to tune the operating system queues and timers, to implement IP address filtering to network ingress, and to add a stateless TCP cookie handshake to operating systems.

The same kind of resource management vulnerabilities in network and application programs are behind today's larger distributed Denial-of-Service (DDOS) floods.

Old Root Causes, New Attacks

I would like to say that we've learned from previous failures, and that these kinds of things will never happen again. But while we have analyzed the root causes of these vulnerabilities, know the kinds of software flaws that cause these vulnerabilities, and can detect and protect against the faults, new software continues to be written every day with the same fundamental flaws.

Unified Threat Management

Since the threats are not going away, they need to be managed. That can be difficult, however, because these threats occur at multiple layers of the network. Some attacks occur at the network operating system level, others at the application level, and still others in files and content. Some attacks such a viruses have fingerprints that can be positively identified, while others, such as TCP SYN flood, may be indistinguishable from the good traffic of a flash crowd.

Vulnerability Databases

A number of organizations collect and classify information about different kinds of computer threats. These groups are generally incident-response organizations for a geographic area. One of the oldest computer incident-response groups is the CERT, which operates at Carnegie Mellon University. CERT was formed in 1988 in response to the Morris worm. In

researching and responding to security incidents, CERT created and operates the Vulnerability Notes database (VU#). (For more information on the CERT Vulnerability Notes Database, see http://www.kb.cert.org/vuls/. Information from the U.S. National Vulnerability Database can be found at http://nvd.nist.gov/.)

Public vulnerability databases are like the better business bureau, where researchers, customers, and vendors can report vulnerabilities. (However, common practice is to not publicly report vulnerabilities until there is a fix, and vendors have given customers a chance to patch it.)

Bug Databases

In addition to public vulnerabilities, every vendor has a private internal engineering bug database. Not all flaws are resolved before a product is released, and some flaws are found only after release. Software of average quality typically contains one bug per 1000 lines of source code. That means a software product with 1,000,000 lines of code has 1000 bugs. Not all bugs are security vulnerabilities. But if 50 percent are, there are likely 500 vulnerabilities waiting to be fixed or exploited.

Common Name Dictionary

Since there are many different public and private vulnerability and bug databases, which can use different names and identities for the same vulnerability, a standard was needed for common naming. The Common Vulnerabilities and Exposures (CVE) naming standard is a public dictionary of software security flaws. The CVE currently contains about 20,000 names. Vendor security alerts and patches don't necessarily create CVE items.

All Juniper Networks attack objects provide CVE names for linking to public documentation. This is better than every security product providing its own different documentation. (For more information on CVE, see http://cve.mitre.org/.)

The Juniper Security Research Team

The security team is a central group of experienced security researchers and engineers that collect, analyze, build and test vulnerability signatures. Why is this important? The nature of security is preparation for the worst-case scenario. Before any specific attack is discovered in the wild, the probability of a specific attack occurring is small. Human nature being what it is, we don't want to hear bad news. But security experts prepare for the unlikely threats, using systematic methods. Left to individual operators, research and preparation tends to be a low priority and receives few resources, because it's usually a just-in-case afterthought. As a central resource, this preparation can be a high priority, can be staffed appropriately, leveraged by many, and kept actively engaged.

Network admins are pushed for efficiency, and are rarely measured by their (just-in-case) preparations.

This is why a central security team, signature subscription services, and local incident response makes good sense for security. (For more information on the Juniper Security Center, see http://security.juniper.net.)

Understanding the Anatomy of an Attack

There are almost as many ways to attack a network as there are hackers, but the majority of attack methods can be categorized as one of the following: manual attacks or automated attacks. Manual attacks are generally still performed by a piece of code or other script, but the attack itself is initiated at the request of a live user who selects his or her targets specifically. Automated attacks cover the kinds of attacks made by self-propagating worms and other viruses. There's also the question of the competence of an attacker or the complexity of an automated attack, which we'll discuss here as well.

The Three Phases of a Hack

Most hack attacks follow a series of phases:

1. **Reconnaissance** Initial probing for vulnerable services. Can include direct action against the target, such as port scanning, OS (operating system) fingerprinting, and banner capturing, or it can be performing research about the target.
2. **Exploit** An attempt to take control of a target by malicious means. This can include denying the service of the target to valid users. Generally, the ultimate goal is to achieve root, system, or administrator level access on the target.
3. **Consolidation** Ensuring that control of the target is kept. This usually means destroying logs, disabling firewalls and antivirus software, and sometimes includes process hiding and other means of obfuscating the attacker's presence on the system. In some extreme cases, the attacker may even patch the target against the exploit he used to attack the box, ensuring that no one else exploits the target after him.

While each step may have more or less emphasis, depending on the attacker, most hack attacks follow this pattern of progression.

Script Kiddies

For manual attacks, the majority of events are generated by inexperienced malicious hackers, known both in the industry and the hacking underground as "Script Kiddies." This derogatory reference implies both a lack of maturity ("just a kid") as well as a lack of technical prowess (they use scripts or other pre-written code instead of writing their own). Despite these limiting factors, what they lack in quality, they more than make up for in quantity.

Under a hail of arrows, even the mightiest warrior may fall. These sorts of attacks will generally be obvious, obnoxious, and sudden, and will usually light up your firewall or IDP (Intrusion and Detection Prevention) like a Christmas tree.

The majority of these attacks have no true intelligence behind them, despite being launched by a real person. Generally, the reconnaissance phase of these sorts of attacks will be a "recon-in-force" of a SYN packet and immediately transition to phase two by banging on your front door like an insistent vacuum cleaner salesman. Script Kiddies (also "Skr1pt Kiddies," "Newbies," or just "Newbs/Noobs") glean through security Web sites like Security Focus (www.securityfocus.com), Packet Storm Security (http://packetstormsecurity.nl), and other sites that provide proof of concept code for exploits for new scripts to try out. Once they have these scripts, they will blindly throw them against targets—very few of these amateurs understand exactly how these hacking tools work or how to change them to do something else. Many sites that provide code realize this, and will purposely break the script so that it doesn't work right, but the script will work correctly with a simple fix after a walk-through of the code by an experienced security professional.

Unfortunately, that only stops the new, inexperienced, or unaffiliated hacker. More commonly, hacking groups or gangs form with a few knowledgeable members at its core, with new inept recruits joining continuously. The people themselves need not live near each other in real life, but rather meet online in Internet Relay Chat (IRC) rooms and other instant messaging forums. These virtual groups will amass war chests of scripts, code snippets, and shellcode that work, thanks to the work of more experienced members. Often, different hacking groups will start hacking wars, where each side attempts to outdo the other in either quantity or perceived difficulty of targets hacked in a single time span. Military targets in particular are seen as more difficult, when in fact the security of these sites is often well below corporate standards. Mass Web site defacements are the most common result from these intergroup hacking wars, with immature, lewd, or insulting content posted to the sites.

A bright side to this problem is that many times a successful breach by these amateurs is not exploited to its fullest, since many of these hackers have no clue as to exactly what sort of system they have gained access to, or how to proceed from there. To them, *owning* (a successful hack which results in a root, administrator, or system-level account) a *box* (a server), and modifying its presented Web page for others to see and acknowledge is generally sufficient. These sorts of attacks commonly do not proceed to phase three, consolidation.

From a protection standpoint, to defend against these sorts of attacks, it is important to keep DI and IDP signatures updated, and all systems patched, whether directly exposed to the Internet or not. Defense-in-depth is also key to ensuring that a successful breach does not spread. The motivation behind these groups is quick publicity, so expect hard, fast, obvious, but thorough strikes across your entire Internet-facing systems.

Black Hat Hackers

Experienced malicious hackers (sometimes called "Black Hat" hackers or just "Black Hats") tend to be either a Script Kiddy graduating from the underground cyber-gangs, or a net-

work security professional or other administrator turning to the "dark side"—or a combination of both. In fact, it is common to call law-abiding security professionals "White Hats," with some morally challenged but generally good-intentioned people termed "Grey Hats." The clear delineation here is intent: Black Hats are in it for malicious reasons, often those of profit. This hat color scheme gets its roots from old Western movies and early black and white Western TV shows. In these shows, the bad guys always wore black hats, and the good guys wore white hats. Roles and morality were clearly defined. In the real world, this distinction is far more muddled.

Black Hats will slowly and patiently troll through networks, looking for vulnerabilities. Generally, they will have done their homework very thoroughly and will have a good idea of the network layout and systems present before ever sending a single packet directly against your network—their phase one preparation is meticulous. A surprising amount of data can be gleaned from simple tools like the WhoIs database and Google or other Web search engines for free. Mail lists and newsgroups when data-mined for domains from a target can reveal many important details about what systems and servers are used simply by monitoring network and system admins as they ask questions about how to solve server problems or configure devices for their networks. A wealth of information can be gleaned this way regarding social engineering as well. Names, titles, phone numbers, and addresses—it's all there for use by a skilled impersonator, allowing them to then make a few phone calls and obtain domain information, usernames, and sometimes even passwords!

Are You Owned?

Social Engineering

Social engineering is the term used to describe the process by which hackers obtain technical information without using a computer directly to do so. Social engineering is essentially conning someone to provide you with useful information that they should not—whether it's something obviously important like usernames and passwords or something seemingly innocuous like the name of a network administrator or his phone number.

With a few simple pieces of valid information, some good voice acting and proper forethought, a hacker could convince you over the phone that he or she was a new security engineer, and that the CEO is in a huff and needs the password changed now because he can't get to his e-mail or someone's going to get fired. "And that new password is what now? He needs to know it so we can log in and check it…"

Be sure to train your staff, including receptionists who answer public queries, to safeguard information so as to keep it out of the hands of hackers. A good idea is to employ authentication mechanisms to prevent impersonation.

The recon portion of the attack for a cautious Black Hat may last weeks or even months—painstakingly piecing together a coherent map of your network. When the decision to move to phase two and actively attack is finally made, the attack is quiet, slight, and subtle. They will avoid causing a crash of any services if they can help it, and will move slowly through the network, trying to avoid IDPs and other traffic logging devices. Phase three, Consolidation, is also very common, and typically includes patching the system from further vulnerability, so some Script Kiddy doesn't come in behind them and ruin their carefully laid plans.

A Black Hat's motivation is usually a strong desire to access your data—credit cards, bank accounts, Social Security numbers, usernames, and passwords. Other times, it may be for petty revenge for perceived wrongs. Or they may want to figure out a way to divert your traffic to Web sites they control so they can dupe users into providing these critical pieces of information to them—a technique known as *phishing* (pronounced like *fishing*, but with a twist). Some phishing attacks merely copy your Web site to their own, and entice people to the site with a list of e-mails they may have lifted off your mail or database server. Sometimes malware authors will also compromise Web sites in a manner similar to a Script Kiddy Web defacement, but instead of modifying the content on the site, they merely add additional files to it. This allows them to use the Web site itself as an infection vector for all who visit the site by adding a malicious JPEG file, Trojan horse binary, or other script into an otherwise innocuous Web site (even one protected by encryption such as Hypertext Transfer Protocol Secure, known simply as HTTPS).

Defense against these sorts of attacks requires good network security design as well as good security policy design and enforcement. Training employees, especially IT staff and receptionists or other public-facing employees, about social-engineering awareness and proper information control policy is paramount. For the network itself, proper isolation of critical databases and other stores of important data, combined with monitoring and logging systems that are unreachable from potentially compromised servers is key. Following up on suspicious activity is also important.

Worms, Viruses, and Other Automated Malware

Mentioned in the following "Notes from the Underground" sidebar, the concept of self-propagating programs is nothing new, but the practical application has only been around for the last 15 to 20 years. Given the Internet's origins stem from 40 years ago, this is significant. Indeed, it's only in the last two to three years that malware has taken a rather nasty turn for the worst, and there's a good reason behind it.

Early worms were merely proofs-of-concept, either a "See what I can do" or some sort of glimpse at a Cyber Pearl Harbor or Internet Armageddon, and rarely had any purposefully malicious payload. This didn't keep them from being major nuisances that cost companies millions of dollars year after year, however. But lately, some of the more advanced hacking

groups started getting the idea that a large group of computers under a single organization's complete control might be a fun thing to have. And the concept of a zombie army was born.

Are You Owned?

Are You a Zombie?

The majority of machines compromised to make a zombie army are those of unprotected home users directly connected to the Internet through DSL lines or cable modems. A recent study showed that while 60 percent of home Internet users surveyed felt they were safe from hackers, only 33 percent of them had some sort of firewall. Of that minority of Internet users with firewalls, 72 percent were found to be misconfigured. This means *less than 10 percent* of home Internet users are properly protected from attack!

Furthermore, of the users who had wireless access in their homes, 38 percent of them used no encryption, and the other 62 percent who did, used wireless encryption schemes with known security flaws that could be exploited to obtain the decryption key. Essentially, every person surveyed who used wireless could be a point from which a hacker could attack—and over a third of them effortlessly.

Find out more information from the study online at www.staysafeonline.info/news/safety_study_v04.pdf.

Zombies, sometimes referred to as *Bots* (a group of Bots is a *Bot-net*), are essentially Trojan horses left by a self-propagating worm. These nasty bits of code generally phone home to either an IRC channel or other listening post system and report their readiness to accept commands. Underground hacker groups will work hard to compromise as many machines as they can to build up the number of systems under their command. Bot-nets comprised of hundreds to tens of thousands of machines have been recorded. Usually, these groups use the bots to flood target servers with packets, causing a Denial-of-Service (DoS) attack from multiple points, and creating a Distributed Denial-of-Service (DDoS) attack. Nuking a person or site you didn't like is fun for these people. But today hackers are out for more than fun.

Once the reality of a multi-thousand node anonymous, controllable network was created, it was inevitable that economics would enter the picture, and so zombie armies were sold to the highest bidder—typically spammers and organized crime. Spammers use these bots to relay spam so ISPs (Internet service providers) can't track them back to the original spammer and shut down their connection. This has become so important to spammers that

eventually they began contracting ethically challenged programmers to write worms for them with specific features such as mail relay and competitor Trojan horse removal. Agobot, MyDoom, and SoBig are examples of these kinds of worms. Organized crime realizes the simplicity of a cyber-shakedown and extorts high-value transaction networks such as online gambling sites for protection from DDoS attack by bot-nets under the mob's control.

Protection from these tenacious binaries requires defense-in-depth (security checkpoints at multiple points within your network) as well as a comprehensive defense solution (flood control, access control, and application layer inspection). Many of the Script Kiddy defense methods will also work against most worms since the target identification logic in these worms is generally limited—phase one recon is usually just a SYN to a potentially vulnerable port. This is because there is only so much space for what the worm needs to do—scanning, connecting, protocol negotiation, overflow method, shellcode, and propagation method, not to mention the backdoor Trojan. Most worms pick targets completely at random and try a variety of attacks against it, whether it's a valid target for the attack or not. To solve the complexity problem, many Trojans are now split into two or more parts: a small, simple propagating worm with a file transfer stub; and a second stage full-featured Trojan horse with *phone home*, e-mail spamming, and so on. The first stage attacks and infects, then loads the second stage for the heavy lifting. This allows for an effective phase three consolidation.

Information obtained by Honeypot Networks (systems designed to detect attacks) shows that the average life expectancy of a freshly installed Windows system without patches connected directly to the Internet and without a firewall or other protection is approximately 20 minutes. On some broadband or dial-up connections it can take 30 minutes or longer to download the correct patches to prevent compromise by these automated attack programs. Using the Internet unprotected is a race you can't win.

Are You Owned?

Multivector Malware

Hacking (the term as used by the media for unauthorized access) is as old as computer science itself. Early on, it consisted mostly of innocent pranks, or was done for learning and exploring. And while concepts for self-replicating programs were bantered around as early as 1949, the first practical viruses did not appear until the early 1980s.

These early malicious software (or *malware*) applications generally required a user's interaction to spread—a mouse button clicked, a file open, a disk inserted. By the late 1980s, however, fully automated self-replicating software, generally known as worms, were finally realized. These programs would detect, attack, infect, and restart all over again on the new victim without any human

Continued

interaction. The earliest worms, such as the Morris Worm in 1988, had no purposeful malicious intent, but due to programming errors and other unconsidered circumstances, it still caused a lot of problems.

The earliest worms and hacking attacks targeted a single known vulnerability, generally on a single computing platform. Code Red is a classic example—it targeted only Microsoft Windows Web servers running Internet Information Server (IIS), and specifically a single flaw in the way IIS handled ISAPI (Internet Server Application Programming Interface) extensions. And while they did significant damage, a single flaw on a single machine tends to confine the attack to a defined area, with a known specific defense.

Unfortunately, this is no longer the case. Malware is now very complex, and the motivations for malware have changed with it. Early malware was limited to pranks like file deletion, Web defacement, CD tray openings, and so on. Later, when commerce came to the Web, and valuable data, like credit card numbers and other personal information were now online and potentially vulnerable, greed became a factor in why and how malware authors wrote their code. Recently, the culprits are spammers with significant financial clout, who pay programmers to add certain features to their malware so that spam (unsolicited e-mail), spim (unsolicited instant messages), and spyware can be spread for fun and profit.

NetSky, MyDoom, and Agobot are the newest breeds of these super-worms. New versions come out almost weekly, and certainly after any new major vulnerability announcement. They don't target just one vulnerability on one platform—they are multi-vector, self-propagating infectors, and they'll stop at nothing to infiltrate your network. Most exploit at least four different vulnerabilities, as well as brute force login algorithms. These worms even attack each other—NetSky and MyDoom both remove other Trojan horses as well as antivirus and other security programs. A variant of Agobot attempts to overflow the FTP (File Transfer Protocol) server left behind by a Sasser worm infection as an infection vector.

Configuring Screen Settings

The network protocols themselves became early targets to attacks, because every Internet host runs TCP/IP, regardless of operating system or application. Nothing is more common on the Internet than TCP/IP itself.

Listed next are five well-known examples of network protocol attacks.

- TCP SYN Flood Panix Vulnerability, CA-96.21.tcp_syn.flooding, CVE-1999-0116
- ICMP Fragment Ping of Death Vulnerability, CA-96.26.ping, CVE-1999-0128
- TCP OOB Winnuke Vulnerability, CVE-1999-0153

- TCP SYN LAND Vulnerability, CVE-1999-0016, CVE-2005-0688
- IP Source Route Option Vulnerability, CVE-2006-2379

Because TCP/IP is necessary, the common hardening strategy of disabling or removing the weak or vulnerable software is generally not an option. A few obscure feature options are almost never used, and can usually be disabled or removed. But in general, we must run TCP/IP, so we must identify and patch specific vulnerabilities, and tune the TCP/IP stack resources for availability.

The Screen options on a NetScreen firewall are one of the earliest forms of an Intrusion Protection System (IPS) found on these firewalls. The Screens address many of the attacks that occur at the network and transport layers of the TCP/IP protocol stack. New protection features have been added over time to address new threats present on the Internet.

It is best to start with a common policy everywhere. These low-level policies are designed to be enforced fairly, meaning "equitably" for everyone. However, these protections are security-zone specific—each zone may have unique settings applied to it so that user zones can have different settings than server zones, and special zones like management zones can have special settings. For all options, these settings are applied as the inspected traffic externally enters the zone—that is, when the stream is read from the interface off the wire, not as it passes through the NetScreen and out another interface.

These options are enforced before policy decisions because they need to be handled efficiently during a Denial-of-Service flood attack. Attacks that occur at the network and transport layers of the TCP/IP stack need to be enforced before upper-layer policies because these network protections perform the preprocessing that closes potential evasion and Denial-of-Service flood weaknesses, and thus result in higher accuracy.

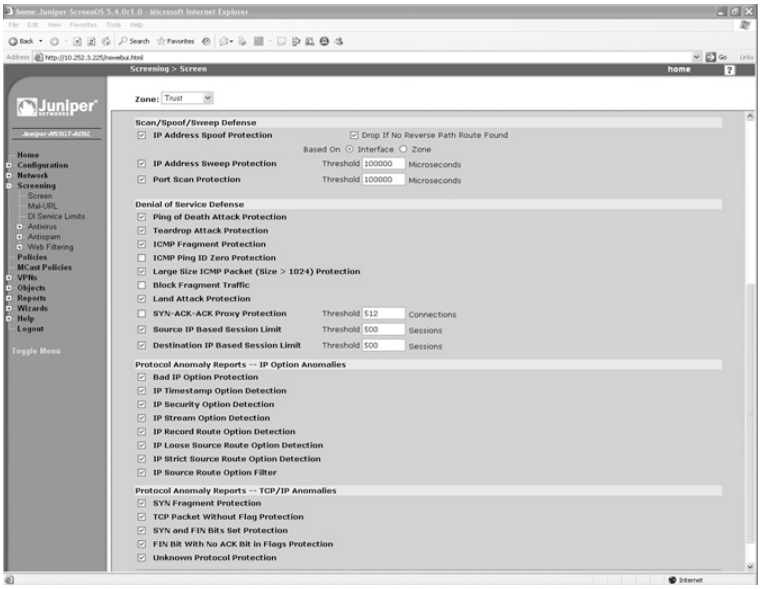
While NetScreen organizes these attacks by layers and protocols, it's easier to talk about them more generically according to their purpose. The three major functions of the SCREEN features are *reconnaissance detection*, Denial-of-Service flood protection, and protocol attack protection

See Figure 10.1 for the ScreenOS version 5.4.0r1 SCREEN setting page.

TCP/IP Behavior Anomaly Detection

In order to provide Denial-of-Service flood protection and reconnaissance detection, the NetScreen tracks and correlates behavior, with user-definable thresholds between the typical and the unusual. The NetScreen tracks connectivity with sessions. It can also correlate behavior across multiple sessions, noting how well connected a host is. A typical user connects in a fan-out pattern, while a typical server connects in a fan-in pattern. By studying the behavior of direction, number, and rate of connections, we can deduce normal client activity, a reconnaissance scan, a worm outbreak, normal server activity, a flash crowd, or a Denial-of-Service flood.

Figure 10.1 SCREEN Settings for ScreenOS 5.4.0r1



While reconnaissance and Denial-of-Service floods may have very different motives, they are often detected by the same or very similar correlations. Reconnaissance detection correlates the fan-out from a single source. Denial-of-Service flood protection correlates the fan-in to a single destination.

Unless done carefully, behavior anomaly detection can be noisy and prone to false alarms. A good strategy is to enable thresholds at least one order of magnitude higher than required (enabling visibility to the correlations), and tune down only when you really understand your normal behavior. Especially when first getting started, you want to detect only the most unusual anomalies.

Reconnaissance Detection

As mentioned earlier, an attacker will more than likely perform some initial reconnaissance on your systems before launching an attack. Generally, these methods are benign, and so are easily lost in the clutter of normal traffic. If you know what to look for, however, they tend to stand out, by counting the number or rate of connections. The same kinds of correlations that detect reconnaissance can also detect floods. Often, automated reconnaissance is part of a fast spreading worm.

Port scanning, especially across multiple machines, is the simplest and most common network reconnaissance method. A variety of tools, most notably NMap (www.insecure.org/nmap/), perform port scanning as well as more advanced system identification such as OS fingerprinting and service banner capture.

An IP source session limit can detect scans for which no specific scan detection is available, including TCP and UDP port sweeps. When sweeping many destinations across your networks, scanners will hit the IP source session limit, especially when a high percentage of scans are unresponsive, leaving failed scan sessions open.

Denial-of-Service Flood Protection

Flooding is one of the oldest, yet still very popular, methods of attack. Essentially, a flooding attack overwhelms the victim's machine with packets faster than the remote machine can process them. Although CPU and memory capacity have improved dramatically since the early days of networking, networks and end systems can still be vulnerable. It's still prudent to block these attacks as far out to the perimeter as possible, if for no other reason than to clean up the clutter and keep unnecessary traffic out of your network.

IP Session Limiting

The most general detection identifies activity per IP source and IP destination. This helps the administrator answer the following questions:

- How many active IP sources do my networks have?
- How many active IP destinations do my networks communicate with?
- How many active IP sources communicate with my networks?
- How many active IP destinations do my networks have?

By correlating session activity per IP source and IP destination, administrators can measure how connected each host actually is. Reconnaissance and worm attacks are both very social. Where traffic logs contain the same information as history, IP session limiting operates in real time.

- **Source IP–Based Session Limit Threshold** This controls how many connections per single IP source are permitted before the NetScreen begins denying new sessions. Any new sessions above this threshold are denied for the remainder of the second until the source closes some sessions.
- **Destination IP Session Limit Threshold** This controls how many connections per single IP destination are permitted before the NetScreen begins denying new sessions. Any new sessions above this threshold are denied for the remainder of the second until the destination closes some sessions.

Good starting numbers for IP source and destination limits are 1000 and 10,000 sessions. Typical clients peak at tens or hundreds of sessions, with off-peak connectivity in the single digits. Typical servers peak at hundreds or thousands of sessions, with off-peak connectivity in tens of sessions.

ICMP Network Scan

Juniper firewalls can even detect an ICMP scan across multiple systems (PING sweep). Attackers will sweep the destination network address space to discover the hosts there. This will usually be followed up with host scans to discover the open ports on each host.

- **IP Address Sweep Protection** This controls how many ICMP packets per second per single IP source are permitted before the firewall begins dropping ICMP packets from that source. Packets are dropped for the remainder of the second. The configuration actually detects a quick series of 10 packet probes in a user-definable period of microseconds. To increase the detection rate, lower the period. To decrease the detection rate, increase the period. The lowest possible detection rate is 10 packets in a period of 1,000,000 microseconds (one second). No sessions are set up for dropped packets.

A good starting number for IP address sweep protection is 100,000 microseconds. (Ten packets in 100,000 microseconds is 100 packets per second.) Network management packages, which perform network discovery and health checks, can be located in a separate security zone with higher thresholds.

ICMP Rate Limiting

ICMP flood is a common flood attack and is often combined with other kinds of flood attacks.

ICMP flood is the most straightforward of the flood protections. A threshold value of total ICMP packets per second to a destination (from all IP addresses) is set, and if that threshold (a default of 1000 p/s) is exceeded in a particular second, the remainder of the ICMP packets for that second, as well as all of the ICMP packets for the next second, are dropped. Furthermore, sessions are not made for dropped packets.

- **ICMP Flood Protection** This controls how many ICMP packets per second per single IP destination are permitted before the firewall begins dropping ICMP packets to that destination. Packets are dropped for the remainder of the second, and all of the next second. No sessions are set up for dropped packets.

A good starting number for ICMP flood protection is 100 packets per second. If you find your routers returning high numbers of ICMP messages, resulting from normal PING health checks, and normal network discovery (ICMP host unreachable, ICMP path MTU discovery), you can adjust your trust security zone with a higher threshold.

TCP SYN Host Scan

Port scanning, especially across multiple machines, is the simplest and most common network reconnaissance method. A variety of tools, most notably NMap (www.insecure.org/

nmap/), perform port scanning as well as more advanced system identification such as OS fingerprinting and service banner capture.

- **Port Scan Protection** This controls how many TCP SYN packets per second per single IP source are permitted before the firewall begins dropping TCP SYN packets from that source. Packets are dropped for the remainder of the second. The configuration actually detects a quick series of 10 packet probes in a user-definable period of microseconds. To increase the detection rate, lower the period. To decrease the detection rate, increase the period. The lowest possible detection rate is 10 packets in a period of 1,000,000 microseconds (one second). No sessions are set up for dropped packets.

A good starting number for Port Scan Protection is 100,000 microseconds. (Ten packets in 100,000 microseconds is 100 packets per second.) Some protocols can open up several ports in rapid succession. If you find this triggering often from trusted machines that you've verified have no malware running on them, you may need to adjust this threshold higher to weed out these false positives.

You'll most commonly detect scans and sweeps from Script Kiddies or other automated, semi-intelligent attacks. More experienced Black Hats will scan more slowly, generally slow enough to avoid being detected by a firewall. This technique of sending port scanning packets infrequently over a long period of time is known as a *slow scan*.

TCP SYN Rate Limiting

TCP SYN flood is a very common flood attack and is often combined with other kinds of flood attacks. The original TCP SYN flood attack depended on flooding a shared resource TCP queue that was only 6 to 12 items. The obvious answer was to make the queue deeper. Using a larger TCP queue, the TCP uses a three-way handshake to validate the source before signaling the application. Even this larger queue is still finite, however. Frequently, floods may be distributed, and may have fake sources. TCP SYN cookies use a stateless handshake to validate the source, before allocating queue resources. Because TCP SYN cookies are stateless, there is no queue, but even so, the TCP SYN cookie transmit rate is still finite. Today, most TCP implementations are already patched to have a larger queue. In addition, TCP SYN cookie can filter fake sources. But what happens when real sources flood the larger TCP queue? The obvious answer is to rate limit and prioritize the queue resource.

TCP SYN is the most complicated flood protection, due to the NetScreen's ability to both rate limit to provide flood protection, and to authenticate the source to provide spoof protection. Rate limiting works by dropping packets, while spoof protection works by handshaking with the source, either through the stateful TCP SYN proxy handshake, or the stateless TCP SYN cookie proxy handshake.

- **Source Threshold** This controls how many TCP SYN packets per second per single IP source are permitted before the NetScreen begins dropping SYN packets

from that source. Packets are dropped for the remainder of the second. No sessions are set up for dropped packets.

- **Destination Threshold** Similar to the source threshold, except the number of packets are compared for a particular destination IP. This controls how many TCP SYN packets per second per single IP destination are permitted before the NetScreen begins dropping SYN packets to that destination. Packets are dropped for the remainder of the second. No sessions are set up for dropped packets.

A good starting number for the source threshold is 100 packets per second. The source threshold is very useful for detecting TCP port scans from worm infections on end-user systems. Set this number relatively low on your user security zones (see the "Zone Isolation" section later in the chapter) and notice that when an infected host tries to open up 100 new connections per second to other targets, attempting to infect them, this feature will throttle that attack to a manageable level.

A good starting number for the destination threshold is 1000 packets per second. This can be used for servers or other important machines to keep the overall level of new TCP connections to a set maximum. This may need to be even higher for aggregation devices like server load balancers.

TCP SYN floods often spoof the source IP because the source has no intention of completing the TCP three-way handshake, and because this makes the attack more difficult to filter with a blacklist. NetScreen can authenticate the source by completing the TCP three-way handshake. If the source does not respond, it was spoofed and there is no need to do anything else. You can configure whether the handshake should be the stateful TCP SYN proxy handshake (which you should use if clients use TCP options; it performs less well under flood), or the stateless TCP SYN cookie proxy handshake (which performs well under flood, but does not preserve client TCP options). (Because the TCP cookie is stateless, the NetScreen does not save any TCP options from the initial TCP SYN.) Since the loss of TCP options happens only when under flood anyway, it's generally accepted to be a good trade, so TCP SYN cookie is preferred. Here is an example of the TCP SYN cookie.

```
home-> set flow syn-proxy syn-cookie
```

- **Attack Threshold** This controls how many packets per second must arrive at a single IP/port pair before the NetScreen begins proxying SYNs. Any SYNs above this threshold for the remainder of that second are proxied, using either the stateless TCP SYN cookie or the stateful TCP SYN proxy queue, until the proxy queue is full.
- **Alarm Threshold** This controls when an alarm should be logged for a potential SYN flood. This number should be *lower* than your attack threshold—it is a warning that you could be having trouble.
- **Timeout** This is how long a SYN should be kept in the proxy queue before being flushed as an invalid connection request. Its default setting is 20 seconds,

which is very generous. I would recommend something lower, perhaps as low as two to three seconds, depending on the latency of your network. Keep in mind that any properly negotiated three-way handshake will automatically clear the entry from the queue, and any non-spoofed TCP will always retry. This setting has no effect on TCP SYN cookies.

- **Queue Size** Specifies the number of SYNs that can be proxied and monitored before dropping new SYNs. A larger number uses more memory (since it needs to remember the IP address and port number of the session requested), and also takes longer to scan the queue for completed three-way handshakes, resulting in a higher initial connection latency. This setting has no effect for TCP SYN cookie.

A good number for the attack threshold is 100 packets per second. Above this, a possible TCP SYN flood may be occurring, and so NetScreen begins authenticating TCP sources. A good number for the alarm threshold is also 100 packets per second. This alerts you that anti-spoofing has kicked in. (When the TCP SYN reaches the destination threshold—recommended for 1000 packets per second and above—additional packets will just be dropped, and another alarm dispatched.)

There is a special case when NetScreen is in *transparent mode* and it needs to proxy the SYN for a session, but the destination MAC (Media Access Control) address hasn't been learned yet and isn't in NetScreen's ARP (Address Resolution Protocol) table. This could occur on a large layer 2 network where the destination MAC has aged off of the NetScreen device, or it could be that the destination IP doesn't exist, and therefore the MAC cannot be learned. The Drop Unknown MAC option allows you to set the behavior of the NetScreen device when this situation occurs. By default, NetScreen will pass a packet with an unknown destination MAC and *not* proxy it. With this option set, NetScreen will drop the packet instead.

UDP Data Rate Limiting

UDP flood is a common flood attack, and is often combined with other kinds of flood attacks. A UDP flood will often use large packets to consume bandwidth. Unlike other flood protections, which correlate the packets of new connections, UDP flood correlates all packets, including all data packets.

UDP flood protection is essentially the same as ICMP flood protection, but uses a separate threshold and queue. It employs a threshold value (default of 1000 p/s) that, if exceeded, drops all remaining UDP packets from all IP addresses for that second as well as the next.

UDP flood protection controls how many UDP packets per second to a single IP destination are permitted before NetScreen begins dropping UDP packets to that destination. Packets are dropped for the remainder of the second, as well as all of the next second. No sessions are set up for dropped packets.

A good starting number for UDP flood protection is 5000 packets per second. (That's 60 Mbps for 1500B packets.) If you find normal servers transferring large amounts of data (such as TFTP file transfer or UDP encapsulated VPN tunnels) are tripping your threshold, you can adjust your trust security zones by using an even higher threshold.

TCP/IP Protocol Anomaly Detection

Protocol anomaly detection works by understanding the network protocols (which generally requires having a protocol engine for each network protocol) and by checking or validating the inputs for known abuses. In fact, much of this is really just part of the basic operation of the firewall, creating sessions, matching packets to flows and sessions, and closing sessions.

One might be tempted to reason that all of the vulnerabilities in TCP/IP have been found and patched. But as recently as June 2006, there was a new vulnerability disclosed using malformed IP source route option packets to execute remote code. Here is an example of this code.

```
TCP Protocol Validation: TCP SYN check, TCP SEQ check (evasion, man-in-the-
middle,)
```

IP Option Validation

Some methods of network mapping involve detecting Internet Protocol (IP)–layer parameters. ScreenOS supports blocking these probes with a slew of IP option anomalies. IP options are rarely, if ever, used, so they are not necessarily as well exercised as the common IP code. If one disables or blocks IP options, the potential for harm largely goes away. Hardening against these IP packets removes a couple more abuses that no longer need to be considered. This simplifies overall analysis.

The disable or block IP options are as follows:

- Bad IP Option Protection
- IP Timestamp Option Detection
- IP Security Option Detection
- IP Stream Option Detection
- IP Record Route Option Detection
- IP Loose Source Route Option Detection
- IP Strict Source Route Option Detection
- IP Source Route Option Filter (Note CVE-2006-2379, MS06-032)

If for some reason you have need for these services, generally you'll already know about it. If these do not sound familiar to you, it's a safe bet you don't need them. Most of these activities have no valid use on a network and are generally safe to block.

IP Fragmentation Validation, Attack Signatures

Fragmentation is a standard IP option that is occasionally, but not heavily, used. Typical fragmentation is less than one percent of all packets. The TCP transport handles its own fragmenta-

tion, and negotiates a maximum segment size to avoid fragmentation. Otherwise, applications must handle their own fragmentation, and many do. But some applications like NSF do depend heavily on fragmentation. And common Ping troubleshooting techniques use ICMP fragmentation.

Fragmentation can be used in possible attacks, abuses, and floods against code and resources handling fragmentation and reassembly. Since the exploit payload is often split up across multiple packets, if the packets are not reassembled, they may be able to evade intrusion detection techniques. The possibilities for IP fragment floods are real and numerous, especially because reassembly happens before any authentication of upper-layer protocols. Fragmentation floods are commonly part of Denial-of-Service attacks. IP fragmentation may also be used to create ambiguity and evade detection.

- **Block Fragment Traffic** Drops all packets with a more fragments flag or a fragment offset. No sessions are set up for dropped packets.
- **Teardrop Attack Protection, CVE-1999-0015, CA-97.28.Teardrop_Land** Drops all fragments with an overlapping fragment offset. This can result in a denial of service. Variations of this fragmentation attack include Teardrop-2, CVE-1999-0104, Nestea, CVE-1999-0257, and Bonk, CVE-1999-0258. The multiple exploits demonstrate the value in detecting and protecting against the underlying vulnerability, not any single exploit packet signature.
- **ICMP Fragment Protection** Drops all ICMP packets with a more fragments flag or a fragment offset. No sessions are set up for dropped packets.

A good starting point is to protect against Teardrop attacks, but permit all IP fragment traffic. You may want to analyze whether services that depend on fragmentation are vulnerable to Denial-of-Service floods, and whether fragmentation can be avoided by some other means such as TCP MSS.

ICMP Length Validation, Attack Signatures

All of the standard control and error messages ICMP is used for are short. Control messages are less than 512 bytes to avoid fragmentation, and error messages contain only the original IP header plus eight bytes of data, well under 512 bytes. Except for common Ping troubleshooting techniques using ICMP fragmentation, there is no reason for control messages to be fragmented.

- **Large Size ICMP Packet Protection** Drops all ICMP packets with a length greater than 1024 bytes. No sessions are set up for dropped packets.
- **Ping of Death Attack, CVE-1999-0128, CA-96.26.ping** Drops all ICMP packets with header and fragment offset, resulting in greater than 65535 bytes. This can result in a denial of service.

A good starting point is to protect against the Ping of Death attack, but permit large ICMP packets for network troubleshooting. You may want to analyze whether services that depend on Ping troubleshooting are required, and whether health checks and troubleshooting can use some other method. It is generally not required to troubleshoot fragmentation greater than the 8KB used by NFS.

TCP Flag Validation

Other methods of scanning involve modifying TCP (Transmission Control Protocol) flags to invalid or improper settings. Many stateless routers that are pressed into service as rudimentary firewalls can detect established communications based upon the TCP ACK flag. Scanners will utilize this logic flaw and send *ACK scans* in which the packet sent will have the ACK bit set—this bypasses most ACL (access control list)–based packet filters, but thanks to the stateful inspection feature in ScreenOS, no TCP packet not matching an established session (created with a proper TCP three-way handshake) may pass.

Hardening against these TCP packets removes a few more abuses that no longer need to be considered. This simplifies overall analysis.

- **SYN Fragment Protection** Detects and protects against initial TCP SYN packets with the IP fragment flag also set. TCP SYN signals have no data, and therefore cannot be fragmented. Any TCP SYN that is also an IP fragment is illegal. This hardens TCP against possible attacks, abuses, and floods against code and resources handling this illegal corner case. In this abuse case, IP fragmentation might be used to create ambiguity and evade detection.
- **TCP Packet without Flag Protection** Detects and protects against TCP packets without any flag set. TCP flags must always have either SYN or FIN signals and/or ACK flags. Any TCP segment without flags is illegal. This hardens TCP against the possible attacks, abuses, and floods against code and resources handling this illegal corner case. In this abuse case, variability in code handling this corner case may be used to fingerprint an endpoint. This is also called a TCP NULL scan.
- **SYN and FIN Bits Set Protection** Detects and protects against TCP packets with both SYN and FIN flags. TCP SYN signals open a connection, while FIN signals (and sometimes RST signals) close them. Any TCP segment with both SYN and FIN signals is illegal. This hardens TCP against possible attacks, abuses, and floods against code and resources handling this illegal corner case. Variability in code handling this corner case may be used to fingerprint an endpoint. This is sometimes called an open-close scan.
- **FIN Bit with No ACK Bit in Flags Protection** Detects and protects against TCP packets with FIN and without ACK. When closing a connection with the FIN signal, you must also acknowledge (ACK) the data received. This hardens TCP

against possible attacks, abuses, and floods against code and resources handling this illegal corner case.

There is no reason not to enable all of these.

TCP Attack Signatures

Here are examples of two TCP attack signatures.

- **LAND Attack Protection, CVE-1999-0016, CVE-2005-0688, CA-97.28.Teardrop_Land** Detects and protects against TCP packets with the SYN signal, the same source and destination IP, and the same source and destination port. This attack also relies on IP-spoofing, and any IP-spoof detection will also be detected.
- **WinNuke Attack Protection, CVE-1999-0153** Detects and protects against TCP packets to port 139 with the OOB flag and urgent pointer. This can result in a denial of service.

There is no reason not to enable both of these.

L7 Protocol Attacks

In addition to flood attacks, the SCREEN functions can also block protocol-specific attacks. These are generally legacy attacks—new attacks are blocked with Deep Inspection, discussed next. Protocols and attacks covered in clued the following:

- **HTTP (Hypertext Transfer Protocol)** Allows the blocking of Java and ActiveX code, as well as ZIP and EXE file downloads.
- **Windows** Allows the blocking of the classic WinNuke (malformed data to port 139) attack.
- **ICMP protocol attacks** Allows the blocking of the *Ping of Death* (fragment boundary overflow attack), ICMP fragments, and large ICMP packets.
- **TCP protocol attacks** Allows the blocking of Teardrop (another fragment boundary overflow attack) and Land (source, destination IP, and port are the same) attacks.

Applying Deep Inspection

Juniper Networks' line of NetScreen firewall products has evolved with security requirements to consistently keep up-to-date with threats that plague network administrators. Deep Inspection takes network security all the way up the stack to the application layer, inspecting traffic as it would be interpreted by the end-host application. This answers the problem

vexing many Administrators who are used to solving security at layer 3 and 4: "How do I defend from attacks when I need to leave port 80 open?"

Deep Inspection is a subset of Juniper Networks' award-winning NetScreen Intrusion Detection and Prevention, with a set of highly accurate attack signatures and support for a complete set of protocols typically considered to be Internet-facing—HTTP, SMTP (Simple Mail Transfer Protocol), DNS (Domain Name System), POP3 (Post Office Protocol v3), FTP, and IMAP (Internet Message Access Protocol), as well as support for file and print services—MS-RPC (Microsoft Remote Procedure Call), NBNAME (NetBIOS Name Service), SMB (Server Message Block), NFS (Network File System), and LPR (Line Printer)—support for Instant Messaging—AIM (AOL Instant Messenger), MSN (Microsoft Messenger), and YMSG (Yahoo Messenger)—and support for network management protocols: NTP (Network Time Protocol), TFTP (Trivial File Transfer Protocol), RADIUS (Remote Authentication Dial In User Service), LDAP (Lightweight Directory Access Protocol), SYSLOG (System Logging), and others. The protocol set has grown since its introduction in ScreenOS 5.0 from just common Internet protocols, to file and printer protocols in ScreenOS 5.1, to over 25 protocols in ScreenOS 5.4. What makes Deep Inspection different is a set of attack signatures selected to minimize false positives and to fit in the reduced memory of entry and midrange systems.

DI examines all incoming packets and assigns them a session (or in the case of stateless protocols such as UDP or ICMP, a *pseudo-session*). It reassembles fragments, rearranges out-of-order frames, and creates data streams from these packets (errors from overlapping fragments and other tomfoolery are handled by IP layer protocol anomaly inspectors). These streams are then handed off to protocol-specific inspection engines, called *Q modules*, which further inspect and parse the stream into protocol-specific elements (called *contexts*) for signature matching. Protocol-specific anomalies are also detected at this stage. For example, DNS requests are matched to DNS replies to ensure the answer matches the question—this prevents DNS poisoning.

These contexts are what make DI so accurate. With this level of parsing, a signature writer can specify a more targeted portion of the data stream for inspection—this also has the added benefit of increased performance, since only the relevant portion of the stream is inspected for attacks. Take the following hypothetical situation:

Say you have a simple, stateless, inline Intrusion Detection System (IDS) monitoring your network. A new vulnerability (in this case, a secret backdoor left by the developer) is discovered in the mail server, whereby if an e-mail arrives from a specific user (littlepig@bigbadwolf.com, for this example), in addition to forwarding the message to the recipient, it also takes whatever attachments are included with the message and attempts to execute them as programs. Being limited to this stateless IDS, you write a signature that says, "If you see the pattern 'littlepig@bigbadwolf.com' go over TCP port 25, block it." You then send an e-mail out to all your users, informing them to report any suspicious e-mails they receive from that address. Later that day, you ask a co-worker if he'd received any of the e-mails you were talking about earlier. He gets very confused and asks you what you're talking about. When you start talking about little pigs and big bad wolves, he gets a funny look on his face and

mumbles something about being late to a meeting and hurries off. It's only after you check your IDS logs that you realize that *no one* received your e-mail because your own IDS blocked it! It detected the string match based upon the data in your e-mail, and took what it thought was appropriate action.

Take that sample situation and instead use DI's SMTP-From context, putting your string match there. The same e-mail message you sent to your users would pass through DI unmolested because it knows the difference between the SMTP command phase of the session and the SMTP data phase of the session. Your email had a matching string in the data portion of the stream, which isn't where the vulnerability lies, so DI ignores it. Later that day, when a hacker tries to test your security, DI detects the match in the SMTP command phase (specifically, in an SMTP From command) and blocks the message from arriving on your mail server.

But what if you missed the memo? Security issues come up every day, and it's more than a full-time job just to keep abreast of all the details. Is this particular security announcement relevant to your network? Do you run a vulnerable version on any of your servers? Are these servers accessible from the Internet? How does the attack work? What kind of regular expression (RegEx) would detect it? Would your signature trigger on non-malicious traffic (called a *false positive)* and block legitimate traffic? Would your signature fail to trigger on malicious traffic (called a *false negative)* and let attacks through? Did you leave the garage door open this morning?

Since not everyone can be a full-time security researcher, Juniper Networks has the Juniper Engineering Security Team do research for you. With a valid subscription, you can receive a well-stocked signature pack as well as regular and periodic updates as new vulnerabilities are announced. Medium through critical (as defined by CERT/CC—www.cert.org/) severity issues, when possible, are covered by DI.

Deep Inspection Concepts

If you must run a network application, the ability to patch vulnerabilities on the wire, and validate the application protocol, is what Deep Inspection is all about. Using updated attack signatures and application engines, Deep Inspection provides a virtual patch for network applications. Deep Inspection can protect both users when inspecting server-to-client (STC) flows, and protect servers when inspecting client-to-server (CTS) flows.

Deep Inspection provides engines for the following application protocols:

- AIM (AOL Instant Messenger)
- DHCP (Dynamic Host Configuration Protocol)
- DNS (Domain Name System)
- FTP (File Transfer Protocol)
- GNUTELLA (File Sharing Network Protocol)

- GOPHER (Gopher Protocol)
- HTTP (Hypertext Transfer Protocol)
- ICMP (Internet Control Message Protocol)
- IDENT (Identification Protocol)
- IKE (Internet Key Exchange)
- IMAP (Internet Message Access Protocol)
- IRC (Internet Relay Chat Protocol)
- LDAP (Lightweight Directory Access Protocol)
- LPR (Line Printer)
- MSN (Microsoft Messenger)
- MSRPC (Microsoft Remote Procedure Call)
- NBNAME (NetBIOS Name Service)
- NFS (Network File Service)
- NTP (Network Time Protocol)
- POP3 (Post Office Protocol)
- RADIUS (Remote Authentication Dial In User Service)
- SMB (Server Message Block)
- SMTP (Simple Mail Transfer Protocol)
- SYSLOG (System Log)
- TELNET (Terminal Emulation Protocol)
- TFTP (Trivial File Transfer Protocol)
- VNC (Virtual Network Computing, or Remote Frame Buffer Protocol)
- WHOIS (Remote Directory Access Protocol)
- YMSG (Yahoo Messenger)

These engines decode the application protocol, and validate the inputs against potential vulnerabilities by checking many things like maximum lengths (for buffer overruns) and counting login failures (brute force attacks). Checking for the potential vulnerability, not the specific exploit signature, can detect attacks that don't have specific fingerprint signatures, so-called zero day attacks. Even if there is no engine for an application protocol, attack signatures can still be developed using stream or packet contexts.

Deep Inspection won't bury you with logs of false positives requiring complex technical investigation, and it includes only highly accurate signatures. On the entry-level model

(NS-5GT), with the least memory, Deep Inspection signatures include only attacks of critical severity.

Deep Inspection Planning

ScreenOS supports Deep Inspection on all platforms. Deep Inspection is generally better suited for the entry-level and midrange models, and it's not recommended for high-end platforms. For the most complete signature and protocol coverage, a stand-alone IDP can always be used, either in inline mode, sniffer mode, or using policy-based routing to redirect specific traffic. ScreenOS also supports special integrated hardware IDP modules on the ISG platforms.

Because a firewall may be deployed primarily to protect users (inspecting primarily server-to-client (STC) flows) or protect servers (inspecting client-to-server (CTS) flows), Deep Inspection offers specialized signature packages. The specialized signature packages produce fewer false positives, and use less memory and processing than including all signatures.

The following Deep Inspection signature packages are available.

- **Base** General set of worm signatures (both STC and CTS flows)
- **Client** Focuses on preventing users from getting malware while surfing (primarily STC flows)
- **Server** Focuses on preventing servers from attacks (primarily CTS flows)
- **Worm Mitigation** Focus on worm signatures (both STC and CTS flows)

Table 10.1 shows which versions of Juniper Networks NetScreen and SSG firewalls offer deep inspection.

www.syngress.com

Table 10.1 Deep Inspection Support by Firewall Model*

	NS-5GT	NS-25/50	NS-204/208	NS-500	NS-5200/5400	SSG-5	SSG-100	SSG-500	ISG-1000	ISG-2000
Deep Inspection	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Integrated IDP Module	No	No	No	No	No	No	No	No	Yes	Yes

* Data is based on Juniper Networks firewalls running ScreenOS 5.4.

Deep Inspection must have a device subscription service, and requires the setup of a name service and the correct time to update the signatures. Activation also requires taking a Deep Inspection action in the policy. You should plan ahead and purchase the Deep Inspection subscription, which you associate online with your registered device. The online service reserves a Deep Inspection subscription, and your device only needs to connect to, and authenticate, the service to retrieve its Deep Inspection subscription key.

Policy is the second level of specialization available to administrators. Specific rule settings can include attacks only against the applications known to be running on that host.

Getting the Database

NetScreen firewall products need a valid DI license key before DI is used. Your Juniper customer service representative can assist you with obtaining one, as well as helping out if there are problems with loading the key onto the device. Once the license is on the device, you're ready to load your database.

NOTE

All license keys are tied to the serial number of the device for which the key was granted—there is no such thing as a *universal* key. Trying to load a license key created for one unit and trying to load it on another unit—even the same model—will fail. Additionally, if you ever need to return your unit for replacement under the RMA policy, the new device you receive must have new keys issued for it. Support generally will handle this for you automatically, but you should still check it in case something doesn't work with the new unit. Many configuration settings are hidden until a valid license key is loaded to activate those features—loading a configuration from an entitled firewall onto another firewall without entitlements could cause the inactive portions of the config to be dropped.

The database file is a precompiled binary database file and can be downloaded by the device directly from the Juniper Web site if the firewall is attached to the Internet and firewall policies are in place to permit it to do so. This can be configured to automatically occur on a set schedule so you never miss an update. You can also force an update from the Internet via the WebUI.

TIP

With ScreenOS 5.1 and later, you can also use the Retrieve Subscriptions Now button in the Configuration | Update | ScreenOS/Keys WebUI page for the device to automatically retrieve keys assigned to it from Juniper Networks' enti-

tlement server. Note that this requires the firewall to have Web access to the Internet in order to connect to the server. For devices that cannot reach the entitlement server directly, the key file must be loaded on the device by hand.

If the firewall is on a private network, or if access is restricted, the file can be manually downloaded from the Internet, and then either placed on an internal Web server for your firewalls to download from (the URL that specifies the location of the attack file is mostly configurable) or it can be loaded by hand by either using the WebUI via HTTP upload direct from the browser or the command-line interface (CLI) via TFTP (Thin File Transfer Protocol) file transfer. For the latter method, a TFTP server is required.

Configuring the Firewall for Automatic DI Updates

One of the more handy features of DI is its ability to automatically check for new signature packs and download them as necessary without user intervention. Configuration for this is easy.

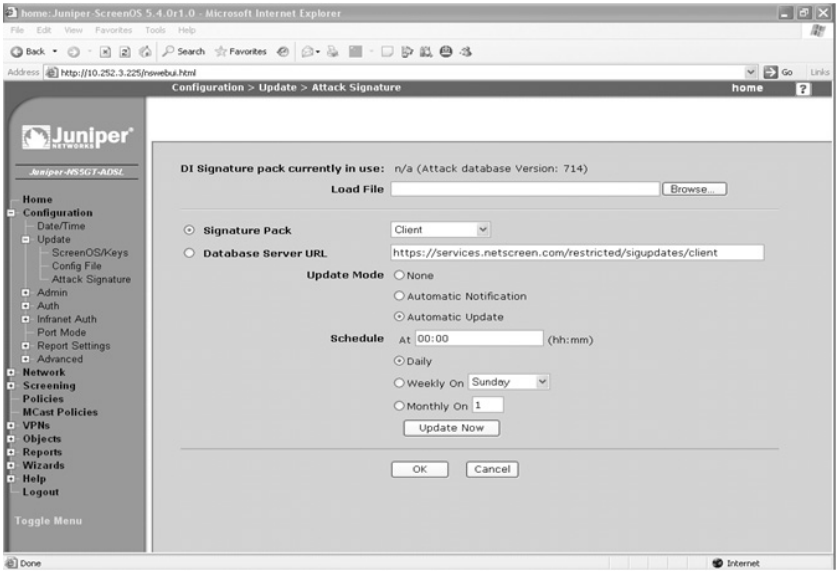
NOTE

Remember that the device has to have HTTPS access to the Internet in order to automatically update. Read on regarding how to perform a manual update if HTTPS access is not possible.

Using the WebUI, access Configuration | Update | Attack Signature. Figure 10.2 shows this screen. The Database Server field is used to select the partial URL from which to download (the current default is still apparently https://services.netscreen.com/restricted/sigupdates; however, since Juniper acquired NetScreen some time ago, I expect this URL may be updated soon). Depending on the choice of base, client, server, or worm signature pack, that will also be part of the URL. I say partial because the latter half of the complete URL is hard-coded as /[version]/[model-name]/attacks.bin (for example, /5.4/ns5gt/attacks.bin). If you're looking to configure an internal DI update server, you can use HTTP or HTTPS, and be sure to put the newest attacks file in a subdirectory called /[version]/[model-name] for the update to work.

The section below the URL entry line allows you three update modes: None, Automatic Notification, and Automatic Update. None turns auto-update off, while Automatic Notification checks to see if there is a new file, but does not download and update. Instead, it puts an entry in the logs that an update is available. If you don't check your logs often, but want to make sure the device always has the most current coverage, you can select Automatic Update, which checks for new signature updates and, when available, automatically downloads and installs them.

Figure 10.2 Automatic DI Signature Update Settings



The remainder of the options for this screen are fairly self-explanatory and include settings for when the device will auto-update (daily, weekly, or monthly), and what time of day to update. There's also a handy Update Now button that allows you to test your settings.

Loading the Database Manually

Sometimes, due to architectural decisions, a firewall may not have direct access to the Internet. In these circumstances, an automatic update may not be possible. Manual updates require you to obtain the update by hand from another system connected to the Internet, and then take that image and manually place it on the device in question.

Using the WebUI, manual loading is a straightforward affair. If you have your DI key properly installed, the Deep Inspection Signature Update field will be available on the Configuration > Update > Attack Signature screen. This field is hidden if the DI key is not installed. Use the Load File field to enter the local path to the signature update file, or use the Browse button to locate and select the file location. Once you have specified or selected the file location, click OK to update the device from the local file.

To perform this action via the CLI, you'll need to download the signature update to a TFTP server. The syntax for the command is save attack-db from tftp [server-IP] [path/filename] to flash. If successful, you'll see a string of dots generated across your console as TFTP packets arrive. Exclamation points (!) mean packet loss or other network error. Missed packets from an otherwise successful stream are present, but if the NetScreen can't connect to your TFTP server, you'll receive a number of exclamation points followed by a TFTP timeout error.

NOTE

While your NetScreen device is signed up for subscriptions, you can update the device as many times as you like. Once your subscription has expired, you'll be ineligible for new updates, but your existing signature pack will continue to work as before. You'll also still be able to create your own custom signatures even if your subscription has run out.

Using Attack Objects

NetScreen-supplied attack objects are organized into groups based upon three criteria: protocol, severity, and type. For ScreenOS 5.0, the only valid severity levels were critical, high, and medium. Beginning with ScreenOS 5.1, the new severity levels of low and info are included. For ScreenOS 5.0, only six protocols were supported: HTTP, FTP, DNS, POP3, SMTP, and IMAP. Beginning with ScreenOS 5.1, this protocol list has expanded to include SMB, MS-RPC, NetBIOS, Gnutella (a popular peer-to-peer file sharing protocol), as well as several instant messaging protocols. With ScreenOS 5.2, even more protocols are added. For type, there are signatures and anomalies. Signatures are specialized regular expression pattern matching strings applied to contexts that then match malicious or other unwanted traffic in network flows. Protocol anomalies are protocol-specific functions that ensure the flow adheres to protocol standards or other settings.

Using Attack Groups

Attacks cannot be used individually in a policy—they must be assigned to a group, even if that group contains just a single entry. If a predefined group has entries in it you don't want to use, you may deactivate them from the group by accessing the **Objects | Attacks | Predefined Groups** window and clicking **View** for the group you wish to edit. You will see a listing of all attacks included in the group. The right-hand column has a checkbox next to each entry. To remove an entry from inspection by the group, remove the checkmark from its checkbox.

Changing active entry settings within a group is a global action that affects the entire device for all policies that use the group. Also note that this does *not* remove the entry from inspection, it only removes it for the purposes of action against the event – there is *no* performance improvement for removing signatures from a group. Likewise, there is no performance impact for using DI groups over and over again in different policies. When DI is on, it's *ON*, and when it's off, it's *OFF*. The first time you use DI in a policy, DI inspection automatically turns on. When it is removed from every single policy, DI automatically turns off.

Enabling Deep Inspection with a Policy Using the WebUI

In order to use DI, you must first create an access policy. The final policy listing summary should be similar to the one shown in Figure 10.3. Figure 10.4 shows a ScreenOS 5.4 policy crafting window in the WebUI. Create appropriate entries for **Source Address**, **Destination Address**, and **Service**. To choose which DI groups will inspect the traffic through sessions matching this policy, click the **Deep Inspection** button.

Figure 10.3 ScreenOS 5.1 Policy Editor, Basic

Figure 10.4 ScreenOS 5.4 Policy Rule Configuration

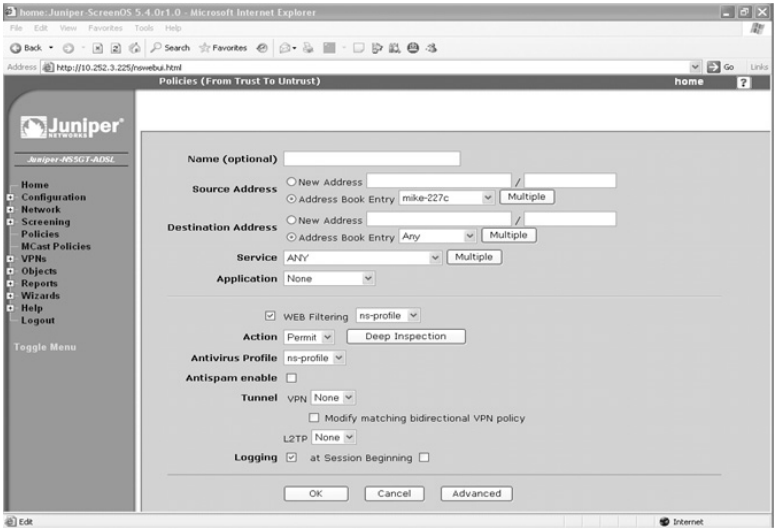


Figure 10.5 shows the Deep Inspection Configuration window.

Here you'll find an unsorted, unfiltered drop list of all available Deep Inspection groups. In DI 5.4, you'll have your pick list of at least 30 items. Following that is a drop-down list that allows you to select which Action to perform on the selected group, as well as a Log option checkbox. Below this, you'll find a table showing the Currently Defined Attack Groups assigned to this policy. Different groups within a policy can have different action and logging settings. This is useful, since there can be only one policy that matches traffic between two hosts on a port—multiple duplicate policies are not permitted. As in ScreenOS, the first matched policy for a connection is used. Click the **Add** button to add the selected group with the selected action and log setting to the defined attacks table. Click the **OK** button when finished. These DI options will be applied to your policy. Click the **OK** button

in the main policy editing window to apply your changes to the policy. The resulting DI-enabled policy has the DI inspection magnifying glass icon in the Actions column of the policy list. Policy 1ID 1 in Figure 10.6 has Deep Inspection enabled, while policy 2 has both Deep Inspection and Juniper-Kaspersky antivirus enabled.

Figure 10.5 Policy Deep Inspection Configuration

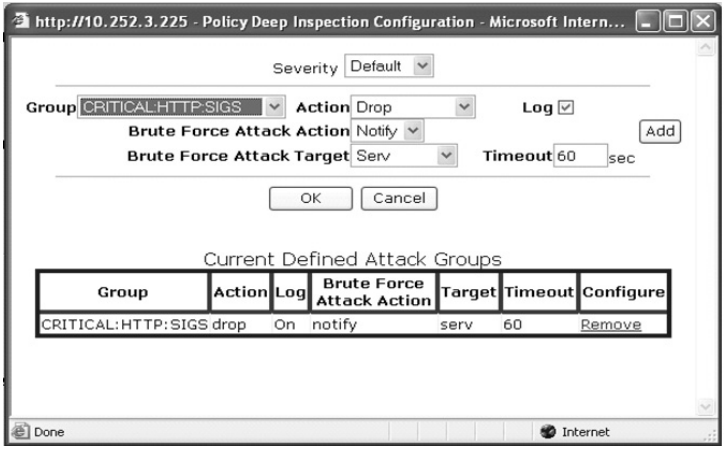


Figure 10.6 Policy Listing with Icons in the WebUI

Enabling Deep Inspection with a Policy Using the CLI

Creating a policy that inspects traffic using many attack groups is a major chore in the WebUI. My personal recommendation is to create the initial policy in the way you feel comfortable—either WebUI or CLI—and then for bulk DI inspection configuration, use the CLI. You'll find this a vastly superior method.

If you are managing multiple firewalls and policies, using NetScreen Security Manager (NSM) will make your task many times easier—simply define an attack and policy once, then specify which devices to apply it to.

To get a listing of which attack groups are available, use the *get attack group sort-by name* command, as shown next:

```
ns5gt-> get attack group sort-by name
Total number of attack groups is 30
```

You should get results similar to those shown in Code Listing 10.1:

Code Listing 10.1 ScreenOS 5.4 CLI Attack Groups Sample Listing

```
Ns5gt-> get attack group sort-by name
Total number of attack groups is 30
Name                                    Type      Defined
CRITICAL:HTTP:ANOM                      group     pre-defined
CRITICAL:HTTP:SIGS                      group     pre-defined
CRITICAL:MS-RPC:SIGS                    group     pre-defined
CRITICAL:NNTP:SIGS                      group     pre-defined
CRITICAL:SCAN:SIGS                      group     pre-defined
CRITICAL:VIRUS:SIGS                     group     pre-defined
HIGH:APP:SIGS                           group     pre-defined
HIGH:DNS:ANOM                           group     pre-defined
HIGH:HTTP:SIGS                          group     pre-defined
HIGH:LDAP:ANOM                          group     pre-defined
HIGH:MS-RPC:SIGS                        group     pre-defined
HIGH:POP3:SIGS                          group     pre-defined
HIGH:SMB:SIGS                           group     pre-defined
HIGH:SMTP:ANOM                          group     pre-defined
HIGH:SMTP:SIGS                          group     pre-defined
HIGH:TROJAN:SIGS                        group     pre-defined
HIGH:VIRUS:SIGS                         group     pre-defined
INFO:DNS:ANOM                           group     pre-defined
INFO:LDAP:ANOM                          group     pre-defined
INFO:MS-RPC:SIGS                        group     pre-defined
INFO:P2P:SIGS                           group     pre-defined
INFO:VIRUS:SIGS                         group     pre-defined
LOW:HTTP:SIGS                           group     pre-defined
LOW:LDAP:ANOM                           group     pre-defined
LOW:POP3:SIGS                           group     pre-defined
MEDIUM:HTTP:SIGS                        group     pre-defined
MEDIUM:LDAP:ANOM                        group     pre-defined
MEDIUM:POP3:SIGS                        group     pre-defined
MEDIUM:TROJAN:SIGS                      group     pre-defined
MEDIUM:VIRUS:SIGS                       group     pre-defined
Total number of attack groups is 30
```

Cut and paste this output to a text file for handy reference. Now let's edit this policy to add DI. From the command line, type **set policy id [x]** where [x] is the ID number of the policy we're editing. Then, press **Return**. This puts us unto policy-edit mode—you'll notice

the command prompt has changed, and added a (policy:x) to the end of the prompt. This lets us know we're editing policy "x." All subsequent commands apply only to our current policy until we use the *exit* command to end policy-edit mode.

From here, it's a simple matter of adding *set attack [attack-group-name] action [action]* to add attack [attack-group-name] to the policy with action of [action]. To enable logging of this group, we need to add another set attack command, this time with logging instead of an action command, like so: *set attack [attack-group-name] logging*. Once you've added all your attack groups, remember to use the *exit* command.

Explanation of Deep Inspection Contexts and Regular Expressions

After using the built-in signatures, I'm sure you're eager to write a few of your own. Before we jump into how to write a signature, we must first cover some basics on how DI looks for patterns, and how to write instructions to make it recognize bad traffic.

As mentioned earlier, Deep Inspection uses contexts to examine relevant portions of network streams for content. In ScreenOS 5.0, a very limited set of DI contexts were exposed to end users. Table 10.2 shows the *only* contexts a user could use to make a signature; many more were available to Juniper signature writers.

Table 10.2 ScreenOS 5.0.0 User-Accessible Contexts

Deep Inspection Protocols	End User Contexts
FTP	ftp-command, ftp-username
HTTP	http-url-parsed
SMTP	smtp-from, smtp-header-from, smtp-header-to, smtp-rcpt

With ScreenOS 5.1, a whole new slew of protocols with new contexts are available, as well as additional contexts for existing protocols (see Table 10.3).

Table 10.3 ScreenOS 5.1.0 User-Accessible Contexts

Deep Inspection Protocols	End User Contexts
AOL Instant Messenger (AIM)	aim-chat-room-desc, aim-chat-room-name, aim-get-file, aim-nick-name, aim-put-file, aim-screen-name
DNS	dns-cname
FTP	ftp-command, ftp-password, ftp-path-name, ftp-username
Gnutella Peer-to-Peer Protocol	gnutella-http-get-filename

Continued

Table 10.3 continued ScreenOS 5.1.0 User-Accessible Contexts

Deep Inspection Protocols	End User Contexts
HTTP	http-authorization, http-header-user-agent, http-request, http-status, http-text-html, http-url, http-url-parsed, http-url-variable-parsed
IMAP	imap-authenticate, imap-login, imap-mailbox, imap-user
Microsoft Network Chat (MSN)	msn-display-name, msn-get-file, msn-put-file, msn-sign-in-name
Post Office Protocol ver 3 (POP3)	pop3-auth, pop3-header-from, pop3-header-line, pop3-header-subject, pop3-header-to, pop3-mime-content-filename, pop3-user
Server Message Block/Common Internet File System (SMB/CIFS)	smb-account-name, smb-connect-path, smb-connect-service, smb-copy-file-name, smb-delete-filename, smb-open-filename
SMTP	smtp-from, smtp-header-from, smtp-header-line, smtp-header-subject, smtp-header-to, smtp-mime-content-filename, smtp-rcpt
Yahoo! Instant Messenger (YMSG)	ymsg-alias, ymsg-chatroom-message, ymsg-chatroom-name, ymsg-nickname, ymsg-p2p-get-filename-url, ymsg-p2p-put-filename-url, ymsg-user-name

There's not enough space to cover all of these contexts in detail, so in the next section we'll hit the highlights and give some examples of what you can do with some of the more popular contexts. A complete context reference can be found in the Juniper documentation.

Before we can talk about writing signatures in contexts, we need to cover the pattern matching syntax, also known as DFA (Deterministic Finite Automaton) syntax (see the following "Deep Inspection Search Algorithm" sidebar). NetScreen DFA syntax is similar to regular expression syntax, but not quite the same. Next, we'll cover the basics of how NetScreens match patterns.

The most straightforward way of looking for data would be an exact match. For example, to find the exact byte-pattern of root in a context, we would simply type *root*. Note that if the context presents any additional data, like rooter or, if the capitalization does not match, like Root or rOoT, then this simple match string will *not* match.

In order to insert special matching commands within a search string, the commands have to be identified as commands instead of just more matching text. This command delineation method is commonly known as *escaping*. In DFA syntax, commands are identified by a preceding backslash (\).

Sometimes, an exact string match is what you want. Most often, however, you want to detect variations and permutations of strings. For a case-insensitive match of alphabetic characters, enclose the string within escaped square brackets. Our earlier search for root with case-insensitive added would be \[root\]. Failing to close the case-insensitive range with an ending delimiter will cause the signature not to work.

This is useful, but what if we need to match this string at the beginning of the stream and more information comes after it (such as in our previous rooter example)? For this, we turn to our good friend, dot-star. The dot is used to match any one-byte value (in order to match a literal dot, it must be escaped, like so: \). Star means *zero or more of the previous match* (again, to match a literal asterisk, it must be escaped like so: *). Put these two elements, dot and star, together, and it will match zero or more of anything, which is quite handy. For example, \[root\].* matches Rooter and rooTMan, but not iamroot. For that last match, a dot-star at the beginning is the trick. For example, .*\[root\] matches nicely, as well as .*\[root\].*, which will also match IamRootMan. Many Juniper-authored signatures work exactly this way.

NOTE

The dot-star implementation used by Juniper's IDP and DI is different from the common RegEx implementation. Standard (java/perl/grep) RegEx treats .* as a greedy match; thus, if you put dot-star at the beginning, it will always match. Juniper's implementation is a bit more intuitive, but may surprise someone who is already familiar with using a posix regep.

While these work great for ASCII character matches, many protocols use non-ASCII bytes. There are two ways to match arbitrary binary data in DI: hexadecimal (hex) and octal representations. For hex, DFA uses an escaped X (for heX), while for Octal, DFA uses an escaped zero (0), which represents the letter O in Octal. Another fundamental difference between these two methods is that an octal match always represents a single byte (so the maximum value is \0377, two bytes would be \0377\0377), while a hex match always represents one or more bytes with \x delimiting the start and end of the range of characters to be evaluated as hex (that is, \xff\x or \x0123456789abcdef\x). White space within the hex range is ignored, so you can space out your match characters by nybbles, bytes, words, or something else. For example, you can enter *\x 0123456789ABCDEF \x = \x 01 23 45 67 89 AB CD EF \x = \x 0123 4567 89AB CDEF \x*. Failing to close a hex range with an ending delimiter will cause the signature to not work.

There are times when an attack will have one or more methods for gathering the same results, or perhaps you want to combine similar signatures into a single entry. In order to define elements of a match string, parentheses are used, but they are *not* escaped. To use literal parentheses in a match, you must escape them like so: \(\) or use their ASCII hex or

octal values. Don't confuse escaped parentheses with case-insensitive matching brackets that *must* be escaped in order to work so they are not misinterpreted as a character class (see the following). When selecting one or the other of a series of options, we use the pipe (|) character for an *OR* operator. That is, match A or B using the entry (A|B). Several ORs can be chained together—any one of them will match: (A|BC|DEF). Note that they need not be a single byte, nor have the same amount of bytes.

There are times when you might want to match a large range of characters that would make ORing them altogether entirely impractical. For example, "all capital letters" would be (A|B|C|D|E|F|G and so on until |X|Y|Z), which is way too long, and makes reading difficult. To solve this dilemma, we have the character class feature. Character classes use unescaped brackets with a list of characters or a single character range to match on. [A-Z] would solve our "all capital letters" problem. For an arbitrary character class, merely add the characters within brackets in any order: [ABCcba]. Character classes also allow for octal codes for non-printable byte value matches: [\000-\017] or [\011\013\020], and so on.

Yet another use of the character class is to define values *not* to match. This is known as a *negate character* class. To negate a character class, merely place a caret (^) as the first character inside the class. This will *not* match on a caret—it will negate the remainder of the character class. In ScreenOS 5.0, only a single character is allowed after the caret, while in ScreenOS 5.1, multiple characters are permitted—for example, [^A] or [^123]. A common state-saver to the traditional dot-star in the middle of a match string (see the "Deep Inspection Search Algorithm" sidebar) is a not-space-star, or [^]* match string.

The question mark (?) makes the directly preceding match optional. For example, *html?* matches html as well as just htm. This is also handy for using with parentheses to make an entire element optional. For example, *super(duper)?man* matches both *superman* and *superduperman*.

One final major matching syntax we'll cover before jumping into signature writing is the unicode decoder. Many Windows protocols, like SMB, NetBIOS, and MS-RPC, can use either traditional ASCII encoding or the new international-friendly unicode encoding. To convert ASCII to unicode, nulls (\000) are inserted after every character. Traditionally, it was very messy to make a string match both normal ASCII and ASCII in Unicode. For example, to match *Windows* would require *W(\000)?i(\000)?n(\000)?d(\000)?o(\000)?w(\000)?s(\000)?*, which is almost unreadable. The same match using the unicode decoder is merely *\uWindows\u*. Be sure to close your decoder with a second *\u*, otherwise the signature won't work.

Table 10.4 includes a quick reference to the aforementioned match strings.

Table 10.4 NetScreen Search String Syntax Summary

Match String	Usage Notes and Syntax
.	The dot character matches any one byte. When a literal dot match is needed, try escaping the dot like so: **www\.juniper\.net**.
*	The asterisk (or star) matches zero or more of the preceding match. When a literal asterisk is needed, try escaping it: *

Continued

Table 10.4 continued NetScreen Search String Syntax Summary

Match String	Usage Notes and Syntax
.*	Dot-star is a useful combination that matches zero or more of any characters. Place at the beginning of a match string to search anywhere in the context. Place at the end of a match string to ignore any additional data after the matched string. Remember that Juniper's implementation of dot-star is *not* greedy.
+	The plus sign character matches one or more of the preceding match. For example, **AA+** matches **AAA**, but not **AA** or **AAB**.
?	The question mark makes the preceding character/element an optional match. For example, **html?** matches both **htm** and **html**.
\xAB CD\x \XABCD\X	Matches hexadecimal values. Be sure to close your decoder with a second **\x** or the signature will not work.
\0oct	Slash-zero matches a single byte of octal values \000 through \0377. Permitted octal characters are 01234567 only.
\[match\]	Case-insensitive search. Alphabetic characters are compared with both upper and lower case. For example, **\[dog\]** matches **dog**, **DOG**, **Dog**, **dOG**, and **DoG**.
( )	Parentheses are used to group portions of match strings into a single element. Parentheses are also useful with the pipe character for OR comparisons. For example, **AA(AA|BB)BB** matches **AAAABB** or **AABBBB**. For a literal parentheses, try escaping them like so: **\(\)**
[abc123] [a-z] [\0123-\0321]	Character class. Counts as a single byte that matches any symbol or symbol range inside. Cannot be used inside a case-insensitive search. For example, **\[abc[def]ghi\]** is illegal. Instead, try **\[abc\]\[def\]\[ghi\]**. Also note that multiple ranges are not allowed, such as **[a-zA-Z]**. Octal is also supported in order to define non-printable ranges.
[^abc] [^] [^\000]	Negated character class. Counts as any single byte that does *not* match the contents inside the brackets. Note that octal is still supported in negated character classes. ScreenOS 5.0 DFA only supports a single character in a negate character class, while ScreenOS 5.1 DFA supports multiple characters.
\s	New in ScreenOS 5.1 is the white-space character, or slash-s. This matches a single space or tab. For ScreenOS 5.0, in order to match the same value, an octal OR group was used: **(\011|\020)**. This is much easier to read.
\uUnicode\u	New in ScreenOS 5.1 is the unicode decoder, or slash-u. Be sure to close your decoder with a second **\u** or the signature will not work.

For an exhaustive reference to regular expressions, I highly recommend the O'Reilly book, *Mastering Regular Expressions, 2nd Edition*, by Jeffrey E. F. Friedl (ISBN: 0-596-00289-0).

Tools & Traps…

Deep Inspection Search Algorithm

NetScreen IDP and DI both use a method of searching traffic for malicious patterns very quickly using a technique known as a *Deterministic Finite Automaton*. A simple explanation of DFA is a tree of all possibilities the search is looking for, combined into a logical table where similar matches are grouped together and searched simultaneously. When a difference between two unique patterns occurs along the line, the search line forks and each subsequent possibility then gets its own line. The total unique search lines these forks create are known as states.

A DFA with a large number of states takes significantly longer to parse though to find a match. Regular expression symbols that generate a large number of states are the wildcard symbols ., *, and +, and the conditional symbol ?. When placed in the middle of a match, they can expand the number of states exponentially, severely impacting performance and memory. Use these symbols sparingly in the middle of your signatures. Using them at the beginning or end of your signature does not add states and is actually a handy way to scan for a match where the beginning of the stream of information to match is unknown or variable.

Creating Your Own Signatures

Now that we've covered the two major aspects of signature creation—contexts and syntax—let's put them together and write a few signatures! This section will cover a few of the more popular contexts with some RegEx usage on how to get the most out of them.

To make a new custom signature, access **Objects | Attacks | Custom**, and click the **New** button. This opens the signature editor window, which has five fields: Attack Name, Attack Context, Attack Severity, Attack Pattern, and Negate. Also note that custom signature names must start with the string CS (for custom signature). We'll be using this window as we experiment with some of the more common contexts in the following.

HTTP is the most common protocol, and by far the highest bandwidth consumer. Adding new HTTP signatures impacts the performance of this already heavily burdened protocol, so add new signatures here with care, and try to avoid high-state wildcards (see the "Deep Inspection Search Algorithm" sidebar earlier) in the middle of signatures and after common matching strings.

To understand how the HTTP contexts work, we need to first examine the HTTP protocol itself. HTTP is a stateless client-server protocol, where a server generally supplies files based upon requests by the client. In addition to the file transfer itself, several protocol-related data exchanges occur, mostly at the beginning, before the actual file transfer. These are known as *HTTP headers* (inspected by the *http-header-user-agent* and *http-authorization* contexts). The client request itself is called a *Uniform Resource Locator*, or URL. The URL is generally broken down into two elements: the path/file and the parameters/variables. The path and file includes all characters after the request verb but before the question mark, exclusive. The parameters (also called variables) include everything after the question mark, also exclusive. DI has an *http-url-parsed* context, as well as an *http-variable-parsed* context. These contexts take any kind of URL obfuscating encoding and parse it the same way the end server would, afterward applying the signature against the result. This allows us to write a nice, clean URL signature without worrying about encoding schemes or other kinds of IDS evasion techniques. If such encoding attacks are what you're looking for, there's also the unparsed *http-url* context for just the URL, or *http-request* for the entire request, completely unparsed. Let's try some practical examples of these contexts using the sample exchange that follows:

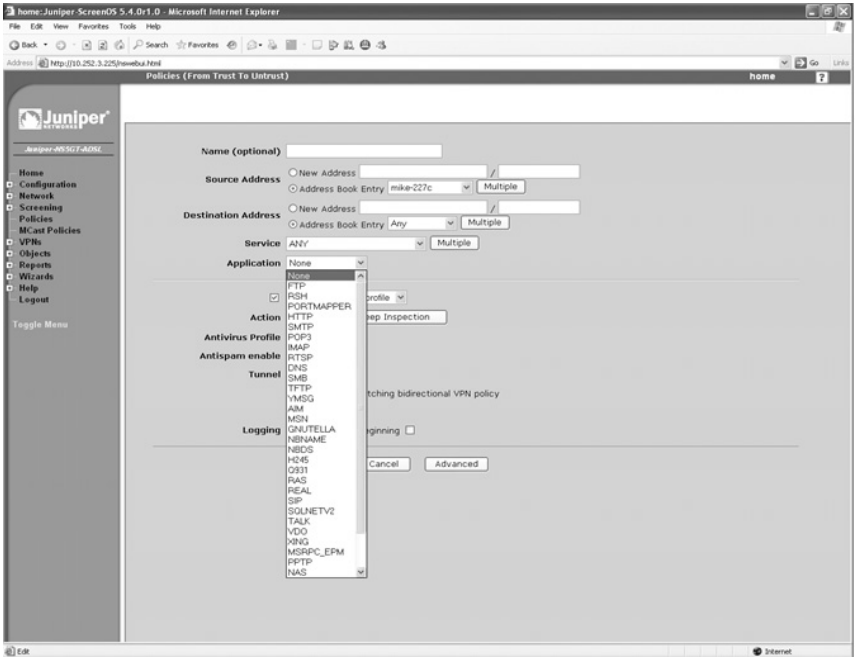
```
The client requests:
GET /etc/pass%77d?bar=yes HTTP/1.1
User-Agent: HappyBrowser v1.1
Host: www.foo.com
Authorization: Basic dXNlcjpwYXNzd29yZAo=

The server responds:
HTTP/1.1 200 OK
Date: Sat, 25 Dec 2004 00:00:01 GMT
Etc…
```

Notice the %77 in the URL? That decodes to an ASCII *w*, making the path */etc/passwd*; someone was trying to hide the true name of the filename he or she was requesting. This is a fairly common evasion method and it's easily defeated by the *http-url-parsed* context since the context itself normalizes (parses) the URL before inspecting it. To match this attack, merely enter **\[/etc/passwd\]** (remembering to add case-insensitivity to catch further evasion) as the attack pattern with an attack context of **HTTP Decoded HTTP URL**. Name this something useful, starting with the **CS:** identifier, such as **CS:HTTP:ETC-PASSWD**, then assign it a severity (like **Medium**), and you're done! Note we did not need a dot-star at the end of this pattern to account for the *bar=yes* parameter since the *http-url-parsed* context stops before the question mark that delineates path from parameter. This is very similar to the Juniper-supplied signature HTTP:INFO-LEAK:HTPASSWD-REQUEST, whose match pattern is *.**/\.\[htpasswd\]. The important differences are that this signature is looking for the file .htpasswd at the end of any path, which is covered by the dot-star and a forward slash. The dot in .htpasswd is also escaped for a more accurate match.

Sometimes, you may want to match a particular protocol's traffic (for example, HTTP or FTP) on a port other than the typical ports used by that protocol. This is where the *Application* setting in the policy editor (see Figure 10.8) comes in handy. The *Application* setting is a list of all Application Layer Gateways (ALGs) and DI protocols parsed. The *Application* setting activates all ALGs and DI contexts for that protocol on whatever the service (predefined or custom) is set to. So to use FTP (with dynamic-gate ALG support) on a nonstandard port, you would merely create a custom service for the command port used and then bind the FTP application setting to it. All traffic permitted by this policy will be inspected for FTP protocol conventions, including PORT commands that will be used to open data connection gates. Additionally, any FTP-based DI anomalies or signatures assigned to the policy in the Deep Inspection editing window will also be applied to this custom service.

Figure 10.7 Policy Listing with Icon

Figure 10.8 Policy Editor Application Field



Now let's tackle another protocol—Transmission Control Protocol. "But that's not an Application Layer Protocol!" you shout. And you're quite right. But there is a single context that can be bound to any TCP service supported by DI called *stream256*, and so for lack of a better description, this is a TCP context. stream256 is a very simple, but very powerful, con-

text. It should be used with care. This context is powerful since it gets around the problem that certain useful contexts may not be customer-accessible. It's also useful for inspecting protocols not supported by DI. The downside is that it only inspects the first 256 bytes of the stream, but you'd be surprised what you can do with 256 bytes. It inspects any DI-supported service bound by policy. In order for stream256 to work, however, an *Application* setting (other than None or Ignore) *must* be selected. Be sure *not* to use a dynamic-port protocol like H.323, SIP, PORTMAPPER, MSPRC-EPM, or FTP, unless you are specifically writing a signature for these services, since the ALG will try to unnecessarily parse the protocol for dynamic-gate opening, and will slow traffic going through it. That is, if you're writing a stream256 signature for HTTP, be sure to apply the HTTP *Application* setting. For non-DI supported protocols, use an uncommon *Application* setting like TALK to make stream256 work best.

The majority of the other contexts are fairly simple and self-explanatory. A few quick comments on some of the more interesting contexts:

- **dns-cname** examines just the DNS hostname of a DNS request—for example, www.juniper.net.
- **ftp-pathname** includes both the path and the filename. Add a dot-star to the beginning of the search pattern to look for files downloaded via FTP.
- **http-authorization** automatically decodes the Authorization: Basic Base64 code into a username:password format. Use *\[username\]:.** to look for specific usernames and *.*:password* to look for specific passwords.
- **http-header-user-agent** can be used for browser identification, especially peer-to-peer and spyware applications that use the Web. Quite a few of them aren't clever enough yet to hide their actual program names. Use *Gator.** to match the Gator spyware program.
- **pop3-user** and **pop3-auth** provide you with username and password, respectively, for POP3 users.
- **smtp-rcpt** is essentially the To: field of an SMTP mail exchange. Good for looking for specific destination e-mail addresses.
- **smtp-mime-content-filename** provides filenames to attachments in e-mails sent via SMTP.

Also recall that signatures cannot be used by themselves; they must be incorporated into an attack group before they can be employed. So before we add these attack objects to a policy, we need to make our own custom attack group. Attack groups must begin with the letters *CS:*, for example, *CS:HTTP-ATTACKS, CS:FTP-SIGS*, and so on. Create a new custom group by accessing **Objects | Attacks | Custom Groups** and clicking the **New** button. Once in the new group creation window, you'll see a **Group Name** field (again, you must start with **CS:**), as well as the **Selected Members** and **Available Members** lists.

Merely select which custom signatures (Juniper-supplied signatures cannot be assigned to a custom signature group) you want this group to contain and click the << button to move them over to the **Selected Members** list. Remove selected members with the >> button. Once you have this new custom signature group, you're ready to use it in a policy, just like any other attack group.

Tools & Traps…

Advanced DI Signature Writing Using an IDP

One of the more powerful features of DI is the ability to write your own signatures for it. But one of the frustrating things about this feature is the fact that you can't see quite how DI will actually inspect the traffic within its contexts. Does it include the HTTP GET verb? Are the parameters parsed, or just the path?

Since DI is truly a subset of the IDP feature set, if you have an IDP handy, you can use it for advanced DI signature development. This is the same technique used by the actual DI and IDP signature writers at Juniper Networks to develop production signatures. This can be used on a production IDP with minimal impact to network performance, but because of the volume of data provided, you might want this to be done on a lab IDP instead. Also note that the Environment Security Profiler (ESP) feature of the IDP (disabled by default) will have to be turned off before this technique can be used.

To start, log in to your IDP remotely via SSH and become the root user. Using the command scio ccap all will show you every context that the IDP is parsing against traffic currently flowing through the device, as well as exactly how that traffic is parsed. Press Ctrl + C to stop the display. While displaying, this will slow the unit some, especially if there is a lot of traffic. You can either limit the services to be examined by replacing the all with a service limiter command such as scio ccap svc http or scio ccap svc ftp. The command has a help system to show you valid services, but for the purposes of DI, remember that the services are somewhat limited.

Once you have your ccap (short for Context Capture) running, execute an attack you're interested in blocking through the IDP, and watch it parse your attack into contexts. At this point, you should be able to write a regular expression string to match your attack. Keep in mind that some contexts that exist on an IDP are limited in DI and may not be available. Work backwards from the contexts that are available in DI and find out how it is examined using an IDP. Since DI and the IDP view these contexts in exactly the same way, you can be reasonably certain that if it detects it on an IDP, it will work on DI.

Setting Up Content Filtering

Juniper Networks' NetScreen firewalls support content filtering through two major methods: Web filtering and antivirus. Of course, Deep Inspection could also be used as a sort of content filtering, but it's not quite as suitable since it doesn't present the end user with an appropriate error page when a violation is detected, as the URL filtering feature does.

Web Filtering

Content classification systems are one of the earliest forms of access control. Labeling systems or data as secret, classified, or unclassified, and filtering based on user security clearance level, has been around a long time.

- Malicious Web sites may embed attacks, but just the act of browsing downloads code or data exploits. Web sites for some subject classifications, such as hacking and adult content, have less regulation and trustworthiness than other classifications, like shopping or banking. Malicious Web sites may also spoof another Web site to trick a user into revealing confidential information. Some Internet neighborhoods are just safer than others.
- Web sites may contain content that some user populations require or want protection from, including adult, hacking, hate, and violent content. Some groups require the practice of parental controls or even public censorship.

Web filtering is a subject matter rating or classification system, not unlike the motion picture rating system, except with many more categories. Web filtering works by permitting access to content by subject matter classification, and denying or fencing off the rest. Web filtering is used extensively by public institutions like schools and libraries that are required to exercise oversight over content according to the maturity of their users, local standards of decency, or other community norms. Web filtering is also used by private enterprises like businesses to enforce a nondiscriminatory and nonhostile workplace.

Web Filtering Concepts

Typical security policy permits Web browsing. You can't just block user Web access, because that's what the Internet is for. But permitting good content and blocking bad content is what Web filtering is all about. Often, you cannot come up with a list of explicitly permitted or denied sites by yourself. Perhaps you want to allow or deny access to certain sites based upon a category such as Shopping, Gambling, or Hacking. Web filtering does not quarantine or disinfect content, rather it blocks access *to* the content. The blocked content is replaced with a notification message which the user will see instead of the intended Web page.

Web filtering is the process of examining HTTP requests for content. Requests to inappropriate sites like those that host pornography, racism, or other offensive pages can be blocked using this feature. This works by comparing the requested URL against a database of

classified sites. URLs can be categorically permitted or denied with a variety of configuration settings, which depend on the filtering server/service used. These services charge a subscription fee for updates to the database, since new sites are constantly created on the Internet. The Web filtering software provides Internet usage reports and also keeps track of repeat violators.

Web filtering primarily protects the users by inspecting the upstream, or client-to-server flows, for content addresses. Web filtering also helps supervise Internet browsing, because the administrator is notified immediately of attempted policy violations through log events.

Web filtering provides an engine for the following application protocol: HTTP:URL (Internal and External). It can not inspect encrypted (HTTPS using SSL protocol) traffic, however.

Web Filtering Planning

ScreenOS supports external Web filtering on all platforms, and internal Web filtering on selected platforms. Internal Web filtering is generally supported on both the newer and older branch office platforms, while external antivirus can be supported through redirection across the entire product line.

Starting with ScreenOS 5.1, NetScreen firewalls support two different Web filtering redirection protocols—the SurfControl Redirect protocol, as well as the legacy WebSense Redirect protocol—but not both at the same time. ScreenOS 5.1 and later also support a special SurfControl *integrated mode* on the NS-5GT, NS-25, NS-50, and new SSG platforms. This loads the filter database directly on the device.

Table 10.5 shows which versions of Juniper Networks firewalls running ScreenOS 5.4 offer Web filtering.

www.syngress.com

Table 10.5 Web Filtering Support by Juniper Firewall Model*

	NS-5GT	NS-25/50	NS-204/208	NS-500	NS-5200/5400	SSG-5	SSG-100	SSG-500	ISG-1000	ISG-2000
SurfControl Embedded	Yes	Yes	No	No	No	Yes	Yes	Yes	No	No
SurfControl Redirect	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
WebSense Redirect	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

* Data is based on Juniper Networks firewalls running ScreenOS 5.4.

The internal Web filtering needs a device subscription service and requires the setup of a name service and the correct time to update the filter database. Activation also requires taking a Web filtering action in the policy. External redirection does not require device subscription since that is handled on the external Web filtering server itself.

You should plan ahead and purchase the Web filtering subscription, which you can then associate online with your registered device. The online service reserves a Web filtering subscription and your device only needs to connect and authenticate to the service to retrieve its Web filtering subscription key.

Web Filtering Configuration

In this section, we'll discuss Web filtering configurations with WebSense Redirect Mode.

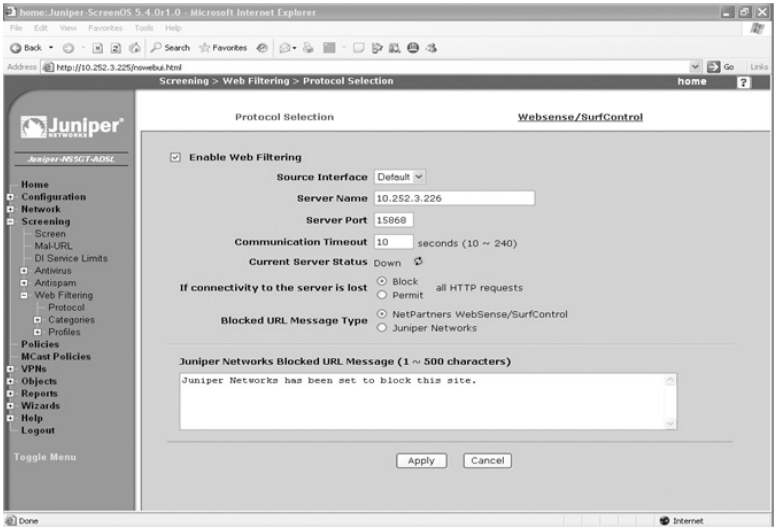
WebSense Redirect Mode

WebSense Redirect Mode is the only method of Web filtering for ScreenOS 5.0 and earlier. It's fairly straightforward to configure on the NetScreen device—the majority of work required is on the WebSense server side, configuring users, user groups, policies, and exceptions. To use WebSense with ScreenOS 5.1, select this as protocol to use for Web filtering.

WebSense requires a Web server like IIS or Apache in order to operate. Usually that means installing it on Windows 2000 or 2003 Server, although it will run on Windows 2000 or XP Professional with Apache installed. If a Web server is not found, the installation will automatically download and install Apache for you if you permit it to. Setup and installation is easy. The minimum system requirement is 512MB of RAM, but 1GB at least is strongly recommended. See www.websense.com/products/about/Enterprise/ for more product details. They offer a 30-day free trial.

From the firewall side of things, setup is a snap. See Figure 10.9 for an example of a firewall configured for WebSense Redirect Mode. First, enable the **Enable Web Filtering** option. Next, type the IP address (or DNS domain name if configured for your WebSense server, and your NetScreen has a DNS server configured) into the **Server Name** field. Enter the port number that the WebSense server is listening on for URL validation requests in the **Server Port** field (the default port is 15868). Now set a reasonable **Communication Timeout** value (10 is the default, but you may need more if the WebSense server is being accessed via a VPN). At this point, if the server is up and properly configured, clicking the yellow circle next to **Server Connection Status** should show that the server is running. If it doesn't work right away, give it 10 to 15 seconds and try the yellow circle button again. If the server is still down, check your settings (including DNS!), your routes, and your server. Try to Ping the server by IP from a CLI prompt on the NetScreen if you're having trouble getting the system to work.

Figure 10.9 A Web Filtering Configuration with WebSense Redirect Mode



There's also an option to set the behavior of the firewall in the event that the URL server cannot be contacted. Using this option to either block all HTTP or permit all HTTP is a policy decision you'll have to make on your own, depending on your business requirements and the stability of your WebSense setup.

SurfControl Redirect Mode

SurfControl Web Filter for Juniper Networks Security Devices is a competitor to WebSense, and with ScreenOS 5.1 you now have a choice of URL filtering services to select. Like WebSense, SurfControl will work with 512 Megabytes of RAM, but would prefer 1GB or more. SurfControl requires either an external MS-SQL database or an internal Microsoft Desktop Engine 2000 (MSDE2000) database. If an MSDE2000 database is not already installed, SurfControl will download and install it for you, similar to how WebSense handles a missing Web server—very handy. Like WebSense, they also offer a 30-day free trial. After using both, I found WebSense easier to set up and configure, but unlike WebSense, SurfControl has an Integrated mode that uses public servers (which means no local installation!) that we'll examine next.

Since it's essentially the same concept, the configuration settings for SurfControl Redirect Mode are the same as WebSense Redirect Mode. To use SurfControl with ScreenOS, you'll first need to select it as the protocol to use for URL filtering. Then fill in the options as you would in WebSense, such as turning on Enable URL Filtering, setting a Server Name, Server Port (the default for the Surf Control Filter Protocol (SCFP) is 62252), and a Communications Timeout value. Now it's time to check the server availability. Click the yellow circle to ensure you've set everything up correctly. Try some of the trou-

bleshooting tips found earlier in the "WebSense Redirect Mode" section if things aren't working right.

SurfControl Integrated Mode

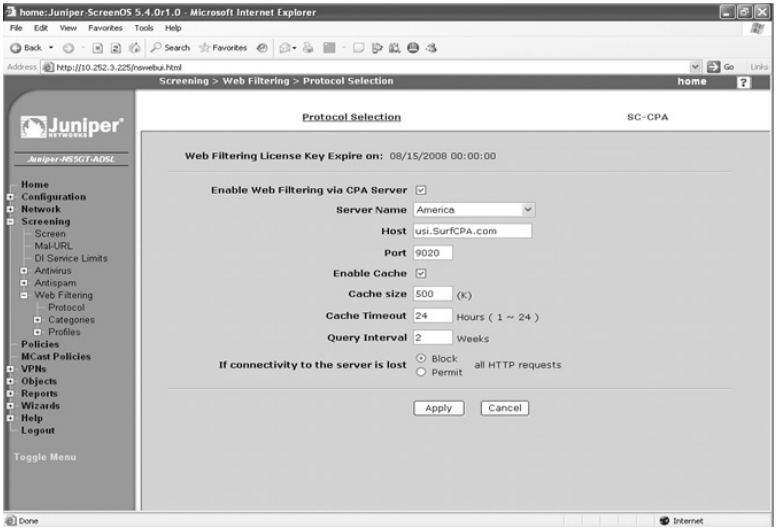
This mode is only available on the low-to-midrange model NetScreen firewalls—the NS-5GT, NS-25, NS-50, and new SSG platforms. SurfControl Integrated Mode also requires a subscription. To use SurfControl, you'll first need to select it as the protocol to use for Web filtering. The protocol used for this mode is the SurfControl Content Portal Authority (SC-CPA) protocol.

Once Integrated Mode is selected, configuration options for this mode appear. See Figure 10.10 for a reference. These options include the following:

- **Server Name** A drop-down list that allows you to select in which major continent (America, Europe, Asia) the device is located so the closest SurfControl server is selected.
- **Host** The actual hostname for the server to use. This value is automatically filled in from the Server Name field, but can be overridden.
- **Port** The port to use for communication with the URL filtering server database. SC-CPA's default port is 9020.
- **Enable Cache** This allows the NetScreen device to cache the results of SC-CPA lookups, decreasing the response time to end-user requests.
- **Cache Size** The size (in kilobytes) allocated for the cache.
- **Cache Timeout** The length of time an entry will age off the cache if not requested.
- **Query Interval** How often the NetScreen device will check with the server for major category updates.
- **Permit** or **Block** Included as a default fallback decision if the server does not respond to a request.

Integrated Mode URL filtering supports the concept of *blacklists* (always deny regardless of classification) and *whitelists* (always permit regardless of classification) rights on the device. These lists, as well as custom URL lists, are created by accessing **Screening | URL Filtering | Profile | Custom List** and clicking the **New** button. In the custom list edit window, add a name to this category (Whitelist, Blacklist, Competitors, and so on) and then add your first URL and click the **Apply** button. The category name will be saved and locked, and the URL added to the list. Add more URLs (up to a maximum of 20) by entering them in the URL field and clicking the **Apply** button. When you are done adding URLs, click the **OK** button to save.

Figure 10.10 URL Filtering Configuration with SurfControl Integrated Mode



Web Filtering Rules

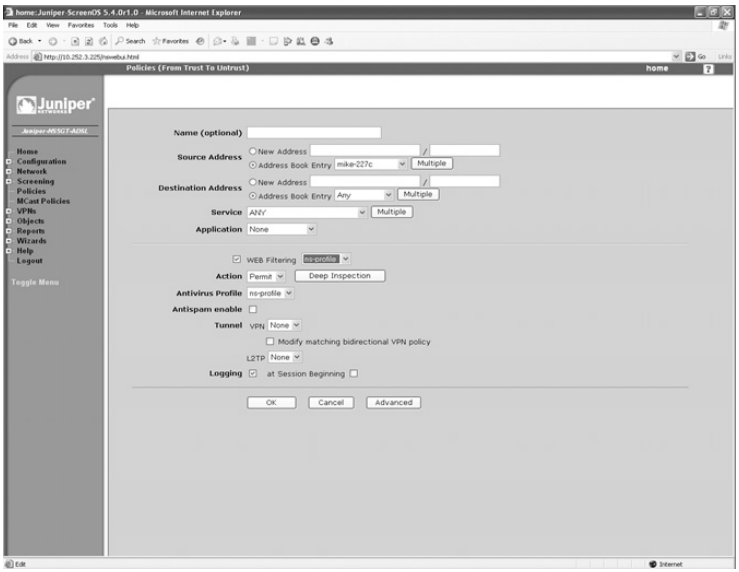
In order to instruct the NetScreen device as to which sources and destinations are to be inspected for Web filtering, a policy definition is required. The final policy listing summary should look similar to Figure 10.11. Refer to Figure 10.12 for the basic policy editing window. Note that at the top of the second section there is a checkbox to enable Web filtering for the policy. For Integrated Mode filtering, Web filtering protection is activated by associating a Web filtering profile with each rule. Filter decisions (called a *filter profile*), based upon which category the URL matches, are made on the device itself. If no custom filtering profile exists, then the default *ns-profile* is compared and action is taken. This profile is a read-only template that can be cloned and the resulting copy modified and used. If a custom profile is created, either from scratch or cloned from the ns-profile, the inspection process becomes a little more complicated. As URLs are offered to the NetScreen for inspection, it first checks the user-defined whitelist, then the user-defined blacklist, and then the NetScreen checks the URL for categorization. Once it knows the URL's category, it first compares it to the user-defined filtering profile. Any match against the user profile is acted on as configured, and the default profile check is skipped. If the URL category matches no defined categories in the user-defined profile, the user-defined profile default action is set to Permit, a final check against the ns-profile is performed, and whatever action is set for that category is carried out.

For Redirected Mode filtering, filter decisions (based upon which category the URL matches) are handled by the remote filtering server. This allows for advanced policy editing, including the ability to override a profile based upon a user login, or different profiles based upon the time of day or day of the week. These features are offered by the third-party filtering servers, so check their product sheets for specific filters.

Figure 10.11 Web Filtering Policy Listing with Icon

mike-227c	Any	ANY			

Figure 10.12 Web Filtering Policy Configuration



For each rule with Web browsing that needs Web filtering protection, associate the Web filtering profile. For instance, the ns-profile represents the default subject rating permissions.

Verify Web Filtering Protection

How do you test whether your Web filtering protection is configured and operating correctly? You don't necessarily want to browse banned sites. But you can create a harmless content address you can include in your blacklist and whitelist, and exercise the Web filtering engine and URL matching. (You can create test URLs at www.juniper.net/blacklist and www.juniper.net/whitelist.) The Web site should exist, but the path should not.

In fact, the test Web site should exist because the client first connects to a Web server before the URL request is made, but the test URL should not exist. If your Web filtering is operating correctly, when you go to the blacklist URL, in your Web browser you should see the following message:

> Your page is blocked due to a security policy that prohibits access to category N/A.

And your logs should contain the entry:

```
UF-MGR: URL BLOCKED: 10.252.3.237(1514)->207.17.137.229(80),
Your page www.juniper.net /blacklist is blocked due to a security policy
that prohibits access to category N/A action: BLOCK, category: N/A,
reason: BY_BLACK_LIST
```

When you go to the whitelist URL, you should succeed.

Antivirus

Viruses were one of the earliest computer threats. Similar to other automated threats today, viruses are a form of malicious code, designed to execute on your system. The first viruses spread by piggybacking on boot files on infected media. This was before networks, so computer viruses required infected media to spread. Since then, viruses have attached themselves to every kind of executable file, and have spread through file shares, Web file downloads, and e-mail attachments. Viruses have also gotten very good at hiding themselves. They disguise themselves by wrapping themselves in packages and archives, and encrypting themselves, to change their appearance. Virus writers have even developed virus writing frameworks and toolkits, making virus writing easier. Indeed, ever since the early days of computers and the Web, tens of thousands of viruses have propagated across the Internet.

A computer virus has similar traits to a biological virus. A virus is a program that "attaches" itself to a file, generally an executable file. How does a virus work? A virus replicates, and does something malicious, known as its objective. Often, this objective is further system compromise by installing other spyware, keyloggers, or spam engines. Viruses piggyback on real programs, often rewriting parts of them. (File size checks, for a known set of programs, can detect a virus, but files are updated by new versions and patches.) Viruses also usually try to hide themselves, with layers of packaging and encryption.

Antivirus software was one of the first security products. The antivirus engine decodes file protocols, which are the file formats that contain your executables and data. The antivirus engine buffers a file, decompresses archives, and decodes the file protocol. The antivirus software then scans the files for virus signatures, which are written centrally, and automatically updated. When an infected file is found, it is quarantined or removed.

While early viruses piggybacked on executable files, virus writers have since taken advantage of security holes in data files. If data files either directly support instructions (supporting a scripting language) or have code vulnerabilities like buffer overflows, even a data file can be infected by a virus.

In all cases, though, infected files must be opened in order to run. (However, it is often easy through file renaming or message content to trick a user into running or opening a file.)

For virus advisories about the latest active viruses, visit the Juniper Security Center at http://security.juniper.net.

Network Antivirus Concepts

If you must transfer executable files, and to a lesser extent data files, the ability to block infected files on the wire, is what Antivirus is all about. Using application engines and updated file virus signatures, antivirus provides a filtering service for file transfers. Network antivirus does not quarantine or disinfect the file; it blocks the file and notifies the user. Antivirus signatures are highly accurate; thus, they won't bury you with logs of false positives requiring complex technical investigation.

Antivirus primarily protects the users by inspecting the file downloads in the server-to-client flows.

The anti-virus feature can be divided into two main parts, the application protocol engine, found in the firmware, and the pattern/signature updates, updated from the network. The application protocol engine decodes the content file types, as well as the file transfer protocols.

Antivirus provides engines for the following file transfer application protocols:

- HTTP (downloads, server-to-client flows) ICAP
- FTP (downloads, server-to-client flows)
- POP3 (download, server-to-client flows)
- IMAP (downloads, server-to-client flows)
- SMTP (uploads, client-to-server flows) ICAP

These engines decode the file transfer application protocol, and scan for files. Finding a file, they then decode file formats, unpacking and decompressing as needed, checking for virus signatures. These engines cover the typical file transfer applications. There are ways to transfer files, particularly larger files, that are not scanned. Because the protocol engine must buffer the file, and this buffer must fit in memory, scanning is limited to small and medium-sized files. Content must be buffered and then reassembled before scanning, which might increase delay. This may be perceptible for interactive applications like HTTP.

Because viruses typically piggyback on executable files, one potential optimization is to not scan data content file types (Application/x-director, application/pdf, audio/*, image/*, text/css, test/html, video/*). Due to recent backdoors in Microsoft JPEG and BMP file formats, even data files may contain viruses, so this is not recommended.

Antivirus Planning

ScreenOS supports internal and external antivirus scanning on selected platforms (see Table 10.6). Internal antivirus is generally supported on the newer branch office platforms, while external antivirus can be supported through redirection across the entire product line.

Table 10.6 Antivirus Engines on Juniper Firewall Models*

	NS-5GT	NS-25/50	NS-204/208	NS-500	NS-5200/5400
Juniper-Kaspersky Embedded	Yes	No	No	No	No
ICAP Symantec Redirect	No	No	No	No	No
Policy-Based Routing Redirect	Yes	Yes	Yes	Yes	Yes
	SSG-5	**SSG-100**	**SSG-500**	**ISG-1000**	**ISG-2000**
Juniper-Kaspersky Embedded	Yes	Yes	Yes	No	No
ICAP Symantec Redirect	No	No	No	Yes	Yes
Policy-Based Routing Redirect	Yes	Yes	Yes	Yes	Yes

* Data is based on Juniper Networks firewalls running ScreenOS 5.4.

ScreenOS 5.0 introduced an antivirus engine from Trend Micro for the NS-5GT platform. ScreenOS 5.3 introduced an antivirus engine from Juniper-Kaspersky for the NS-5GT platform and new SSG platforms. The Juniper-Kaspersky signatures include signatures not just for viruses, but also for files used for spyware and phishing. Only one antivirus engine may be run at a time since the antivirus engine is part of the firmware.

The internal antivirus requires a device subscription service, upgraded memory on the SSG series, and needs the setup of name service and the correct time to update the signatures. Activation also requires taking an antivirus action in the policy. External redirection does not require device subscription since that is handled on the external antivirus server itself.

You should plan ahead and purchase the antivirus subscription, which you associate online with your registered device. The online service reserves an antivirus subscription, and your device only needs to connect and authenticate to the service to retrieve its antivirus subscription key.

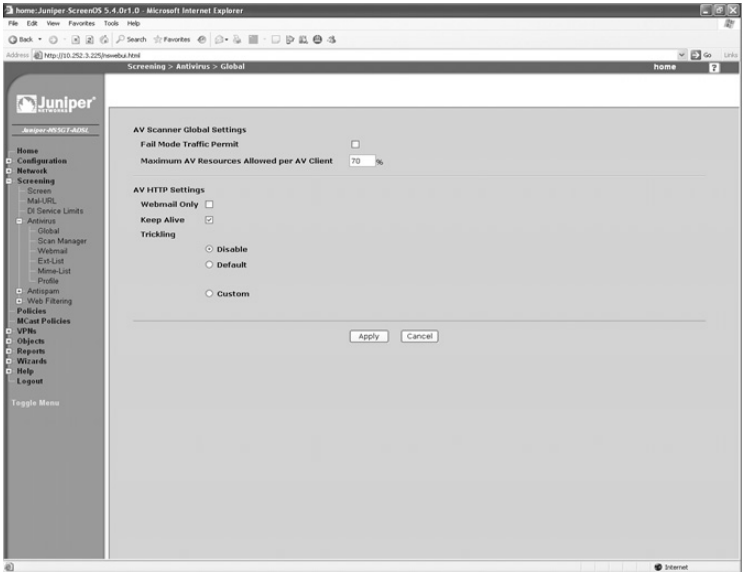
Configuring Global Antivirus Parameters

Access **Screening | Antivirus | Global** to display the ScreenOS 5.4.0 screen shown in Figure 10.13. The options to configure include the following:

- **Fail Mode Traffic Permit** If this option is enabled, and for some reason the AV scan fails (either because it's too large, reaches a compression recursion limit, or some other scan failure), the traffic is still permitted. If the scan is successful, and no virus is found, the traffic is permitted regardless of this setting. If the scan detects a virus, the traffic is dropped, regardless of this setting.

- **Maximum AV Resources Allowed per AV Client** This setting enforces fairness among multiple AV clients, by limiting the memory resource per AV client.
- **Keep Alive** If this option is enabled, the NetScreen device will keep the HTTP session open to the server with a keep-alive request after the file arrives on the NetScreen device, but before it has finished scanning the file. This decreases overall latency of the connection, but it is less secure.
- **Trickling** The method of sending a small portion of the file on to the requesting client so that the client's browser won't timeout the connection. The three options for this are Disable (which disables the trickling feature), Default (if the received file is larger than 3MB, it will trickle 500 bytes for every 1MB of data scanned), or Custom, in which you can set your own trickling settings:
  - **Minimum Length** This sets the minimum file size to start trickling. Files smaller than this will not be trickled at all. Files of this size or larger are trickled according to the following settings.
  - **Trickle Size** This sets the trickle packet size.
  - **Trickle for Every** This sets the amount of traffic sent before a single packet is sent.

Figure 10.13 ScreenOS 5.4.0 Global Antivirus Parameters



These settings determine how the device handles traffic it inspects, but how does the NetScreen get these attacks to match against? For these settings, we need to configure the Scan Manager.

Configuring Scan Manager Settings

An antivirus is only as good as its virus database, so to stay protected you need to stay updated. These settings are found by accessing **Screen** | **Antivirus** | **Scan Manager** in your WebUI. Figure 10.14 shows the 5.4.0 version of this screen. This shows important information such as AV license entitlement, as well as how current your virus definitions are. Other important features on this screen include the following:

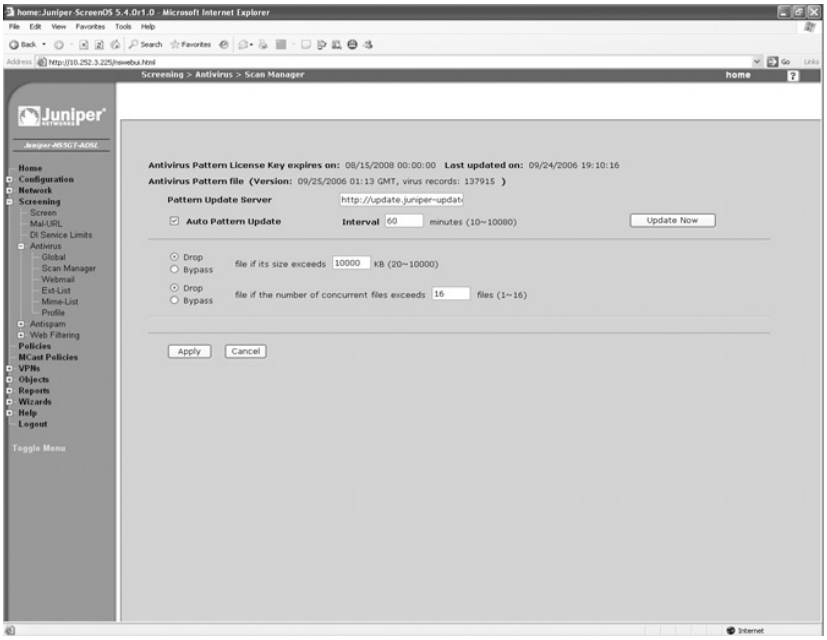
- **Pattern Update Server** This is the server the device will automatically connect to in order to obtain AV pattern updates from. The default is

  http://update.juniper-updates.net/AV/*[device-model]*/.

  (The [device-model] portion of the URL for my NetScreen-5GT was 5gt.)
- **Auto Pattern Update** This permits or disables the auto-update feature. Leave this option enabled unless you're experiencing problems obtaining the update.
- **Interval** This sets how often (in minutes) it checks for an update, with a default of 60. Don't be concerned if the Last Updated On: value doesn't coincide within the current time minus the update time. The date/time group at the top is the date/time of the AV pattern file, and not necessarily when it was loaded. Viruses spread most quickly during the first 24 hours (due to lack of antiviral protection), so be sure to leave this interval low.
- **Update Now** A handy button to check your settings and to manually refresh your AV definitions if you've turned off automatic updates.
- **Drop/Bypass** and **Size Exceeds** This sets the limit for the size of uncompressed files to scan, and also an action if this limit is reached.
- **Drop/Bypass** and **Number of Files Exceeds** This sets the limit for the number of uncompressed files to scan, and also an action if this limit is reached.

For the Webmail blocking option, the NetScreen has to have the URL path portion configured for relevant Webmail systems in order to determine if it has to check attachments for viruses. By default, popular Webmail sites are preconfigured—AOL, Yahoo, Hotmail, and others. The settings for these sites may change as the developers for these sites add new features or make other changes. Also, any other Webmail site you want filtered will have to be added manually and monitored for effectiveness due to changes later. I would recommend *not* using this feature unless the NetScreen device simply cannot pass HTTP traffic fast enough in All HTTP mode. Also keep in mind that if you do use this mode, you can set any URL path, not just Webmail, to check for viruses, such as popular file download sites like TuCows, Freshmeat, or FilePlanet.

Figure 10.14 ScreenOS 5.4.0 Scan Manager Antivirus Parameters



Configuring Antivirus Profile Settings

An antivirus profile is what is used in a policy. A profile contains the configuration options for application protocols that will be inspected for viruses. Protocol options include FTP, HTTP, POP3, IMAP, and SMTP.

- The ns-profile represents the Juniper-Kaspersky engine settings.
- The scan-mgr profile represents the default Trend Micro engine settings.

Profiles include options to disable protocols. Any protocol disabled in a profile will not be scanned, regardless of the configuration of the policy the AV profile is applied to. Disabling any of these will relieve your NetScreen of some processing burden, but will increase your likelihood of infection. Turn off these protocols *only* if you are certain they cannot be passed (for example, if you are using an explicit blocking policy).

Below that first section, there are a few configuration options for handling compressed files. Since a single compressed file can contain one or more files, and those files themselves can be compressed, in order to ensure that all content is checked, the NetScreen device will decompress Zip files if possible and examine the results for viruses. This can be time- and resource-consuming, so some practical limits have been introduced, but these are user-configurable. Note that compressed file nesting is a common AV evasion technique. Another AV evasion technique is sending a very small compressed file that expands into an extremely

large uncompressed file, attached with a second file that is malicious. The idea is that the AV scanner would max out on the beginning, but extremely large, file and then skip checking any other files after it, including the resulting malicious files. To prevent these kinds of evasion techniques, you can adjust the following settings:

- **File Decompression** The number of recursions the scanner will go down into, from 1 to 4.
- **Scan All**, **Scan Intelligent**, or **Scan By Extension** Scan for all signatures, only those common in the wild, or scan by file extension type. Scanning only for common signatures may increase performance and reduce false positives.
- **Skipmime Enable** If this option is enabled, the following MIME (Multipurpose Internet Mail Extension) types and subtypes are skipped from AV scan:
  - application/x-director
  - application/pdf
  - audio/*
  - image/*
  - text/css
  - text/html
  - video/*

  This improves AV scanning performance since the majority of HTTP traffic uses these MIME types. Until recently this was considered a safe setting, so bypassing is enabled by default. Thanks to recent Microsoft vulnerabilities with JPEG and BMP file formats, however, this is no longer the case. Unless performance is suffering considerably, do *NOT* enable this option.

Figure 10.15 shows the default antivirus profile for the Kaspersky engine.

Antivirus Rules

Activating AV couldn't be easier. Antivirus protection is activated by associating an Antivirus profile with each rule. Select the AV profile object from the drop-down list. Do this for each policy that needs antivirus scanning on any of the supported protocols. The final policy listing summary should look similar to Figure 10.16. Refer to Figure 10.17 for the policy editing window.

Figure 10.15 ScreenOS 5.4.0 Antivirus Profile Parameters

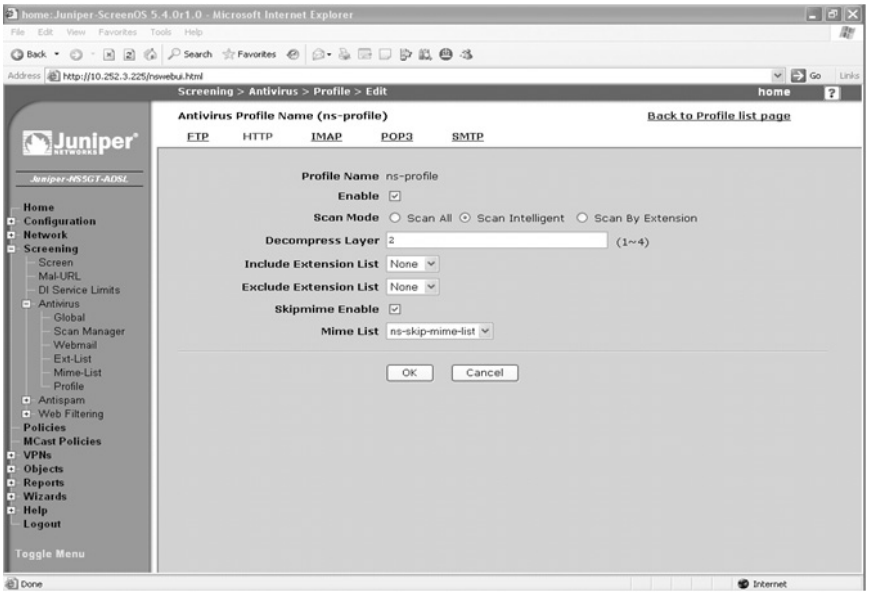
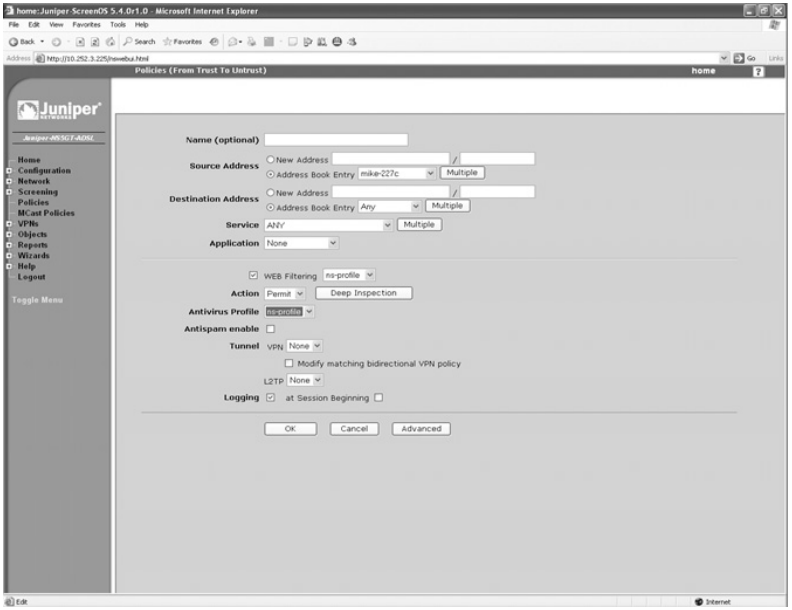


Figure 10.16 Antivirus Policy Listing with Icon

mike-227c	Any	ANY			

Figure 10.17 Antivirus Policy Configuration



Verify Antivirus Protection

How do you test whether your Antivirus protection is configured and operating correctly? For obvious reasons, Antivirus researchers don't redistribute potent malicious viruses. But they have created a harmless standard pattern that every Antivirus product can include in its signature database, and test files that contain that pattern, to exercise the Antivirus engines and pattern matching. This Antivirus test pattern file, in various packages and archives, was created and can be found at the European Institute for Computer Antivirus Research (EICAR). EICAR does more than just Antivirus research these days. (To get the EICAR test files, see www.eicar.com/.)

If your Antivirus is operating correctly, in your Web browser you should see the following:

> VIRUS WARNING 10.252.3.237:1273->83.246.65.3:80 Download contaminated file: www.eicar.com/download/eicar.com/ with virus EICAR-Test-File.

And your logs should contain the entry:

> AV: VIRUS FOUND: 10.252.3.237:1283->83.246.65.3:80, http url: http://www.eicar.com/download/eicar.com, file www.eicar.com/download/eicar.com/ virus EICAR-Test-File.

You can test whether the test pattern is correctly blocked for executables (.com), data files (.txt), and archives (.zip) depending on your configuration.

Understanding Application Layer Gateways

Application Layer Gateways are algorithms within ScreenOS that handle dynamic firewall policies that certain protocols require, such as FTP. Many such protocols were designed without security or other access controls in mind, which can cause problems when firewalls are introduced.

For example, FTP uses multiple sessions to facilitate file transfers—a primary command channel, and secondary data channels for directory listings and file transfers. Often, these data channels will flow in a direction opposite that of the original command channel. Since these data channels could connect on any port, it's almost impossible to create a static firewall policy that would permit these data channels and yet still provide adequate protection.

The FTP ALG automatically solves this problem by monitoring the FTP command channel, looking for FTP *port* commands that specify which source and destination ports are being requested, and dynamically opening up that specific combination of source IP/port and destination IP/port firewall policy (called a *gate*) that permits the session to flow. Once the session is complete, the gate is immediately closed.

The FTP ALG also handles the special case where the FTP session flows through a NAT interface. In this circumstance, the endpoints don't always realize their addresses are being translated midstream. The FTP port commands use whatever IP the endpoint hosts' interfaces are configured for, which, in the case of a host behind a NAT firewall, will typically be unreachable from the Internet.

The ALG handles this at the application layer by modifying the ASCII port command in-situ, replacing the inside IP with the IP of the NAT interface. Since port commands are passed as ASCII text, including the IP address, the chances are high that the number of characters that represent the inside IP and the external IP won't exactly match (for example, an inside address of 192.168.1.5 contains 11 characters, which may be translated to something like 123.123.123.123 at 15 characters, or something like 1.2.3.4, which contains only 7). The firewall cannot inject these extra bytes of data without modifying the TCP checksum as well as the TCP sequence numbers. It achieves this by essentially proxying the connection at the TCP layer. This is similar to the SYN proxy feature used by the TCP flood SCREEN setting.

NetScreen ALGs are different from many competitors' products. Several other firewall vendors utilize full protocol proxies, which themselves are vulnerable to attack, misconfiguration, or protocol obsolescence as new commands, options, and features are added to a protocol. CheckPoint's FireWall1 uses tiny proxies to validate data on protocols like HTTP, FTP, and SMTP. While this method is very flexible, it can still cause problems if the proxy encounters a valid command it has not been programmed to handle. This can cause the session to break since the proxy won't forward what it thinks is an invalid command. Furthermore, since the firewall is participating in the stream at the application layer, it's very possible (and has even happened) that the proxy itself is vulnerable to a security concern. Since FireWall1 runs on Windows, Linux, and Solaris, shellcode for these platforms is relatively easy to find. NetScreen firewalls do not participate in the exchange at the application layer, which isolates them from these sorts of attacks.

Some protocols just don't support being proxied. Microsoft's Server Message Block and Remote Procedure Call both require a real endpoint connection. While these are not commonly Internet-transiting protocols, a good defense-in-depth strategy would still have this traffic flowing through firewalls that need to know how to handle it. A new ALG found in ScreenOS 5.1 allows users to filter at the application layer for MS-RPC by parsing g*lobally unique identifiers* (*GUIDs*)—a unique 128-bit number used by Microsoft to label process endpoints. Custom-defined services are created based upon GUIDs, which are then used in a policy. This enables you to create policies that allow or prevent access to individual processes on a Windows system. This is very handy for protecting from attacks such as Blaster, Sasser, Agobot, and others that use MS-RPC as one of their attack vectors.

Others vendors tend to cut corners and, for the sake of performance, will make a very simple ALG-like algorithm that should solve a problem, but has unexpected consequences. Just recently, Symantec issued a security update for its DNS ALG. Apparently, the DNS ALG worked like so: if a UDP packet arrived with a source port of 53, it was a DNS reply to a DNS request that had already gone out through the firewall, and would permit the packet through without any session lookup. The ALG would also bypass any incoming policy

explicitly blocking the packet, such as destination port, destination IP, or source IP. The flaw in the firewall was so fundamental it would even bypass protections designed for its own management interface. When this oversight was made public, hackers discovered that by sending management packets *to* the Simple Network Management Protocol (SNMP) port on the firewall *from* UDP port 53 they could successfully command the firewall and change its settings without being authenticated. A patch was later released. ScreenOS features are subject to rigorous security reviews at various stages of the development process to avoid fundamental logic flaws such as this.

ScreenOS currently has 26 ALGs, including FTP, DNS, and H.323, with more being released with every new version. These ALGs require little to no configuration to operate properly. They automatically detect appropriate traffic on the registered ports for the protocol they handle and then do their jobs. As mentioned earlier, these ALGs can be reapplied to arbitrary ports using custom service objects as needed.

Applying Best Practices

NetScreen firewalls have a wealth of security features they can use, but even the best tool can be rendered ineffective through poor implementation. This section hopes to instill some good security practices to use with your NetScreen device.

Defense-in-Depth

How many locks do you have on your front door? Just one? Or do you have one lock for the doorknob, another for the deadbolt, and a chain? Do you have an alarm system as well? How about a bat by the bed? If you have all of this, then you already understand what *defense-in-depth* means. Network security is no different. Having a NetScreen firewall protecting your network is a good start to an overall effective network security system. However, it is the components of the whole system working together—internal firewalls, perimeter firewalls, IDPs, authentication services, management, antivirus software, and monitoring services—that make you more secure. The National Security Agency (NSA) has recently released a very informative paper on this subject, located at www.nsa.gov/snac/support/WORMPAPER.pdf.

Zone Isolation

An extension of the defense-in-depth concept, zone isolation, involves placing different system types (for example, servers, end users, Engineering, Finance, Information Technology, and so on) on different zones, which then allow for firewall policies between these dissimilar groups. Do your end users need access to the Finance department's systems? If so, what kind? Find out how to limit access to just what is necessary. Don't just assume that because a computer is inside your perimeter it's safe. Keep access to specific areas, zones, and data to the smallest number of computers necessary.

Egress Filtering

Egress filtering is the process of putting restrictions on outgoing traffic as well as incoming traffic. Many locations only get half of the security picture straight—they block traffic from coming in except to specific Internet-facing servers (mail, Web, DNS, and so on), but let all inside traffic go back out completely unfiltered. Ideally, your outgoing policies should be as complex and stringent as your incoming policies. Or better yet, *no* traffic initiated by your end users should be allowed out to the Internet. Instead, all traffic should go through approved and configured proxy servers (such as HTTP or FTP; I very much recommend the highly configurable open-source proxy called *Squid* that comes with almost every distribution of Linux out there) or internal-only servers (such as mail or DNS). This locks down infections since most backdoors don't support proxies, and those that do can be detected and blocked at your proxy.

Explicit Permits, Implicit Denies

The idiom of "You don't know what you don't know" is never truer than in the security business. Firewall administrators who block specific ports and let all others through are asking for trouble through their ignorance. Why Windows XP listens on 10 different ports to perform the same function, I'll never know. If there's just one port I miss, a worm or other malicious attack could slip by and tear up my network from the inside. Instead, permit what you *know* you want permitted, and block everything else—this keeps things simple and the threats *known*.

Retain Monitoring Data

If something does happen, and it usually does, you need to know the breadth and width of the problem, when it started, and how it happened so you can properly clean it all up and keep it from happening again. You're going to have enough trouble as it is from hackers hiding their activities through evasion and log file deletion on compromised systems. Don't compound the problem by not having dedicated secure machines for logging and keeping the data for a historically significant period of time. This may be the only way to track a Black Hat attacker who spaces his attacks out over hours or days. It doesn't hurt to *look* at the logs from time to time as well to ensure things are copasetic.

Keeping Systems Updated

If there's one thing 2004 has taught network security professionals, it's that automatic operating system updates are a good thing. Windows, MacOS, and many flavors of Linux now support some sort of automatic patching system to react to newly discovered security issues. *Use* these tools to your advantage, and keep your systems patched. Check in with Windows Update the first Tuesday of the month to see what new vulnerabilities Microsoft has 'fessed up to, and get to patching!

Summary

This chapter has covered a lot of ground in a short amount of space. Indeed, complete books are available just on some of the subjects covered here. Despite this, we've managed to cover all SCREEN, Deep Inspection, URL filtering, and antivirus features, as well as Application Layer Gateway functions enough to give you a good understanding of the capabilities of each, and how to set them up.

SCREEN features are the legacy security protection, covering things like SYN, UDP, and ICMP floods (with user-configurable thresholds), session-exhaustion attacks (with separate thresholds for source and destination), classic IP and TCP header manipulation, port scans and sweeps, and certain OS-specific DoS attacks. Enabling these attacks imposes almost no performance hit.

Deep Inspection takes security all the way to the application layer for selected protocols. Using stateful contexts to accurately isolate malicious traffic, you can minimize false positives and false negatives and maximize valid hits. Protocols covered in ScreenOS 5.0 included HTTP, FTP, DNS, SMTP, IMAP, and POP3, with a very limited set of contexts exposed to end users for use in their own signatures. ScreenOS 5.1 introduced several new protocols including SMB, MS-RPC, NetBIOS, Gnutella, AIM, and YMSG and included a significant increase in the number of contexts supported for end user's signatures.

We covered how ScreenOS uses contexts to break down a protocol stream into inspectable fields and uses a DFA to inspect traffic, and how to use customer-available contexts and the DFA RegEx syntax to write our own custom signatures.

Also covered was ScreenOS's support for HTTP URL filtering with a variety of URL filtering options from WebSense and SurfControl, including a method of storing filtering profiles right on the device with SurfControl Integrated Mode. The Integrated Mode requires a license key.

Antivirus is an important new feature in ScreenOS 5.0 that allows HTTP, FTP, SMTP, POP3. and IMAP protocols to be inspected for viruses. While there are a variety of settings to configure, the majority of the defaults work well, and with a few clicks AV can be up and running almost effortlessly. Antivirus also requires a license key.

Application Layer Gateways are a powerful mechanism that allow certain security-impaired protocols to work through a firewall. FTP, H.323, and dynamic-channel protocols could play havoc with a firewall policy if it wasn't for ALGs and the gate feature that opens dynamic firewall policies automatically to allow data channels of already permitted command channels of these troublesome protocols.

Finally, we covered some security basics—how to put this all together into a more secure whole. Defense-in-depth, zone isolation, egress filtering, implicit permits, explicit denies, logging, and keeping systems up-to-date aren't just topics in a CISSP book—they're real-world solutions for making your network more secure. Applying even just a few of these concepts will go a long way toward making the Internet a safer place.

Solutions Fast Track

Understanding Attacks

- ☑ NetScreen firewall products pack a diverse range of features into a small, easy-to-use system.
- ☑ A firewall's primary function is always security.
- ☑ NetScreen firewall products have a variety of different security methods to stop many different types of attack.
- ☑ SCREEN features generally cover IP and TCP layer attacks.
- ☑ Deep inspection covers advanced IP, TCP/UDP, and application layer attacks.
- ☑ Content filtering enforces local usage policy.
- ☑ Antivirus keeps mass-mailing worms and other malware out of your network.
- ☑ Application Later Gateways handle problem protocols securely.

Understanding the Anatomy of an Attack

- ☑ Generally, there are three phases of an attack: reconnaissance, exploit, and consolidation.
- ☑ Different kinds of attackers require different methods for protection.
- ☑ Script Kiddies want publicity for their activities and will make their presence well known.
- ☑ Black Hat hackers are more insidious and patient; it takes diligence and consistency to detect them.
- ☑ Self-propagating worms are becoming more aggressive and complex as time goes by. They will attack like a Kiddy but exploit like a Black Hat.
- ☑ Defense-in-depth, updated systems and signatures, event log vigilance, good social engineering awareness training, and a sound security policy can keep these attacks at bay.

Configuring SCREEN Settings

- ☑ NetScreen legacy screen settings protect networks through a variety of detection methods.

- ☑ Network reconnaissance uses port scans, sweeps, and protocol option manipulation to avoid detection.
- ☑ Port scans and sweeps are detected, and can be tweaked with user-configurable threshold settings.
- ☑ Several TCP flag manipulation techniques are detected, including null scan, SYN/FIN scan, and others.
- ☑ A variety of IP protocol option manipulation techniques are also detected, such as source-route, time-stamp, and others.
- ☑ SYN, UDP, and ICMP flood attacks are prevented with user-definable thresholds to maximize throughput while still providing protection.
- ☑ Rudimentary HTTP content filtering is supported, with the optional blocking of ActiveX and JavaScripts, as well as EXE and ZIP file downloads.
- ☑ Additional protocol attacks—such as WinNuke, Land, Ping of Death, and others—are also covered.

Applying Deep Inspection

- ☑ Deep Inspection covers application layer attacks on selected protocols.
- ☑ With ScreenOS 5.0, six protocols are covered: HTTP, FTP, DNS, SMTP, POP3, and IMAP.
- ☑ Customer-exposed contexts were also extremely limited in ScreenOS 5.0.
- ☑ ScreenOS 5.1 introduced several new protocols with new contexts: SMB, MS-RPC, NetBIOS, Gnutella, AIM, and YMSG
- ☑ ScreenOS 5.1 also added new contexts for ScreenOS 5.0–supported protocols for end users to write signatures with.
- ☑ Signatures use a custom subset of regular expressions and a DFA string-matching algorithm to detect attacks.

Setting Up Content Filtering

- ☑ NetScreen firewall products support both URL filtering and, more recently, antivirus filtering.
- ☑ Starting with ScreenOS 5.1, NetScreen now also supports SurfControl as well as the legacy WebSense URL filtering system.
- ☑ On newer low to midrange NetScreens, SurfControl can also be used in Integrated Mode right on the device.

- ☑ Starting with ScreenOS 5.1, NetScreen firewalls support off-loading as well as onboard antivirus inspection.
- ☑ Onboard antivirus inspection uses the Trend Micro engine.

Understanding Application Layer Gateways

- ☑ Many commonly used protocols, such as FTP, MS-RPC, or H.323, were never designed to be firewalled.
- ☑ ALGs enable these security-impaired protocols to work through a firewall by parsing the command channel.
- ☑ ALGs are *not* protocol proxies, which are limited to certain protocols and are subject to change.
- ☑ Other competitors' over-simplified ALGs can contain logic errors leading to security breaches.

Applying Best Practices

- ☑ Defense-in-depth distributes security across your infrastructure to prevent single-point-of-failure compromises.
- ☑ Zone isolation increases the granularity of control over devices in your network.
- ☑ Egress filtering ensures protection against unknown activity leaving your network.
- ☑ Explicitly permitting desired traffic and blocking everything else ensures you know what's allowed where.
- ☑ Keeping logging systems secure and monitored will help you isolate problems and keep security events from recurring.
- ☑ Most common operating systems now support automatic updates—*use them*!

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: What's a "Grey Hat" hacker?

A: A Grey Hat is someone who knows both sides of the security coin (Black Hat and White Hat) and will dabble with both. Like many things in life, "good" and "bad" distinctions are not so binary in the real world. Is it okay to use a malicious attack against an offensive site (hate crime, child porn, spam relay, and so on)? Is it okay to attack a host that attacked you first? Computer ethics classes are now a regular component of most security certification courses.

Q: What's a good setting for the (SYN|UDP|ICMP) flood threshold?

A: There's no universal setting that could be applied effectively to every network. Network activity, purpose, and traffic are different for each segment, and they need to be tweaked appropriately. Ideally, you'd want a setting that's just above dropping normal volumes of legitimate flows, so when an actual flood occurs, the NetScreen can react immediately.

Q: Why doesn't DI have the same coverage as an IDP?

A: NetScreen firewall devices have purpose-built Application-Specific Integrated Circuits (ASICs) that handle the majority of firewall operations, such as policy matching and data encryption. These are physical devices—hardware accelerators—that can't be modified. Since DI is a relatively new feature, it has to run in software on the system's CPU. Older NetScreen firewall CPUs were generously sufficient for device management, but were significantly slower than the ~3 GHz single and dual CPUs found on an IDP. This difference in computing power means that a full IDP would be impractical on the currently available platforms. Juniper is actively investigating a new ASIC that incorporates IDP functionality in hardware, and also ensures that new generations of NetScreen firewalls have beefier CPUs.

Q: Why is SurfControl Integrated Mode only available on lower-end products?

A: Many users of these more inexpensive products don't have permanent or full-time onsite IT support personnel, and may not have the time, money, or expertise to configure, run, and maintain their own WebSense or SurfControl URL filtering software. Yet these customers still want (and need) URL filtering. The Integrated Mode utilizes SurfControl's

public servers that SurfControl maintains and supports. Also at risk is the performance impact (including latency) of sending large volumes of URL look-up requests over the Internet to a public server. The high-end products support speeds well over 10 Gb/s of sustained traffic. To look up every URL requested in real-time would waste significant bandwidth.

Q: Why should I bother with egress filtering? It's a lot of work, and my users are bound to complain about something not working.

A: The initial effort of configuring egress filtering now will save you several orders of magnitude's worth of work later. The majority of business-related software supports proxying or other filtering. Chances are, complaints from some end users regarding connectivity problems generally arise from software you don't want running on your network anyway.

Q: Why are there so many different license keys for features?

A: Juniper, like any for-profit company, wants to make money. It also understands that it needs to be competitive in the market. If Juniper sold its devices at a high price with all features enabled (many of which you may or may not use), it would have a difficult time selling them to customers who only needed some of the features and were willing to buy them at a reduced price. This way, a compromise is reached—they will sell you a useful product for a reasonable price, but the flexibility is there for additional features (which you can buy a la carte). Time-limited license keys also facilitate subscriptions.

Chapter 11

VPN Theory and Usage

Solutions in this chapter:

- **Understanding IPSec**
- **IPSec Tunnel Negotiations**
- **Public Key Cryptography**
- **How to Use VPNs in NetScreen Appliances**
- **Advanced VPN Configurations**

- ☑ **Summary**
- ☑ **Solutions Fast Track**
- ☑ **Frequently Asked Questions**

Introduction

As you progress through this chapter, you will understand the concepts of virtual private networks (VPNs), how VPNs operate, and how to implement VPN tunnels using Internet Protocol Security (IPSec) on NetScreen appliances. A VPN is a means of creating secure communications over a public or insecure network infrastructure. VPNs use encryption and authentication to ensure that information is kept private and confidential. This means that you can share data and other network resources among two or more locations without the worry of the data or its integrity being compromised.

The primary advantage of a VPN is its capability to make use of an insecure public network infrastructure to create a private and secure network. The most common method of deployment is over the public Internet. Today fast, cheap, and fairly reliable Internet access is available almost anywhere. Organizations are widely replacing legacy frame relay and point-to-point networks with VPNs deployed over the public Internet. This provides several advantages when compared to legacy networks including cost savings, increased bandwidth, and additional redundancy. Without using the Internet as a transport mechanism, you would have to purchase point-to-point circuits or some other form of leased line to connect multiple locations. Leased lines are traditionally expensive to operate, especially if the two points being connected are across a large geographic region. Using VPNs instead of leased lines reduces the operating cost for your company. VPNs enable connectivity where leased line access just isn't possible or cost justified. Companies today need global connectivity. It might not be feasible for a US-based company to purchase a leased line to a new office in China, but as long as local Internet access is available, the same level of connectivity and private access is still achievable.

VPNs are also a necessity for traveling and telecommuters. In today's fast-paced, always-connected work environments, VPNs, once considered optional, and are now a requirement. The days of dialing into modem banks are gone and even if dial-up access were still available, the speed of a modem is not sufficient for productive remote access anymore. Employees are working from home, organizations are hiring remote employees who might not ever report to an office, and even vendors supporting or consulting for an organization do so remotely.

Understanding IPSec

IPSec is a framework consisting of several protocols for securing communications at the Internet Protocol (IP) layer. Though there are several types of VPNs, IPSec is the most common deployment method today and almost always used in a site-to-site VPN tunneling between two local area networks (LANs). It is also commonly used in a dial-up VPN scenario with host-based software being used to connect to a remote IPSec gateway. IPSec was engineered to provide several services:

Privacy and Confidentiality Addresses the requirement that the data is kept secret in the event that there is a man in the middle intercepting communications; if someone were to capture the data it could not be understood or decrypted.

Integrity and Origin Authentication Even though the data is encrypted, it's possible that it could be modified. Authentication makes sure that both ends are who they say they are and that the data hasn't been modified.

Protection Against Replay Attacks This ensures that communications cannot be captured by a third party and retransmitted at a later time.

As you can guess, IPSec protocols are abundant with two- and three-letter acronyms. IPSec consists of two modes: transport and tunnel. IPSec also consists of two protocols: encapsulating security payload (ESP) and authentication header (AH). IPSec allows for manual or automatic negotiation of security associations (SAs). All of this information makes up the domain of interpretation (DOI) for IPSec, which is used to establish SAs and Internet key exchange (IKE).

IPSec Modes

As mentioned earlier, the IPSec protocol provides us with two modes of operation: *transport mode* and *tunnel mode*. Each of these modes provides us with similar end results, but work differently to achieve the same goal. For starters, transport mode requires that both endpoints of the VPN tunnel be hosts. *Tunnel mode* must always be used when one endpoint is a security gateway, such as a NetScreen appliance or router. NetScreen appliances always provide IPSec tunnels in tunnel mode.

Transport mode only encrypts the *payload*, or data portion, of the IP packet. The header of the packet is not encrypted or altered. Think of it as a sealed envelope. You are able to see the address of who the letter is to and where it is from, but you cannot view the envelope contents. Transport and tunnel mode packets are illustrated in Figures 11.1 and 11.2.

In tunnel mode, the original packet, both header and payload, is encapsulated entirely into another IP packet. This new packet has its own header, containing source and destination address information. These addresses are the actual endpoints of the tunnel. Although both modes encrypt the actual payload, tunnel mode is generally thought of to be more secure than transport mode.

Protocols

As we previously mentioned, IPSec has two methods for verifying the source of an IP packet as well as verifying the integrity of the payload contained within—authentication header (AH) and encapsulating security payload (ESP).

Figure 11.1 Transport Mode Packet

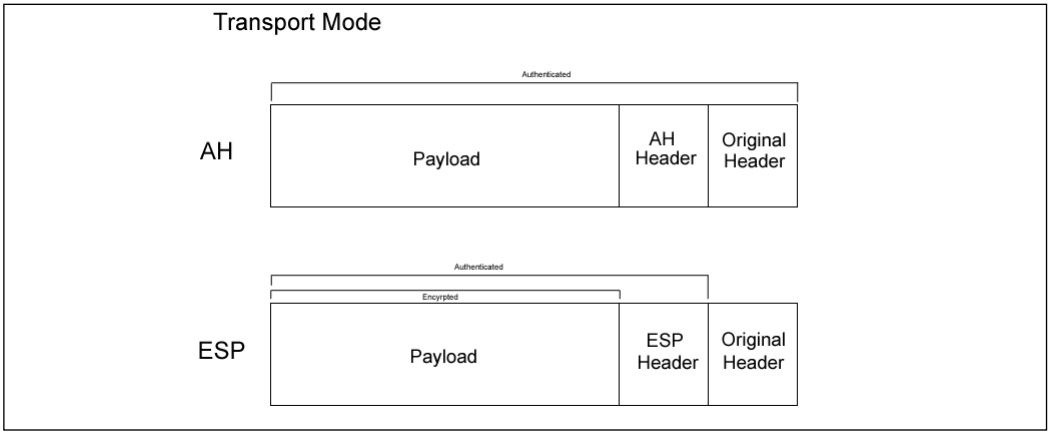
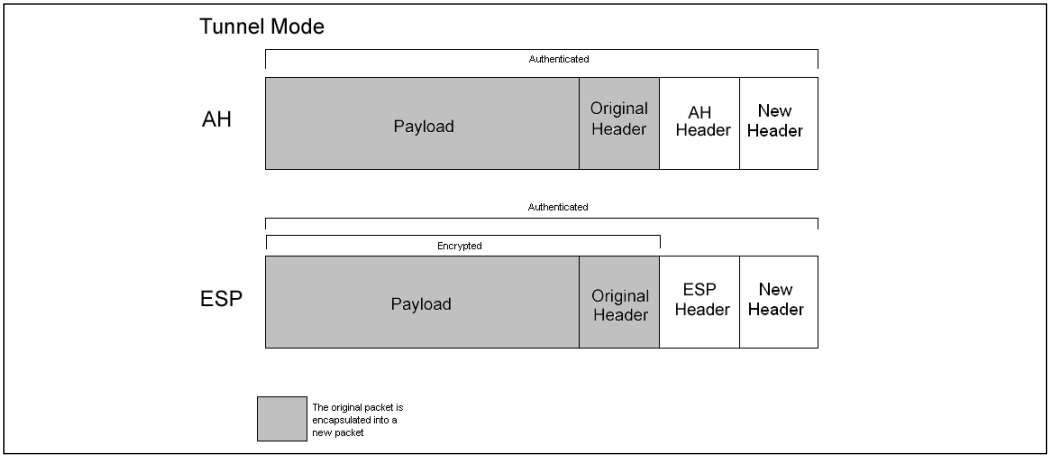


Figure 11.2 Tunnel Mode Packet



Authentication header, or AH for short, provides a means to verify the source of an IP packet. It is also used to verify data integrity of the payload the packet contains. When used in transport mode, AH authenticates the IP packet's payload and portions of the IP header. When AH is used in tunnel mode, the entire internal IP header is authenticated as well as selected portions of the external IP header. AH can also protect against replay attempts. AH can be used by itself, or it can be used in conjunction with encapsulating security payload.

Encapsulating security payload, or ESP, provides methods to ensure data privacy, source authentication, and payload integrity. ESP may also protect against replay attacks. ESP, when used in tunnel mode, encrypts the entire IP packet and attaches a new IP header to the packet. The new IP header contains all the information necessary to route your packet to its destination. ESP also enables you to choose what to do with the packet: encrypt the packet,

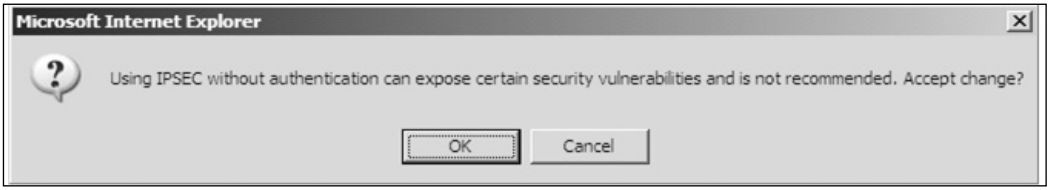
authenticate the packet, or both. ESP, with transport mode, encrypts the IP payload, but not the IP header. Optionally, with transport mode, ESP can also authenticate the IP payload. When using ESP with tunnel mode, both the IP header and payload are encrypted. Like transport mode, ESP also optionally enables authentication of the IP packet.

Key Management

Probably the most critical part of a VPN is key management and distribution. IPSec supports the use of both manual and automatic key distribution.

In manual key configurations, all security parameters are configured at both ends of the tunnel manually. Although this method works well in smaller networks, there are some issues with using manual keys. This can be especially troublesome when the key is initially distributed, since there may be no way to verify the key was not compromised before reaching its final destination. This also becomes cumbersome when you choose to change the key, which results in a need for redistribution. When using a manual key VPN, the key is never changed unless the administrator chooses to change it. Configuring a manual key on a NetScreen device now will provide a warning message reminding you that it is a serious security risk (see Figure 11.3).

Figure 11.3 Warning about Configuring a Manual Key on a NetScreen Device



To help lessen the burden on administrators, the IPSec protocol supports Internet Key Exchange (IKE). IKE generates and negotiates keys and SAs automatically based on preshared secrets or digital certificates. A preshared secret is nothing more than a key both parties have prior to initiating the negotiations. Like manual key VPN, the preshared secret must be exchanged securely before use. However, unlike manual key VPNs, IKE can change the key automatically at a specified interval. This is seen as a significant security enhancement over that of manual key VPN. We will discuss the use of preshared secrets later in this chapter.

As previously mentioned, IKE can also use digital certificates. During IKE negotiation, both sides generate public and private key pairs with the use of digital certificates. If the issuing certificate authority (CA) is trusted by both parties, the participants can verify their peer's signature by retrieving the peer's public key.

There are also several other advantages of using IKE instead of a manual-key VPN. IKE eliminates the need to manually specify the IPSec security parameters at both peers, reducing the management load on the administrator. IKE also enables the use of anti-replay services,

certification authorities, and dynamic peer authentication in IPSec VPNs, which are discussed in more detail later in this chapter.

Security Associations

Security associations (SAs) are the concept used by IPSec to manage all the parameters required to establish a VPN tunnel. In simple terms, SA is a set of parameters describing how communications are to be secured. SAs contain the following components: security keys and algorithms, mode of operation (transport or tunnel), key management method (IKE or manual key), and lifetime of the SA. IPSec stores all active SAs in a database called the security association database (SAD). The SAD contains all parameters needed for IPSec operation, including the keys currently in use. In order to have bidirectional communication, you must have at least two SAs, one for each direction of traffic flow.

Tools & Traps…

Configuration & Troubleshooting Advice

To view the active and inactive SAs on a NetScreen device, in the WebUI navigate to VPN | Monitor Status or from the command line use the command *get sa*. When viewing the SA status, pay attention to the lifetime values (indicating when the SA will expire/renegotiate), the state of the SA (indicated as A/A or A/-, etc.). The first letter in the state indicates the status of the SA itself. The second letter in the state (or after the /) indicates the VPN monitor state (if configured). A stands for active, I stands for Inactive, and for the monitor state a "-" indicates it's not configured. The NetScreen internal event log will also provide great detail as to why an SA might be failing or having problems.

For advanced troubleshooting, view fragmentation and other errors by using the command *get sa stat* from the command line. You typically want to avoid fragmentation on the SA. If you have a tunnel that is showing high fragmentation, pay attention to the configuration parameters under *get flow* and configure them using *set flow*.

IPSec Tunnel Negotiations

When using a manual key VPN for communications, negotiations are not required between the two endpoints of the VPN tunnel. This is because all of the necessary SA parameters were defined during the creation of the manual key tunnel. When traffic matches a policy

using a manual key VPN, traffic is encrypted, authenticated, and then routed to the destination gateway.

An IPSec tunnel using IKE negotiation takes place in two phases. Phase 1 of IKE negotiation establishes a secure tunnel for negotiation of the SAs. Then, during Phase 2, IPSec SAs are negotiated defining the method for encrypting and authenticating the actual data exchange. The next section explains what happens in each phase of negotiation in detail.

Phase 1

From our previous discussion you already know that Phase 1 negotiations consist of exchanging proposals on how to authenticate and secure the communications channel. Phase 1 exchanges can be done in two modes: main mode or aggressive mode.

NOTE

Phase 1 on a NetScreen is configured in the WebUI under VPN | AutoKey Advanced | Gateway or on the command line using *set ike gateway*.

In main mode, three two-way exchanges, or six total messages, are exchanged. During a main mode conversation, the following is accomplished:

- **First exchange** Encryption and authentication algorithms for communications are proposed and accepted.
- **Second exchange** A Diffie-Hellman exchange is done. Each party exchanges a randomly generated number, or nonce.
- **Third exchange** Identities of each party are exchanged and verified.

NOTE

In the third exchange, identities are not passed in the clear. The identities are protected by the encryption algorithm agreed upon in the exchange of the first two sets of messages.

In aggressive mode, the same principle objectives are completed, but are done so in a much shorter conversation. Phase 1 negotiations in aggressive mode require that only two exchanges be made and that a total of three messages are exchanged. An aggressive mode conversation follows the following pattern:

- **First message** The initiating party proposes the SA, starts a Diffie-Hellman exchange, and sends its nonce and IKE identity to the intended recipient.
- **Second message** During the second message, the recipient accepts the proposed SA, authenticates the initiating party, sends its generated nonce, IKE identity, and its certificate if certificates are being used.
- **Third message** During the third message, the initiator authenticates the recipient, confirms the exchange, and if using certificates, sends its certificate.

In an aggressive mode exchange, the identities of communicating parties are not protected. This is because the identities are sent during the first two messages exchanged prior to the tunnel being secured. It is also important to note that a dial-up VPN user must use aggressive mode to establish an IKE tunnel.

Are You Owned?

What is Diffie-Hellman?

The Diffie-Hellman (DH) key exchange protocol, invented in 1976 by Whitfield Diffie and Martin Hellman, is a protocol enabling two parties to generate shared secrets and exchange communications over an insecure medium without having any prior shared secrets. The DH protocol consists of five groups of varying-strength modulus. Most VPN gateways support DH Groups 1 and 2. NetScreen appliances, however, support groups 1, 2, and 5. The DH protocol alone is susceptible to man-in-the-middle attacks, however. Although the risk of an attack is low, it is recommended that you enable Perfect Forward Secrecy (PFS) as added security when defining VPN tunnels on your NetScreen appliance. For more information on the DH protocol, see www.rsasecurity.com/rsalabs/node.asp?id=2248 and RFC 2631 at **ftp://ftp.rfc-editor.org/in-notes/rfc2631.txt**.

The use of DH Group 1 is strongly discouraged and never used in the field today because it is cryptographically weak. DH Group 2 is almost always used today for IPSec.

Phase 2

When Phase 1 negotiations have been completed and a secure tunnel has been established, Phase 2 negotiations begin. During Phase 2, negotiation of SAs of how to secure the data being transmitted across the tunnel is completed.

NOTE

Phase 2 on a NetScreen is configured in the WebUI under VPN | AutoIKE or on the command line using *set vpn*.

Phase 2 negotiations always involve the exchange of three messages. Phase 2 proposals include encryption and authentication algorithms, as well as a security protocol. The security protocol can either be ESP or AH. Phase 2 proposals can also specify whether or not to use PFS and a DH group to employ. PFS is a method used to derive keys that have no relation to any previous keys. Without PFS, Phase 2 keys are generally derived from the Phase 1 SKEYID_d key. If an attacker were to acquire the SKEYID_d key, all keys derived from this key could be compromised. During Phase 2, each side also offers its proxy ID. Proxy IDs are simply the local IP, the remote IP, and the service. Both proxy IDs must match. For example, if 1.1.1.1 and 2.2.2.2 are using the SMTP (Simple Mail Transfer Protocol) service, then the proxy ID for 1.1.1.1 would be 1.1.1.1-2.2.2.2-25 and for 2.2.2.2 it would be 2.2.2.2-1.1.1.1-25.

Are You Owned?

Key Lifetime—Short vs Long and PFS

When planning your VPN deployment, consideration should be given to the key lifetime and perfect forward secrecy in relation to security. Since enabling PFS requires additional processing time and resources, some administrators choose not to use it, instead opting for a shorter key lifetime. This, however, can be a bad practice. If a successful man-in-the-middle attack were able to discover the SKEYID_d key, all keys derived from this key could be compromised. Enabling PFS, even with a longer key life, is actually a more secure practice than having a short key life with no PFS.

Also make sure key lifetimes are in sync when creating VPNs between disparate gateways (that is, between a NetScreen and another manufacturer). Different manufacturers have different default lifetime values.

Public Key Cryptography

Public key cryptography, first born in the 1970s, is the modern cryptographic method of communicating securely without having a previously agreed upon secret key. Public key

cryptography typically uses a pair of keys to secure communications—a private key that is kept secret and a public key that can be widely distributed. You should not be able to find one key of a pair simply by having the other. Public key cryptography is a form of asymmetric-key cryptography, because not all parties hold the same key. Some examples of public key cryptography algorithms include RSA, DH, and ElGamal.

So how does public key encryption work? Suppose John would like to exchange a message securely with Chris. Prior to doing so, Chris would provide John with his public key. John would then take the message he wishes to share with Chris and encrypt the message using Chris' public key. When Chris receives the message, he takes his private key and decrypts the message. Chris is then able to read the message John had intended to share with him. But what if someone intercepts the message and has possession of Chris' public key? Absolutely nothing happens. When messages are encrypted using Chris's public key, they can only be decrypted using the private key associated with that public key.

PKI

PKI is the meshing of encryption technologies, services, and software together to form a solution that enables businesses to secure their communications over the Internet. PKI involves the integration of digital certificates, CAs, and public key cryptography. PKI offers several enhancements to the security of your enterprise.

PKI gives you the capability to easily verify and authenticate the identity of a person or organization. By using digital certificates, it is easy to verify the identity of parties involved in a transaction. The ease of verification of identity is also beneficial to access control. Digital certificates can replace passwords for access control, which are sometimes lost or easily cracked by experienced crackers.

Certificates

Digital certificates are nothing more than a way to verify your identity through a CA using public key cryptography. NetScreen appliances support the use of digital certificates as a method of validating your identity during VPN negotiations. There are certain steps you must take before you can use a certificate to validate your identity. First, you must generate a certificate request from within the NetScreen appliance. When this is done, the NetScreen appliance generates a public/private key pair. You then send a request with the public key to your CA. A response, which incorporates the public key, will be forwarded to you that will have to be loaded into the NetScreen appliance. This response generally includes three parts:

- The CA's certificate, which contains the CA's public key.
- The local certificate identifying your NetScreen device.
- In some cases a certificate revocation list (CRL). This lists any certificates revoked by the CA.

You can load the reply into the NetScreen device either through the WebUI or via TFTP (Thin File Transport Protocol) through the CLI (command line interface), whichever you prefer. Loading the certificate information into NetScreen gives us the following:

- Your identity can be verified using the local certificate.
- The CA's certificate can be used to verify the identity of other users.
- The CRL list can be used to identify invalid certificates.

CRLs

A CRL is used to ensure that a digital certificate has not become invalid. NetScreen appliances use CRLs to check for invalid certificates before connecting VPN tunnels. When speaking in regards to the use of digital certificates with VPNs, the certificate is validated during Phase 1 negotiations. In the event that no CRL has been loaded into NetScreen, the appliance tries to retrieve a CRL via LDAP (Lightweight Directory Access Protocol) or HTTP (Hypertext Transfer Protocol), which is defined inside the CA certificate. NetScreen appliances also enable you to specify an address to refer to for the CRL. If you do not define an address, the default address within the CA's certificate is used.

How to Use VPNs in NetScreen Appliances

In this section we'll explain how you can use VPNs in NetScreen Applicances.

Site-to-Site VPNs

There are two ways to configure site-to-site VPNs when both endpoints have static IP addresses:

- Site-to-site VPN using a manual key
- Site-to-site using an autokey (IKE)

Hosts behind either gateway can initiate the negotiations between the two gateways.

But what happens when you need to create a VPN between a gateway with a dynamically assigned address and a gateway with a static address? Does it mean you cannot create a VPN between the two? No, it doesn't. You can still create a VPN, but you need to create a dynamic peer site-to-site VPN tunnel using *autokey IKE*. Negotiation of a dynamic peer tunnel differs somewhat, though. Only hosts that are behind the dynamic gateway can initiate the VPN tunnel. This is because the remote gateway has a static IP. After the VPN tunnel is established, parties behind either gateway can pass traffic across the tunnel.

Site-to-site VPN tunnels require the configuration of both gateways. Configuration at each endpoint is identical, except the gateways are in reverse order. The default Phase 2 security level is set to *standard* when creating a new VPN tunnel. Standard security, when mentioned in regards to NetScreen appliances, is *nopfsiespdesd5*. Unless you are terminating a VPN to a device that does not support PFS, or any higher encryption and authentication methods, it is recommended that you do not use the default configuration. You should use at least *g2-esp-3des-sha*. Figures 11.4 and 11.5 show screenshots of the NetScreen WebUI VPN configuration and advanced configuration pages, respectively.

NOTE

NetScreen now includes a built-in dynamic DNS (DDNS) client that can register with a DDNS server its rotating public IP address. As a result, a NetScreen with a non-static IP address can still create a main mode IKE tunnel that both ends can initiate.

The gateway definition on the VPN configuration of a NetScreen can be an IP address *or* a hostname. If a NetScreen with a changing IP address registers its IP with a DDNS server, create the VPN connection as if both ends were static and use the DDNS hostname as the gateway address for NetScreen. This will enable both ends to initiate as long as the DDNS client on NetScreen keeps its DDNS server up to date with its current IP address.

Figure 11.4 VPN Configuration Page

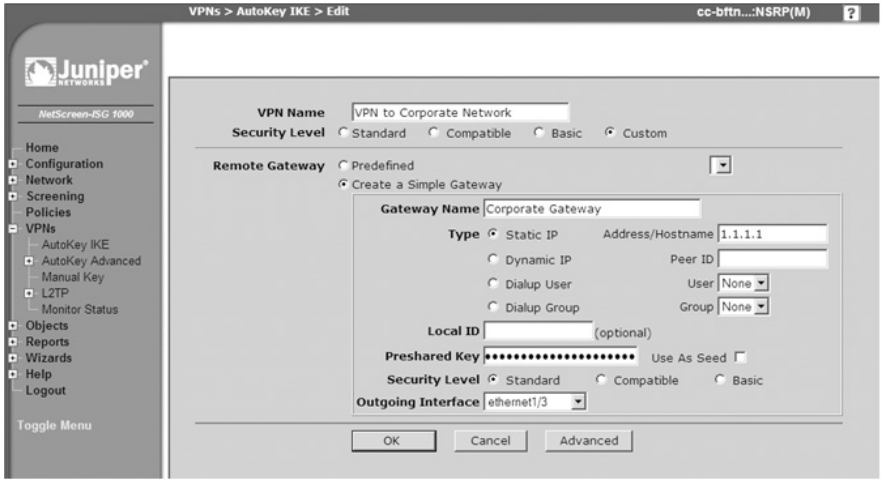
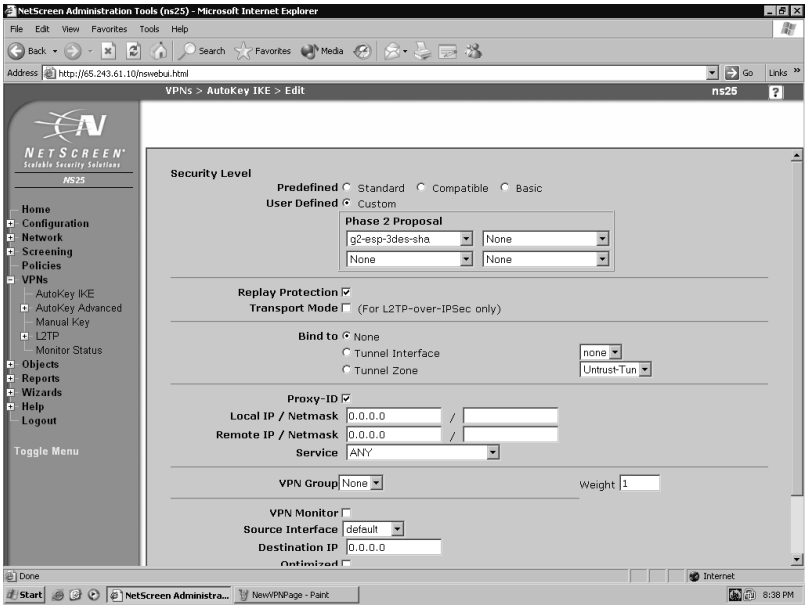


Figure 11.5 The Advanced Features Configuration Page



Policy-Based VPNs

Policy-based VPNs are VPNs that route traffic based on specific policies within a NetScreen appliance. Policy-based VPNs can be either manual key or autokey IKE. A policy-based VPN works based on specific criteria that a packet matches as it reaches the gateway. First, before you can create a policy-based VPN, you must configure the VPN tunnel. After creating the VPN tunnel you then create a policy, choose the action **Tunnel**, and select the VPN object you configured earlier. The action Tunnel works very similarly to the Permit option, except it requires you to select a tunnel object that you have previously created so that it can properly handle the traffic. A policy-based VPN tunnel always permits the traffic so long as it matches all the criteria of the rule. With policy-based VPNs, each separate traffic policy will create its own SA, so using multiple policy-based VPNs will result in using more system resources. This is true even if the destination tunnel is the same for multiple policies.

Policy-based VPNs are best used in the following situations:

- When you do not need to filter specific traffic on the tunnel.
- When you are not using any dynamic routing protocols.
- When there is no need for conserving IPSec tunnels and SAs.
- When you are using the VPN tunnel in conjunction with a dial-up VPN client.

With policy-based VPNs, you are limited in the number of tunnels you can create, depending on the number of tunnels the device can support. A sample of a configured policy using a VPN is shown in Figure 11.6.

NOTE

A policy-based VPN should almost never be used unless creating a VPN to a device that requires multiple Phase 2 SAs using the same Phase 1 gateway (i.e. multiple proxy IDs for the same gateway-to-gateway tunnel).

Figure 11.6 Policy-Based VPN Configuration

Creating a Policy-Based Site-to-Site VPN

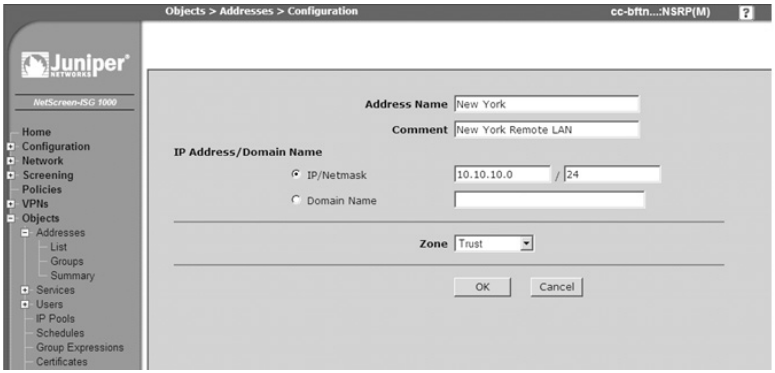
Suppose your company has two offices and wants to share resources among the two via a VPN. You need to create a policy-based site-to-site VPN that does just that. Before you can begin, you need information about the sites. Site1 uses the network 192.168.0.0/24 and has a NetScreen appliance with a static address of 4.4.4.4. Site2 uses the network 10.10.10.0/24 and has a NetScreen appliance with a static address of 5.5.5.5. You will be using autokey IKE and the preshared key will be *dgL-I2G#U438^*gyG(6t!*. You also want to use DH Group 2, AES-128, and SHA-1 for your encryption. Now that you have the necessary information, you can start to build your VPN tunnel.

First and foremost, you need to define your networks at each end of the tunnel. You do this by accessing **Objects** | **Address** | **List**. Select **New** from the top of the screen. Choose a name for the address object, such as Site2, and then add the IP address, netmask, and zone. You also need to create an address object for the local network. Name it **Trusted LAN (192.168.0.x)**. Figure 11.7 shows the configuration page for the Site1 firewall. The configuration for Site2 would also be completed as shown here, substituting the network address for Site1's local network in for the IP address. Like Site1, Site2's firewall would also contain an address object defining the local network.

After you have added the addresses to the address book, you can configure your VPN gateways. To do this, select **VPNS** | **AutoKey Advanced** | **Gateway**. Select **New** from the top of the screen. Enter a name for the gateway. Choose **Custom** for the **Security Level**, because you will be using pre-g2-aes128-sha. Later, you will configure this on the Advanced page of the gateway configuration. Since you know that Site2 has a static IP

address of 5.5.5.5, you choose the default setting **Static IP** and enter **5.5.5.5** in the available field. Now, enter the preshared key into the field labeled **Preshared Key**. You have completed the basic configuration for this end of the VPN tunnel, but you still need to set the correct proposals to be used. Click the **Advanced** button to show the advanced configuration page. In the **Phase 1 Proposal** box, select **pre-g2-aes128-sha**. Because both endpoints have static IP addresses, you should leave your **Mode** set to **Main**. After you have selected the correct proposal, scroll to the bottom of the page and select **Return** to go back to the basic configuration page. Once back at the basic configuration page, select **OK** to save the new gateway. Figures 11.8 and 11.9 show the basic and advanced configuration pages completed with your settings. To configure Site2's VPN gateway, you would use the same steps you just completed, substituting the address **4.4.4.4** as the **Static IP**.

Figure 11.7 Configuring an Address Object



Tools & Traps…

Proxy IDs

When configuring an address book entry for a policy-based VPN, make sure to note that the network and subnet mask is what is literally used for the proxy ID of the VPN. Proxy IDs must match, local and remote, on both ends for the SA to be successfully negotiated. Adding multiple-source or destination addresses to the same policy causes a proxy ID of 0.0.0.0 to be used.

Figure 11.8 Basic VPN Gateway Configuration

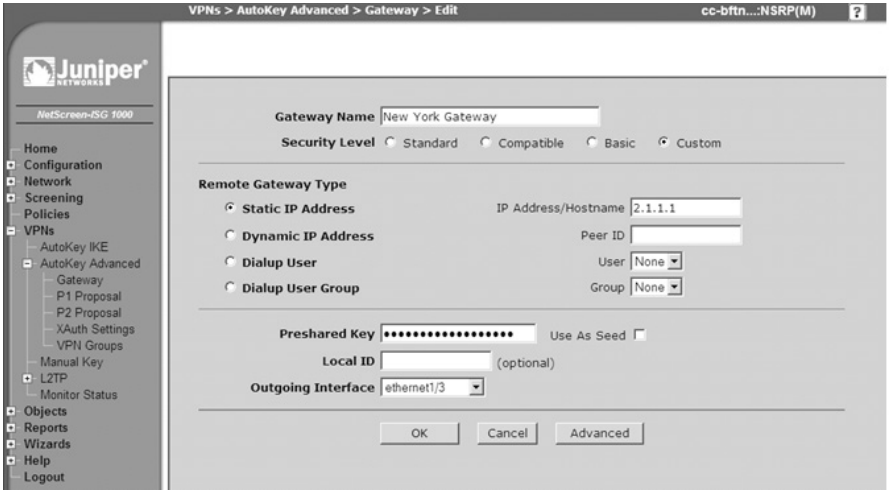
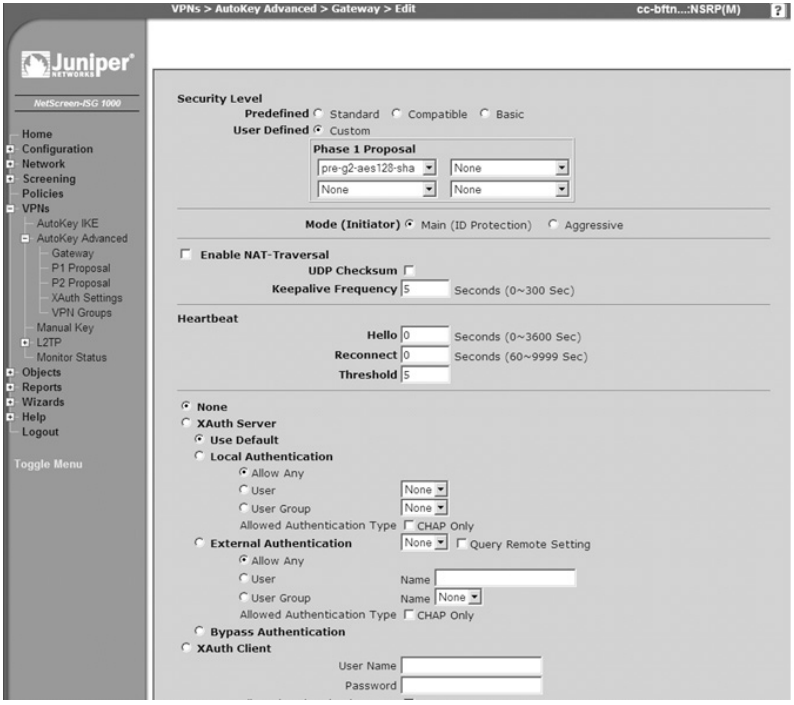


Figure 11.9 Advanced VPN Gateway Configuration



Now that you've created the VPN Gateway, you need to create an AutoKey IKE entry that uses your gateway and configure the security proposals for Phase 2. To do this, select

VPNs | **AutoKey IKE** and select **New** from the top of the screen. Give the VPN a descriptive name, such as VPN To Site2. Again, you choose **Custom** as your security level. Access the drop-down menu to the right of Remote Gateway and choose the gateway you previously configured, To Site2. Click the **Advanced** button to bring up the advanced options for your IKE entry. Use the **Phase 2 Proposal** drop-down list to select **g2-esp-aes128-sha**. Click the **Return** button to go back to the basic configuration page. Choose **OK** to save the new IKE entry. Figures 11.10 and 11.11 show the basic and advanced configuration pages for creating an AutoKey IKE entry.

Figure 11.10 Basic AutoKey IKE Configuration

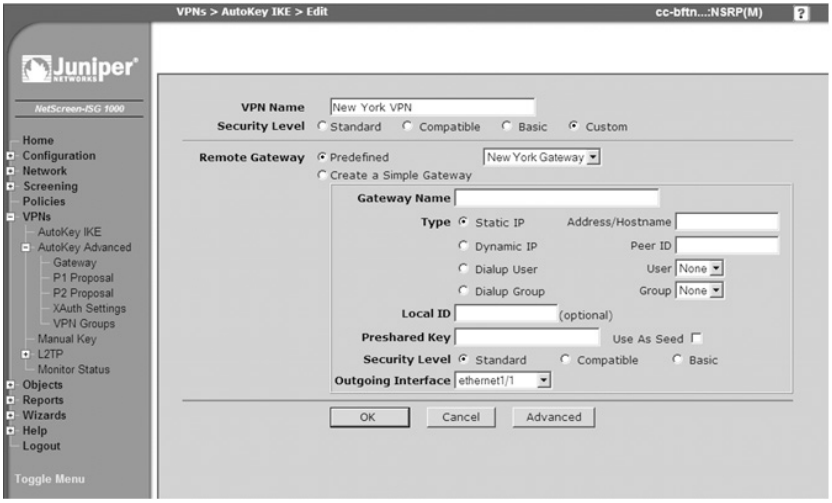
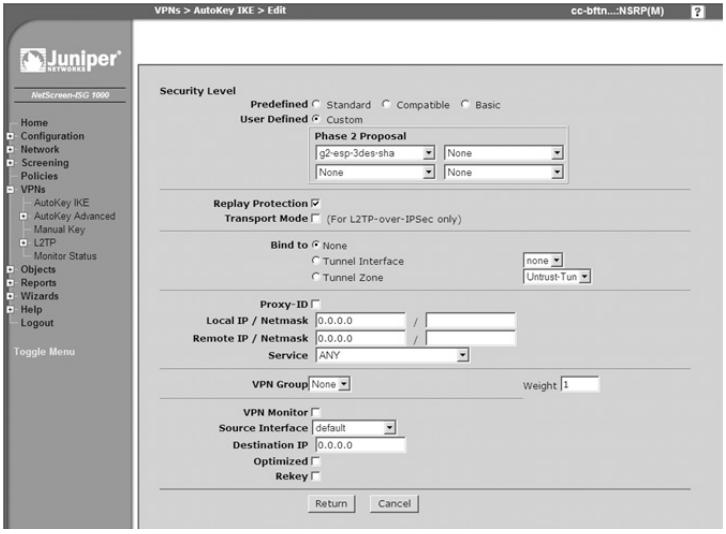


Figure 11.11 Advanced AutoKey IKE Configuration



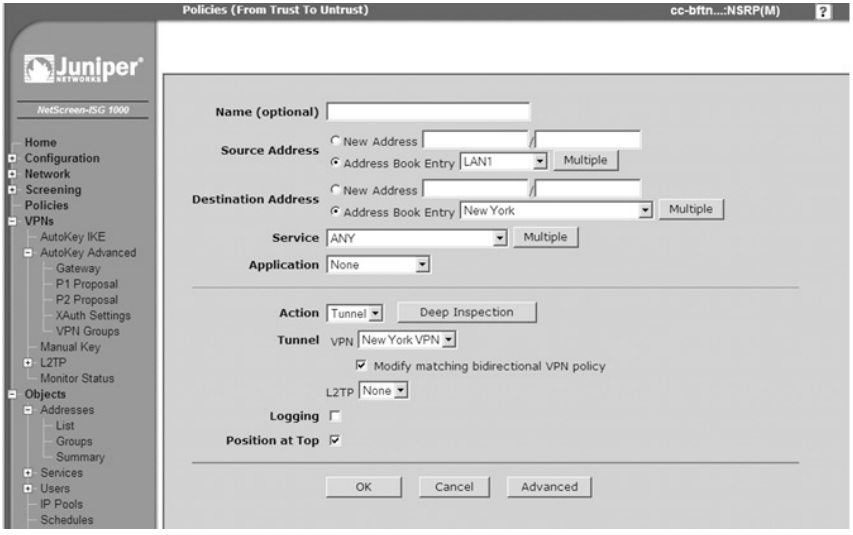
After you've completed the above steps, you need to create a policy enabling traffic to use the VPN. Click **Policies**. At the top of the page choose the options **From: Trust To: Untrust** and click **New**. Name the policy **To / From Site2**. Use the **Source Address** drop-down list to select the local network address book entry you defined earlier. Choose **Site2** as the **Destination Address** from the drop-down menu. Because you want to enable all traffic to flow between the two sites, you will leave the **Service** as **ANY**. Choose the action **Tunnel** and select the IKE entry that you created earlier, **VPN to Site2**. Check the box to **Modify** matching bidirectional VPN policy. Also enable the **Position at Top** option. Figure 11.12 shows what your policy should look like once completed.

Are You Owned?

Tunneling Traffic

Make sure that policies with an action of tunnel are placed above other existing general permit policies. The NetScreen firewall matches policies from a top down method so traffic that needs to be tunneled needs to be matched before traffic with broader source/destination addresses with an action of permit.

Figure 11.12 VPN Policy Configuration



Keep in mind that the configuration for the other end of your tunnel can be completed as outlined above, by using Site1's network information in place of Site2's. When both ends of the tunnel have been configured, the two NetScreen devices will negotiate SAs and establish a VPN tunnel. To the users, this process is transparent. In fact, most users only know they can use resources located at the other site; they have no clue as to what process enables them to do so.

Route-Based VPNs

Route-based VPNs, like policy-based VPNs, can also use either manual key or autokey IKE, but are configured and function somewhat differently. Route-based VPNs do not make reference to a tunnel object, but rather the destination address of the traffic. When the NetScreen appliance performs a route lookup to see which interface it should use to send the traffic, it sees there is a route through a tunnel interface that is bound to a VPN tunnel and uses that interface to deliver the traffic.

There are some advantages to using a route-based VPN. Using route-based VPNs is a good way to conserve system resources. Unlike policy-based VPNs, you can configure multiple policies that allow or deny specific traffic to flow through a route-based VPN, and all of these policies will use a single SA. Route-based VPNs also offer the capability to exchange dynamic routing information, such as Open Shortest Path First (OSPF), on the tunnel interface. Route-based VPNs enable you to create policies that have an action of deny, unlike policy-based VPNs. Route-based VPNs also have different limitations than policy-based VPNs. With route-based VPNs, you are limited by one of two things: the number of route entries your appliance supports, or the number of tunnel interfaces your appliance supports, whichever of the two is the least.

To configure a route-based VPN, the process is similar to a policy-based VPN with only a few exceptions. When defining the Phase 2 AutoIKE, under the Advanced Section, you bind the VPN to a specific tunnel interface. You also create static routes (or configure a dynamic routing protocol) to send traffic down the tunnel interface. Finally, and only if necessary, create policies to permit the traffic (but *not* tunnel). In a routing-based VPN, no policy is needed with an action of "Tunnel." Therefore, as long as the SA is active, the routing-based VPN will function.

Dial-Up VPNs

NetScreen appliances support the use of dial-up VPN. A dial-up VPN is a VPN that connects using either the NetScreen remote VPN client, or another NetScreen appliance that does not have a static IP address (a dynamic peer). NetScreen appliances support dial-up VPN configurations based on a per-user or per-group basis. Using a group saves time and makes things easier on the administrator because there is only the need to configure one tunnel. You can also configure a group IKE user and use the IKE ID for the dial-up users' group.

One of the nifty things about using a dial-up VPN is that you can actually configure policy-based VPNs for the dial-up users. For example, if you want a user to be allowed to connect to 10.11.12.13 to send e-mail, you can create a policy allowing access only to 10.11.12.13 on port 25. Later, if you want to allow that same user access to his POP3 (Post Office Protocol v3) mailbox on server 10.11.12.14, you can simply add another policy to the remote client allowing the user access to 10.11.12.14 on port 110.

For a dynamic peer, you can use either policy-based or route-based VPN. Just remember that in order to use a route-based VPN, you need to configure an internal virtual IP address. The NetScreen Remote client, which we will discuss a bit more in the next section, also supports having a virtual IP address and can be used with a route-based VPN as well.

NOTE

You can also accommodate dial up (or software/client-based IPSec VPNs) without the use of the pre-defined dial-up VPN address entry in NetScreen by binding the VPN to a tunnel interface and setting the proxy IDs appropriately. Additionally, bidirectional policies are possible using the dial-up VPN entry to enable client-to-server– and server-to-client–initiated connections.

NetScreen Remote

NetScreen Remote is Juniper Network's software VPN client. NetScreen Remote can be installed onto almost any Windows-based desktop PC. NetScreen Remote is primarily used for policy-based dial-up VPNs. It is also often used by home users working from a cable or DSL modem connection, which are still considered dial-up VPNs because the IP address of the client is usually dynamic. As previously mentioned, NetScreen Remote also supports the use of route-based VPNs.

After NetScreen Remote has been installed, it starts with each logon to Windows. Configuration of NetScreen Remote is easily completed using the GUI-based Security Policy Editor tool.

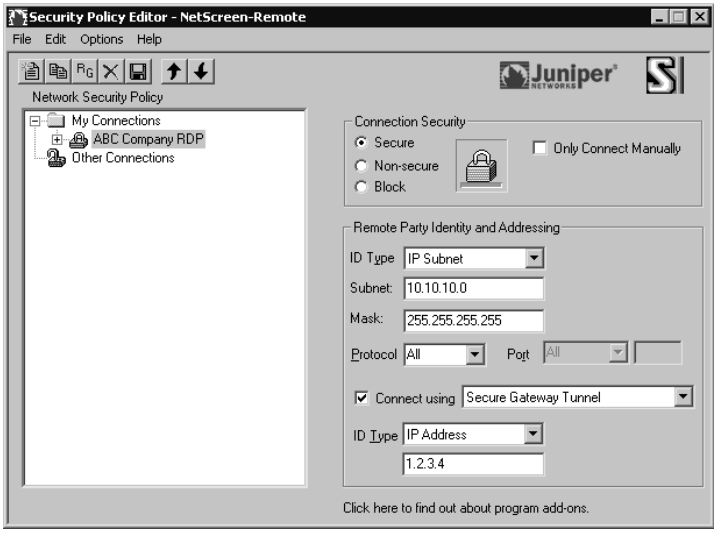
Let's walk through a scenario of configuring NetScreen Remote to connect to ABC Company's corporate network and enable us RDP access to a terminal server. Suppose we will be connecting to ABC's corporate network via a DSL modem with a dynamic IP address. ABC's firewall administrator has provided us with the following information to help us configure a policy in NetScreen Remote.

ABC's NetScreen IP:	1.2.3.4
IP Subnet:	10.10.10.0
Netmask:	255.255.255.0
Pre-shared key	NetScreen-Firewalls-Rule!

ABC's administrator has also told us that he would like us to use replay protection, perfect forward secrecy, DH Group 2, AES-256 for encryption, and SHA-1 for the hashing algorithm in all applicable phases. During our conversation with ABC's administrator, he asked for our e-mail address to use as our identity and we provided him with the address clathem@domain.tld. Now that we have all the necessary information, let's create a policy in NetScreen Remote that will properly direct and secure our traffic.

Click **Start** | **Programs** | **NetScreen-Remote** | **Security Policy Editor** to start the NetScreen Remote Security Policy Editor. Choose **Edit** | **Add** | **New Connection** to add a new policy to the My Connections heading. Since the NetScreen Remote software can contain multiple security policies, we should name this policy with a descriptive name so that in the future we can easily tell what this policy does. Right-click on this policy, choose **Rename**, and rename this policy **ABC Company RDP**. Highlight the name of the policy. We will start here, configuring the remote gateway and destination network on this screen. Under **Remote Party Identity and Addressing** change the **ID Type** to **IP Subnet**. For the **Subnet**, enter our destination network, **10.10.10.0**, and in the **Netmask** enter **255.255.255.0**. Leave **Protocol** as **All**. Next we need to enable the option **Connect Using Secure Gateway Tunnel**. Since we know that ABC's gateway IP is **1.2.3.4**, we enter it into the **IP Address** field. When completed, our initial screen should look as pictured in Figure 11.13.

Figure 11.13 Initial Configuration of NetScreen Remote



Next, click the plus sign to the left of the policy to expand it one level. You now see the option to configure **My Identity**, where you choose to use a preshared key or certificate, and **Security Policy**, in which you configure which encryption and authentication options you will use during communications. Start off by selecting **My Identity**. Since ABC's firewall admin has assigned us a preshared key, we will want to change the **Select certificate**

option to **None**. Upon doing so, a button labeled **Pre-shared Key** appears. Click this button to bring up the Preshared key dialog box and then click **Enter Key**. Figure 11.14 shows an example of the preshared key dialog box. Enter the key as assigned to us by ABC's administrator and click **OK**. Since ABC's administrator is using our e-mail address for our identity, we select the option for e-mail address under **ID Type** and enter our e-mail address, **clathem@domain.tld**, into the field. We will leave the fields **Virtual Adapter** and **Internet Interface** as the default choices. Figure 11.15 shows the settings completed in NetScreen Remote.

Figure 11.14 Entering the Preshared Key

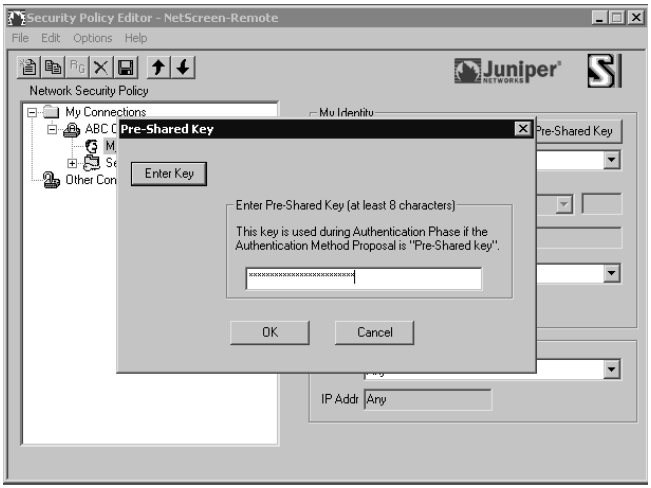
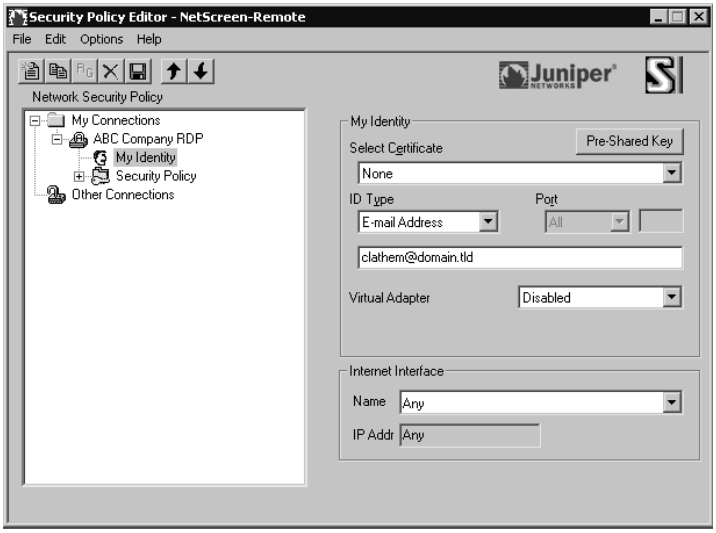
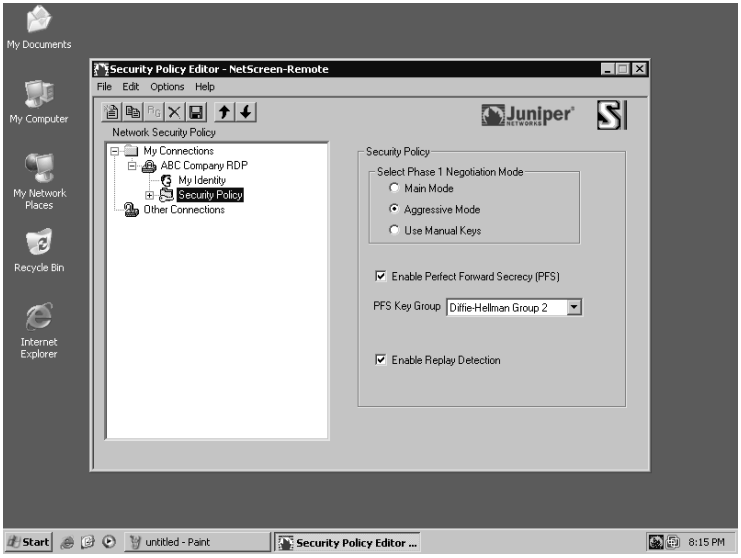


Figure 11.15 Configuring Our Identity



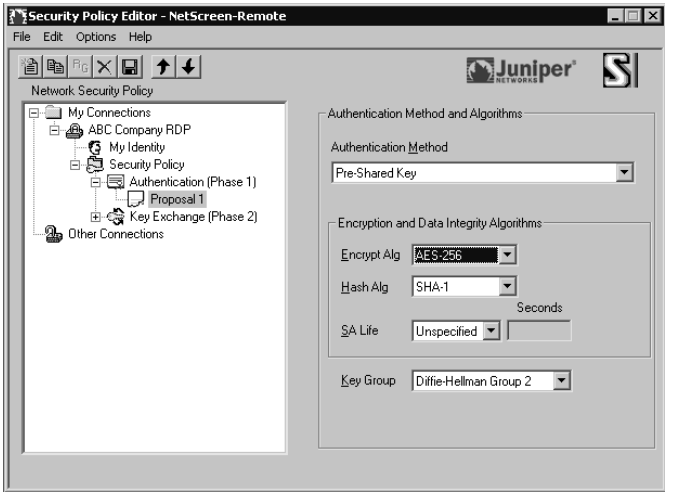
Now that we've completed the setup of our identity and preshared key, we need to complete the configuration of our security policies. Start off by selecting **Security Policy**. Since we will be connecting from a DSL modem with a dynamic IP address, we need to use **Aggressive Mode**, so let's select that option. ABC's administrator also specified that we use perfect forward secrecy, so we need to enable the **PFS** option. We were told to use DH Group 2 in all phases, so the default selection of **Diffie-Hellman Group 2** is fine. Replay detection is also enabled by default. Figure 11.16 shows our Security Policy configuration.

Figure 11.16 Initial Policy Configuration



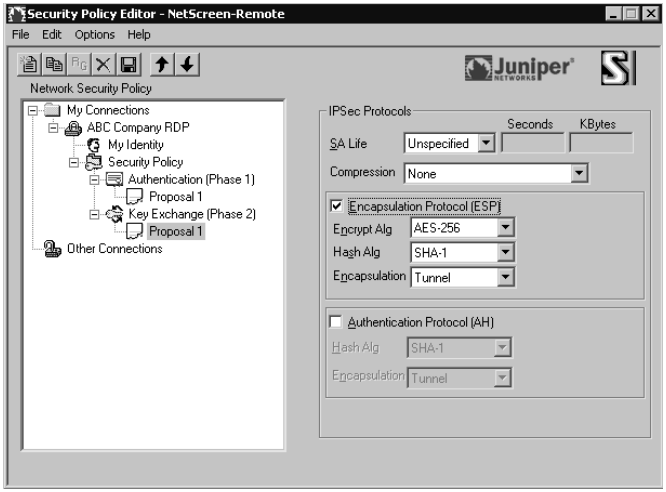
Now that we've completed the initial security policy configuration, we can move on to configuring each individual policy. Click the plus to the left of **Security Policies**. Here you will see the settings for Phase 1 and Phase 2 proposals. Start off by expanding **Authentication (Phase 1)** and then click on **Proposal**. Note that the default settings for Phase 1 negotiations in NetScreen remote is to use a preshared key, Diffie-Hellman Group 2, Triple DES for Encryption, and SHA-1 for hashing, or as I and the NetScreen firewall like to refer to it, PRE-G2-3DES-SHA. Since ABC's administrator specifically noted the use of AES-256 for encryption, we change the **Encryption Alg** field to reflect his specifications. Figure 11.17 shows Phase 1 properly configured as specified.

Figure 11.17 Configuring Phase 1 Proposal



The last configuration modification we must make is under the **Key Exchange (Phase 2)** heading. Expand the heading and select **Proposal**. Look under the **Encapsulation Protocol (ESP)** label. Change the encryption algorithm to **AES-256**. Figure 11.18 shows the completed configuration for Phase 2.

Figure 11.18 Configuring Phase 2 Proposal



After completing the configuration of the policy, we need to save our changes. Click **File** | **Save**. We're done! We've completed the setup of a security policy for NetScreen Remote.

Now that we've completed the setup, we should test to ensure that our policy works properly. Right-click the NetScreen Remote icon in the system tray and choose **Connect… | My Connections\ABC Company RDP**, as illustrated in Figure 11.19. If all is well, you will see a dialog box notifying you that the connection to ABC Company RDP was successful. If negotiations fail, you will receive a message notifying you of the failure to connect. It's probably a good time to open the NetScreen Remote Log Viewer, as shown in Figure 11.20. The Log Viewer is an excellent place to start troubleshooting the VPN.

Figure 11.19 Activating the NetScreen Remote Policy

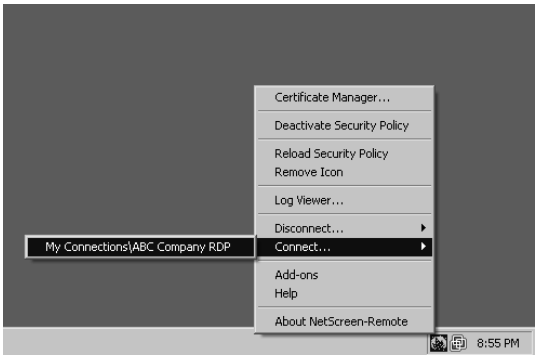
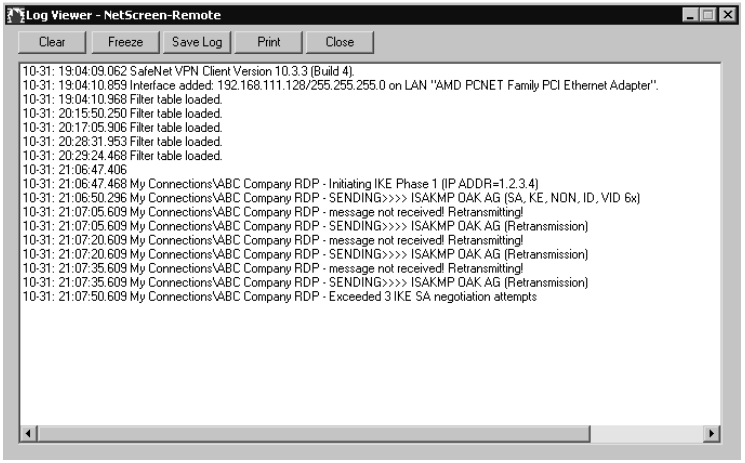


Figure 11.20 NetScreen Remote Log Viewer



L2TP VPNs

NetScreen appliances support the Layer 2 Tunnel Protocol, or L2TP for short, when operating in Layer 3 mode. The L2TP protocol works by sending PPP (Point-to-Point Protocol) frames through a tunnel between the LNS and the L2TP access concentrator. Originally,

L2TP was designed so that a dial-up user could make a virtual PPP connection through an L2TP access concentrator (LAC) at an ISP. The LAC at the ISP would create a tunnel to the L2TP network server at either another ISP, or at a corporate network. The L2TP tunnel never actually extended to the client's desktop, only to the ISP's LAC.

L2TP tunnels are not encrypted, so they are not actually true VPN tunnels. The primary purpose for L2TP is that a dial-up user can be assigned an IP address that is known and can be referenced in policies. To encrypt an L2TP tunnel, you need to use an encryption scheme such as IPSec. Generally, this is referred to as L2TP-over-IPSec. L2TP-over-IPSec requires two things: IPSec and L2TP tunnels to be set up with the same endpoints and then linked together in a policy, and the IPSec tunnel must be in transport mode.

NOTE

Modern operating systems, such as Windows XP, can alone act as an LAC, so that an L2TP tunnel can extend all the way to the desktop. NetScreen devices can act as LNS servers, so an L2TP VPN can easily be created between a NetScreen appliance and a Windows 2000 desktop, provided you don't mind tweaking your registry a bit. To use L2TP without IPSec, change the value of the registry key (or create if one does not exist) ProhibitIPSec at HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\RasMan\Parameters to hexadecimal 1 and reboot.

The NetScreen device does need to be configured with a group of IP addresses to assign to the L2TP clients, and these IP addresses must differ from the subnet in use on the LAN. For example, if your LAN address range is 10.0.0.0/24, then you would need to use something outside this range, such as 10.0.1.0 or 10.0.2.0. Note that you can use private address ranges that are not routable on the Internet. When the client connects to the NetScreen appliance, it is assigned an IP address for the L2TP tunnel, as well as DNS (Domain Name Service) and WINS (Windows Internet Naming Service) servers if applicable. The NetScreen appliance can also perform PPP authentication for the client through RADIUS, LDAP, SecurID, or its own internal database. NetScreen appliances support the use of Challenge Handshake Authentication Protocol (CHAP) with RADIUS and its internal database. NetScreen appliances also support Password Authentication Protocol (PAP) with RADIUS, LDAP, SecurID, and its internal database.

Advanced VPN Configurations

In this section, we'll discuss advanced VPN configurations.

VPN Monitoring

Suppose you want to monitor the status of VPN tunnels you've configured. By enabling VPN monitoring, you can do just that. NetScreen offers the capability to monitor VPN tunnels via SNMP (Simple Network Management Protocol). ICMP (Internet Control Message Protocol) echoes (widely known as *pings*) are sent through the tunnel at user-configurable intervals to monitor connectivity. Should the pings indicate a change in the state of the tunnel, an SNMP trap is triggered. A trap of *up to down* is triggered when the monitoring state of a tunnel is up, but a number of consecutive pings are sent without a reply, and there is no other VPN traffic flowing through the tunnel. A *down to up* trap is triggered when the tunnel monitoring state is down, but a ping request receives a reply. Note that it only takes one successful ping reply to change the state to up, and the rekey option must be disabled. The NetScreen WebUI also has a page that can show you the status of a tunnel. To view this page, click to expand the VPN tab and then click Monitor Status.

Netscreen appliances report the following information on VPN tunnels via SNMP.

- Number of active VPN sessions.
- Start time of each VPN session.
- The SAs for each session, including encryption and authentication method, IKE protocol, VPN type, peer and local IP addresses and gateway ID, security parameter index, and Phase 1 authentication method.
- Tunnel state.
- VPN monitoring state.
- Phase 1 and 2 state and lifetime.

When monitoring is enabled, NetScreen will perform monitoring only when the SA is up and operable. By enabling the rekey option, the firewall will continuously send ICMP echo requests whether or not the SA is even up or active. This will force NetScreen to always attempt to bring up the VPN tunnel (in essence nail it up).

VPN monitoring without continuously sending ICMP echoes can be achieved by enabling the optimized option. When enabled, NetScreen will consider incoming traffic coming through the VPN as an indication that the tunnel is operating. This in essence enables normal incoming traffic to be a substitute for ICMP echoes. NetScreen will also disable ICMP echoes altogether when there is incoming and outgoing traffic present on the VPN.

NOTE

Enabling VPN monitoring on a tunnel that is bound to a tunnel interface (routing-based VPN) can automatically make routes associated for that tunnel interface active or inactive depending on whether or not the VPN monitor is in a

state of Up or Down. This allows for powerful failover routing if a particular tunnel goes down.

You can monitor the state of the VPN monitor by navigating in the WebUI to VPN | Monitor Status or on the command line using the *get sa* command.

Tools & Traps…

Enabling the VPN Monitor

When enabling the VPN monitor, you might need to take into consideration the security policies and routing table of NetScreen. NetScreen must be able to route to the destation of the echoes and a security policy might be needed for the ICMP echoes to be successful. For example, enabling the VPN monitor and rekey option but not having an appropriate security policy (that is, if transiting two security zones) will cause the tunnel to bounce/never remain stable.

Gateway Redundancy

When you have the need for gateway redundancy and failover to provide continuous connectivity, NetScreen has the answer. You can create a group of up to four redundant VPN gateways that policy-based, site-to-site VPNs can connect to. These gateways can have the same parameters for Phase 1 and 2 SAs, or they can be totally different.

NetScreen appliances use two mechanisms to monitor individual endpoints of a VPN group: IKE heartbeats and recovery attempts. By combining these two mechanisms with *TCP-SYN Flag checking*, NetScreen devices can failover to a new gateway without any disruption in service. TCP-SYN Flag checking is a check performed by NetScreen appliances to ensure that the TCP SYN flag is set when an initial TCP (Transmission Control Protocol) session is attempted. When a failover occurs, the gateway that picks up the communications assumes the first packet received is the start of a new TCP session and expects the SYN flag to be set. Since the packet is part of an existing TCP session, the SYN flag is not set, and the packet is dropped. Thus, when TCP-SYN flag checking is enabled, all applications have to reconnect after a failover. To enable a truly successful failover without loss of connectivity, you should disable SYN flag checking in VPN tunnels. Currently there is no way to disable SYN flag checking via the WebUI; you must use the CLI command *unset flow tcp-syn-check-in-tunnel*.

It is important to note that VPN groups do not support L2TP, dial-up, manual key, or route-based VPNs. NetScreen redundant gateway VPN groups also support policy-based

dynamic peer IKE VPNs, provided that the members of the VPN group have static addresses and the dynamic address is on the appliance doing the monitoring.

"But how does it work?" you ask. When an appliance starts negotiation of a VPN that points to a VPN group, negotiations are performed with all members of the group. Traffic is then directed to the destination gateway with the highest priority in the group. So what happens with the other members of the VPN group? The initiating party keeps the tunnels in an active state, sending heartbeat packets through them. If the higher priority active tunnel fails, the tunnel with the next highest priority is quickly put into play by the VPN monitor, and traffic continues to flow.

Back-to-Back VPNs

Back-to-back VPNs are used to enforce interzone policies between two spoke sites through the hub site. There are several advantages to using back-to-back VPNs. Using back-to-back VPNs can reduce the number of tunnels you need to create. This can be especially helpful on NetScreen appliances that support smaller numbers of VPN tunnels. Take a NetScreen 5XP for example, which supports only 10 tunnels. If you had to create multiple VPN tunnels to several different VPN gateways, you would quickly consume all your VPN resources. But if you use back-to-back VPNs, you can create a single tunnel from each site to the hub site, and then route traffic between all of the sites through the hub site. You've accomplished the same results as with multiple VPN tunnels, enabling traffic to pass between all of the sites, but have done so with only one VPN at each spoke's gateway.

Another advantage of back-to-back VPNs is the capability to define policies between sites. Enforcing policies between spoke sites can be accomplished by placing each of the sites into different zones. Since each site is located in a different zone, NetScreen must perform a policy lookup before routing the traffic to the destination site. This effectively enables you to control which traffic is allowed between your spoke sites. Suppose both of the spokes terminate at the same interface, but you still want to be able to control traffic between the two. Simply enable intrazone blocking and then define policies between the tunnel interfaces.

The administrator of the hub site can also control the flow of all traffic from the remote sites. By defining a policy at each spoke site that passes all traffic from the trusted network destined for the outside world across the VPN to the hub site, the administrator can use policies at the hub site to filter traffic.

Hub and Spoke VPNs

Hub and spoke VPN tunnels route traffic directly from one spoke VPN tunnel to another spoke VPN tunnel terminated on the hub appliance. This is done by adding a pair of routes to the route table. When intrazone blocking is disabled, the hub site only needs to perform a route lookup in order to properly forward the traffic. The major advantage of using hub and spoke VPN technology is circuit aggregation. By using hub and spoke VPNs, the hub site can have as few as one circuit connecting it to the spoke sites and use this single circuit to

route traffic to the spoke sites. When using another technology such as frame relay, the hub site will have several circuits terminating at the site and will need to use several ports and routers in order to interconnect the spoke sites.

Multitunnel Interfaces

NetScreen appliances support the capability to have multiple IPSec VPN tunnels bound to one interface. In fact, you can bind as many tunnels as your NetScreen appliance supports to the same interface, provided the route table is not filled first. NetScreen appliances use both the route table and next-hop tunnel binding to link a destination address to a specific tunnel on a tunnel interface.

Summary

A Virtual Private Network is a means of creating secure communications over a public or insecure network infrastructure. In essence, VPNs are a way to share private information over a public infrastructure. VPNs are deployed using a variety of protocols to facilitate encryption (for privacy and confidentiality) and authentication (to maintain the integrity and verification of the source). VPNs are widely in use today over the Internet because they are a cost-effective and a readily available transport medium. As a result, VPNs are widely replacing legacy frame relay and point-to-point networks while still providing the same (if not greater) level of performance, reliability, and security.

The VPN support in a NetScreen appliance is extremely flexible and easily managed. NetScreen appliances support several types of VPNs, including IPSec and L2TP, and they can facilitate network-to-network as well as user-to-network (remote access) tunnels. NetScreen appliances can also be deployed with sophisticated routing-based VPNs, which can support dynamic routing protocols such as OSPF or BGP to facilitate traffic engineering, redundancy, and high availability.

Solutions Fast Track

Understanding IPSec

- ☑ IPSec was engineered to provide several services: privacy and confidentiality of data, origin authentication, data integrity, access control, and protection against replay attacks.
- ☑ The IPSec protocol provides two modes of operation: *transport mode* and *tunnel mode*.
- ☑ IPSec has two methods for verifying the source of an IP packet as well as verifying the integrity of the payload contained within: authentication header (AH) and encapsulating security payload (ESP). While ESP can encrypt and authenticate the entire packet, AH only authenticates the packet.
- ☑ IPSec supports the use of both manual keys and autokey IKE.
- ☑ Internet Key Exchange, or IKE, generates and negotiates keys and SAs automatically based on either preshared secrets or digital certificates and takes place in two phases, 1 and 2.
- ☑ SA is the concept used by IPSec to manage all the parameters required to establish a VPN tunnel, including security keys and algorithms, mode of operation (transport or tunnel), key management method (IKE or manual key), and lifetime of the SA. All this information is stored in the SA database (SAD).

IPSec Tunnel Negotiations

- ☑ For a manual key VPN, all SA information including the IP, index, and shared secret is manually configured. Negotiations do not occur between the two endpoints. Traffic is simply encrypted, authenticated, and routed to the destination gateway.
- ☑ IPSec tunnels using IKE require two phases to complete negotiation: Phase 1 establishes a secure tunnel for negotiation of SAs and Phase 2 IPSec SAs are negotiated defining the method for encrypting and authenticating user data exchange.
- ☑ Phase 1 exchanges can be done in two modes: main mode or aggressive mode. In main mode, six messages are exchanged, while in aggressive mode only three messages are exchanged.

- ☑ Main mode negotiations are considered more secure than aggressive mode negotiations, because the identities of the participating parties are not exchanged in the clear.

Public Key Cryptography

- ☑ Public key cryptography is the modern cryptographic method of communicating securely without having a previously agreed upon secret key.
- ☑ Public key cryptography uses a pair of keys to secure communications: a private key that is kept secret and a public key that can be widely distributed.
- ☑ Some examples of public key cryptography algorithms include RSA, DH, and ElGamal.
- ☑ PKI is the meshing of encryption technologies, services, and software together to form a solution that enables businesses to secure their communications over the Internet.
- ☑ Digital certificates are a way to verify identities through a CA using public key cryptography.
- ☑ CRLs are used to ensure that a digital certificate has not become invalid.

How to Use VPNs in NetScreen Appliances

- ☑ There are two ways to configure site-to-site VPNs when both endpoints have static IP addresses: site-to-site with AutoKey IKE and manual key VPN.
- ☑ A VPN can also be created between two NetScreen appliances when one endpoint has a dynamic IP address. The negotiations of the tunnel must be initiated by the end with the dynamic IP and aggressive mode must be used for Phase 1 negotiations.
- ☑ When creating VPN tunnels it is advisable to always use at least 3DES for encryption and SHA-1 for hashing. It is also advisable to use at least Diffie-Hellman Group 2 and enable PFS.
- ☑ Policy-based VPNs route traffic based on specific policies within a NetScreen appliance and can be either manual key or autokey IKE.
- ☑ When using policy-based VPNs, each separate traffic policy will create its own SA, so using multiple policy-based VPNs will result in using more system resources, even if the destination tunnel is the same for multiple policies.
- ☑ Route-based VPNs can use either manual key or autokey IKE. They do not make reference to a tunnel object, but rather the destination address of the traffic. When

the NetScreen appliance performs a route lookup to see which interface it should use to send the traffic, it sees a route through a tunnel interface bound to a VPN tunnel and uses that interface to deliver the traffic.

- ☑ Using route-based VPNs is a good way to conserve system resources over the use of a policy-based VPN.
- ☑ Route-based VPNs can accommodate dynamic routing protocols such as OSPF or BGP.
- ☑ NetScreen appliances support dial-up VPN configurations based on a per-user or per-group basis. Using the group VPN saves time by enabling the administrator to configure a single tunnel for the group.
- ☑ NetScreen appliances support the L2TP when operating in Layer 3 mode.

Advanced VPN Configuration

- ☑ NetScreen appliances support VPN monitoring via SNMP traps or through the WebUI. They also support VPN monitoring using NetScreen Security Manager or NSM.
- ☑ Netscreen appliances support the creation of a group of up to four redundant VPN gateways that policy-based, site-to-site VPNs can connect to. These gateways can have the same parameters for Phase 1 and 2 SAs, or they can be totally different.
- ☑ To enable a successful failover to a redundant gateway without loss of connectivity, you should disable SYN flag checking in VPN tunnels, using the CLI command *unset flow tcp-syn-check-in-tunnel*.
- ☑ Back-to-back VPNs are used to enforce interzone policies between two spoke sites through the hub site.
- ☑ The major advantage of using hub and spoke VPN technology is circuit aggregation, enabling multiple spoke sites to be terminated through a single circuit at the hub site.
- ☑ NetScreen appliances support the capability to have multiple IPSec VPN tunnels bound to one interface, enabling as many tunnels as your NetScreen appliance supports to be bound to the same interface (provided the route table is not filled first).

Links to Sites

- www.juniper.net/company/communities/Juniper's official discussion forum, which includes firewall communities
- www.qorbit.net/nn/—Juniper NetScreen Mailing List Archive; an excellent place to search for answers to common problems
- www.juniperforum.com/—Discusion forums dedicated to everything Juniper
- ftp://ftp.rfc-editor.org/in-notes/rfc2631.txt—RFC 2631, the Diffie-Hellman key-exchange protocol

Mailing Lists

- www.qorbit.net/mailman/listinfo/nn—The qorbit mailing list dedicated to Juniper NetScreen products

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: Can NetScreen firewalls establish VPN tunnels between other manufacturer's firewalls, such as Cisco PIX or CheckPoint?

A: Yes, NetScreen firewalls have a broad range of compatibility modes built in. By using different Phase 1 and Phase 2 proposals, the capability to create custom proposals with custom lifetimes, and control over proxy IDs, you can make a NetScreen operate seamlessly with any IPSec-based firewall on the market.

Q: What encryption and hashing algorithms do NetScreen appliances support?

A: NetScreen appliances support Diffie-Hellman Groups 1, 2, and 5. NetScreen appliances support DES, 3DES, AES-128, AES-192, and AES-256 for encryption algorithms. For hashing, NetScreen appliances support MD5 and SHA-1.

Q: How can I debug or troubleshoot why my VPN is failing?

A: NetScreen appliances have a wealth of troubleshooting and debugging capabilities. The best place to start looking for the problem is by observing the device log on the responding side of the VPN. This should reveal at what point (and why) the proposals or other criteria are being rejected. Common error messages are listed below in another FAQ. If a SA was successful but traffic is not passing, you can enable logging on the policies, permitting the traffic to observe the traffic and further troubleshoot. Finally, from the command line, use the *debug ike ?* to see all the various options available to get a verbose output of every step along the way. The SA can also be debugged by using the *debug sa ?* command.

Q: How can I clear everything relating to an SA to completely force a new negotiation?

A: Use the command *clear ike X.X.X.X or clear ike all*.

Q: I am seeing the following error in my event log relating to the VPN. What should I do to correct it?

A: Here is a list of common VPN configuration errors and what you should change to resolve them:

Rejected an IKE packet… The preshared keys might not match.
This one is as obvious as it looks. The preshared keys on both ends don't match.

Rejected an IKE packet because there were no acceptable [Phase 1/Phase 2] proposals.
Make sure the encryption and authentication proposals match on both ends. You can also use the *debug ike* command to view what proposals are being exchanged and why a match isn't being made.

Rejected an IKE packet … An initial Phase 1 packet arrived from an unrecognized peer gateway.
Either the gateway proposing the SA isn't configured/its IP address is incorrect on NetScreen, or when you created the gateway it's bound to the incorrect interface. When creating a gateway on a NetScreen, make note of the outgoing interface. Unfortunately, you have to remove the VPN and re-create the gateway to change the outgoing interface.

Phase 2: No policy exists for the proxy ID received.
Generally, this message on NetScreen will also indicate what the proxy ID was that was received. If you have a policy-based VPN, the source and destination address in the policy will determine what proxy IDs NetScreen will use. They must match with what is being proposed. If it's a routing-based VPN, you can specify the proxy IDs in the advanced page of the AutoIKE configuration. Make sure to enable the "Proxy ID"

checkbox when configuring. If you configure a routing-based VPN without specifying a proxy ID, NetScreen will use 0.0.0.0/0.0.0.0 as the proxy ID.

Q: My VPN comes up and the SA goes active, but it stops working and I have to either clear it or wait forever in order for it to come back up.

A: Most likely the lifetime values for the SA/proposals are not the same on both ends.

Chapter 12

High Availability

Solutions in this chapter:

- The Need for High Availability
- High-Availability Options
- Improving Availability Using NetScreen SOHO Appliances
- Introducing the NetScreen Redundancy Protocol
- Building an NSRP Cluster
- Determining When to Fail Over: The NSRP Ways
- Reading the Output from *get nsrp*
- Using NSRP-Lite on Midrange Appliances
- Creating Redundant Interfaces
- Taking Advantage of the Full NSRP
- Failing Over
- Avoiding the Split-Brain Problem
- Avoiding the No-Brain Problem
- Configuring HA through NSM

Introduction

As the reliance on data networks becomes greater, the importance of their availability increases. This chapter provides a comprehensive look at the features provided by various NetScreen firewalls for achieving high-availability (HA) networks. The effort involved in understanding and implementing the most highly available networks is significant and can be a daunting task. This chapter explores the available options in a progressive manner, building on the previous knowledge as much as possible. Juniper Networks have gone to great lengths to provide features that are both complete and appropriate for improving the availability of networks.

This chapter begins with a discussion about the nature of, and justification for, high-availability networks. Having a feel for the multifaceted nature of this topic is a great help, especially when trying to justify planned expenses to upper management.

Next, we examine how high availability can be achieved using the different methods and features available across the NetScreen range of firewalls. Several configuration examples are provided that can be used as a baseline to develop high-availability solutions appropriate to your specific network.

Towards the end of the chapter, some of the more advanced issues are presented and ideas and recommendations are given on how to best approach them.

Throughout this chapter, there are examples with configuration instructions for both the command-line interface (CLI) and the Web interface. Any instructions for configuration via the Web interface assumes that the firewall is already configured with an Internet Protocol (IP) address. It is recommended you use the CLI for several reasons: it is always available via the console port regardless of configuration (unless explicitly disabled), some commands are only available via the CLI, and it is much easier to work with than the Web interface. However, if you have a large installation, the NetScreen Security Manager (NSM) platform is the better choice, because it offers a graphical user interface (GUI) and the ability to easily manage several firewalls. In the end, the interface you use to configure your NetScreen firewalls should be the one you feel most comfortable with and that can get the job done.

The Need for High Availability

Whether due to hardware or software faults, one fact cannot be disputed: network components fail. The only issues that can be debated are the frequency of the failure and the impact that each type of failure will have. HA is about mitigating the risks of network failures and bringing them within acceptable bounds, which are (or at least should be) dictated by your business strategy. Do you depend on your e-commerce Web site to be available 24 hours a day, 365 days a year? If so, your idea of acceptable network outages will be vastly different from someone whose business only relies on the network for sending and receiving occasional e-mails. Knowing your business strategy is the first step towards being able to decide which HA measures, if any, you should add.

The next step in justifying HA is to understand the trade-off between the cost and the improvement gained. Cost is not only measured in money, but also in time and complexity. A highly available network takes longer to implement, results in more maintenance work, takes longer to gain an understanding of, and due to increased complexity, raises the risks of human error. Making an informed decision is not as easy as you might think.

Depending on a business's needs, HA can be anything from having a spare unit in the storage room to using a fully meshed, fully redundant network infrastructure with automatic failure detection and failover. Generally speaking, the term HA is only used for situations where a standby device is already configured and can be brought into play at a moment's notice.

High-Availability Options

Every critical network requires thorough planning for high availability, redundancy, and load balancing. This planning involves combining servers through clusters, making routers highly available by using router redundancy protocols, and managing multiple firewall appliances.

Juniper NetScreen on their firewall range of products offers three types of HA:

- **Active/Passive** In this, there are two firewalls: one primary and one backup. When the primary firewall fails, the backup firewall is activated, ensuring that the network continues to run smoothly. As a result, the full load of the network is on only one firewall at any given point in time, while the other firewall remains idle waiting for a failover to happen.
- **Active/Active** In this, both firewalls actively participate in the network functions and share the network load. Both firewalls act as peers to each other. When one of these firewalls fails, the other handles the full load of the network. Proper design consideration is required while choosing this firewall model. You should carefully consider factors such as concurrent connections and throughput so that at any instance of one firewall failure, the other firewall should have the capacity to handle the full load of the network.
- **Active/Active Full Mesh** In this, the firewall setup is enhanced a step further to ensure that no single point of failure occurs. As compared to an Active/Active setup, where we only ensure no firewall failure but do not consider the other components, such as switches and routers and their failure; in Active/Active Full Mesh, every firewall is wired twice to each network component, such as switches and routers. Optionally, all firewalls are also wired to the others by using dual HA links. This ensures that the alternate path for traffic flow is always available, in case there are simultaneous failures of switch, router, and firewall on a path. This is the maximum level of high availability that can be implemented.

Later in this section we will discuss the cabling requirement for each of these types of high availability.

Juniper NetScreen offers a scaled-down version of HA known as HA Lite. HA Lite provides only configuration synchronization and does not offer session or tunnel synchronization.

The Juniper NetScreen range of firewalls provides a wide variety of options to achieve an HA network, such as fall-back to dial-up on the small office home office (SOHO) appliances, device redundancy using midrange appliances, and the heavy duty features of the Enterprise and Carrier class systems. Table 12.1 shows a matrix of the different HA categories provided by the NetScreen firewalls.

Table 12.1 HA Feature Matrix

HA Feature	HA Category		
	SOHO	**NSRP-Lite**	**NSRP**
Fall-back to dial-up	Yes (some models)	No	No
Active/Passive setup	No	Yes	Yes
Active/Active setup	No	No	Yes
RTO synchronization	No	No	Yes

Table 12.2 shows HA features available on various Juniper NetScreen firewall models.

Table 12.2 Juniper NetScreen Models and HA Features

Models	HA
NetScreen HSC-5, HSC Plus	None
NetScreen 5 Series (except 5-XT)	HA Lite*
NetScreen 25	HA Lite
NetScreen 50	Active/Passive
NetScreen 204	Active/Passive Active/Active
NetScreen 208	Active/Passive Active/Active Active/Active Full Mesh
NetScreen 500, 500-GPRS, ISG Series, 5200/5400,	Active/Passive Active/Active Active/Active Full Mesh

Continued

Table 12.2 continued Juniper NetScreen Models and HA Features

Juniper NetScreen New Series	
SSG-5, SSG-20, SSG-140	Active/Passive
SSG-520	Active/Passive
SSG-550	Active/Passive Active/Active

NOTE

*HA Lite requires an extended license. HA Lite also provides configuration synchronization only. It does not provide session or tunnel synchronization.

Improving Availability Using NetScreen SOHO Appliances

Of all of the SOHO range of firewall appliances available (HSC, NS-5XT, NS-5GT, and the newer range SSG), all but the HSC support providing a secondary path for untrusted traffic. That is, should the normal link fail, a backup link can be activated and thereby the connectivity restored. This is a very useful feature, as anyone who has suffered from unplanned Internet service provider (ISP) outages can attest to. Two different ways to make sure that redundant ISP links are available is either by using two Ethernet interfaces or using one Ethernet interface and the serial interface as the backup. In the first scenario, the common setup has two ADSL modems or routers connected to separate ISPs, with one being the preferred provider. In the second scenario, the typical setup has an ADSL (asymmetric digital subscriber line) modem or router as the preferred link, and a modem connected to the serial interface providing dial-up access if needed.

When setting up redundant links, there are two main issues that must be specified: what will cause the backup link to activate, and how is it activated?

The event of deciding that the primary link is dead and the backup link should be activated is called a *failover*. The deciding factors for a failover include such things as physical link failure, virtual private network (VPN) failure, or an IP address becoming unreachable.

Once a failover is triggered, the backup link must be activated. How this happens depends on your setup. For example, you can have a second ADSL modem where the backup link is activated by setting up a Point-to-Point Protocol over Ethernet (PPPoE) session via that modem, or you can have a normal dial-up modem where the backup link is

activated by dialing a configured phone number followed by a Point-to-Point Protocol (PPP) login.

Failing Over between Interfaces

By default, you must manually initiate a failover. In many cases, this is sufficient and even recommended. You can initiate the failover from the CLI using the *exec failover force* command, or via the Web interface by going to Network | Untrust Failover | Force to Failover. To revert back to the primary untrust interface, use the *exec failover revert* command from the CLI, or go to Network | Untrust Failover | Force to Revert via the Web interface.

If you want to automate failovers, use the *set failover auto* command from the CLI or select Network | Untrust Failover | Automatic Failover | Apply via the Web interface.

Unless told otherwise, NetScreen firewalls determine whether to fail over to the backup interface by monitoring the status of the primary untrust port. If a link failure is detected, traffic is moved to the backup interface after a certain hold-down time has passed (30 seconds by default). This hold-down time is used to prevent rapid switching back and forth between the interfaces. Delaying the failover for a short time gives the interfaces a chance to stabilize. A failover is a serious action that should not be made without good cause. If you find that the 30-second default hold-down time is inappropriate in your situation, you can adjust it with the *set failover hold-down N* command, where *N* is the number of seconds to wait before initiating the failover.

The monitoring of the untrust port only detects a link failure between the NetScreen and the modem or router it is connected to; it does not detect failures beyond that modem or router. Hence, if you have an ADSL modem connected to your NetScreen and the digital subscriber line (DSL) service is interrupted, it will not be detected by port monitoring on the firewall. To handle these types of problems, you must configure other types of monitoring such as VPN monitoring or IP tracking.

Using Dual Untrust Interfaces to Provide Redundancy

Using dual untrust interfaces is suited to any scenario where the untrusted network, generally the Internet, can be reached via two different paths with each path providing an Ethernet connection. This can be via:

- ADSL modem (using PPPoE on the NetScreen)
- ADSL router
- Ethernet Direct (using PPPoE on the NetScreen)
- Cable Ethernet (using PPPoE on the NetScreen)

Combinations of the preceding, or any service that uses PPPoE to provide access, can be used to establish the redundancy desired in this scenario.

To be able to use dual untrust interfaces, the port mode must be set to either *dual-untrust* or *combined*. The combined mode is available on the NS-5XT and 5GT models, but functions identically to the two untrust ports. For the remainder of this chapter, *dual-untrust* is used to refer to either of these two modes.

WARNING

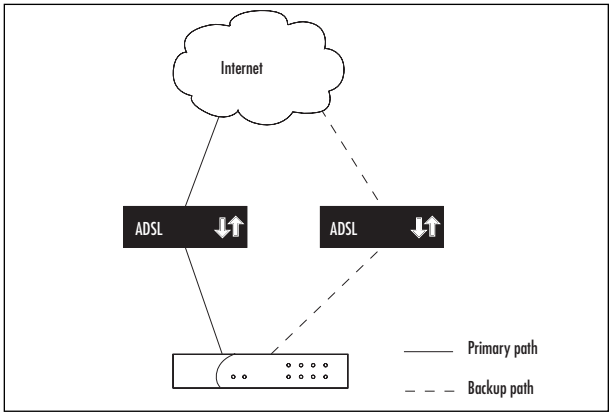
If you change the port mode, the entire configuration will be erased. It is a good idea to back up the existing configuration before you continue. Also, because the configuration is lost (including assigned IP addresses), it is easier to use the CLI via the console port than the Web interface.

Bringing the NetScreen into *dual-untrust* mode is done using the *exec port-mode dual-untrust* command from the CLI, or going to Configuration | Port Mode | Port Mode via the Web interface. Any previous configuration will be erased and must therefore be manually reentered after the firewall has rebooted. If you do not know what mode the firewall is in, you can see it by using the *get system* command from the CLI or by going to Configuration | Port Mode via the Web interface.

Example: Configuration for Dual ADSL Modems

Once in *dual-untrust* mode, you can add the necessary configuration needed for the two separate untrusted paths. For example, imagine a scenario where the preferred path is via one ADSL modem, and the backup path is via a different ADSL modem (or other service relying on PPPoE authentication), as depicted in Figure 12.1.

Figure 12.1 Redundant ADSL Internet Connections



From the CLI:

```
set pppoe name "primary-adsl" username user1 password abc123
set pppoe name "primary-adsl" interface ethernet3
set pppoe name "primary-adsl" clear-on-disconnect
set pppoe name "primary-adsl" idle-interval 0
set pppoe name "primary-adsl" auto-connect 20
set pppoe name "backup-adsl" username user2 password bcd234
set pppoe name "backup-adsl" interface ethernet2
set pppoe name "backup-adsl" clear-on-disconnect
set pppoe name "backup-adsl" idle-interval 0
set pppoe name "backup-adsl" auto-connect 20
set failover auto
```

From the Web interface:

1. Go to **Network** | **PPPoE** | **New**.
2. Name this instance **primary-adsl**.
3. Bind this instance to **ethernet3**.
4. Enter the ISP's username and password.
5. Enable the **Clear IP on Disconnect** option.
6. Enable **Auto-connect** and specify **20 seconds**.
7. Click **OK** to save.
8. Return to **Network** | **PPPoE** | **New**.
9. Name this instance **backup-adsl**.
10. Bind this instance to interface **ethernet2**.
11. Enter the ISP username and password.
12. Enable **Auto-connect** and specify **20 seconds**.
13. Click **OK** to save.
14. Go to **Network** | **Untrust Failover** | **Automatic Failover** and click **OK**.

The *clear-on-disconnect* option is used to ensure that any old interface IP address or default gateway is removed when the PPPoE session goes down. This ensures that packets are routed correctly after a failover instead of being sent to the now unreachable gateway.

The *idle-interval* option allows you to automatically disconnect the PPPoE session after a certain period of inactivity. In this case, we do not want to do that. By setting it to zero, this feature is disabled and the PPPoE session will not be dropped due to inactivity.

The *auto-connect* option enables the NetScreen to attempt to reconnect to the PPPoE session if it is dropped. If this is not set, the connection must be manually brought up via the *exec pppoe connect* command, or by power-cycling the NetScreen. Clearly, this is not a desirable feature in our case; thus, we use auto-connect.

Another interesting aspect of this option is how it interacts with the automatic failover (if enabled). To give the primary untrust interface a good chance of recovering before a failover is initiated, set the auto-connect value lower than the failover hold-down setting, which is 30 seconds by default. This way, the PPPoE connection on the primary interface is retried before the failover is triggered. This might be enough to prevent a failover altogether depending on the reason for the failure of the PPPoE session.

Example: Advanced Configuration for ADSL Modem Plus ADSL Router

For this example, imagine a setup where the primary link is via an ADSL modem and the backup is via an ADSL router to a different ISP. They are used in this order to get the IP address assigned directly to the NetScreen. This way any NAT can take place on the NetScreen. The NetScreen provides more power and flexibility in this area than most ADSL routers commonly do, so it is the sensible thing to do.

To make matters a bit more interesting in this example, also assume that the primary ISP has assigned a static IP address to us (1.1.1.1) that should always be used, even if the remote end attempts to assign a different IP address during the PPP negotiation. Any decent ISP should be able to assign static IP addresses via PPPoE itself, but for the sake of this exercise, let's assume that this ISP cannot. The ISP's equipment is really bad at responding to the Link Control Protocol (LCP) Echo requests that are used to verify that the link is up. Unless more conservative timings are used on our end, the link will keep getting dropped despite actually being up and working fine. Furthermore, this ISP supports only Password Authentication Protocol (PAP); the considerably more secure Challenge Handshake Authentication Protocol (CHAP) is not supported.

From the CLI:

```
# public IP address as assigned by the primary ISP
set interface ethernet3 ip 1.1.1.1/24
# private IP address used between the NetScreen and the backup DSL router
# the DSL router gets the public IP address via PPPoE and then NATs traffic
set interface ethernet2 ip 172.16.32.2/30
set pppoe name "adsl" username user1 password abc123
set pppoe name "adsl" interface ethernet3
set pppoe name "adsl" static-ip
set pppoe name "adsl" authentication PAP
set pppoe name "adsl" clear-on-disconnect
set pppoe name "adsl" auto-connect 20
set pppoe name "adsl" idle-interval 0
set pppoe ppp lcp-echo-retries 20
set pppoe ppp lcp-echo-timeout 600
# set a default gateway for the backup path, when in use
```

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet2 gateway 172.16.32.1
set failover auto
```

NOTE

Comments can be used in external NetScreen configuration files. They are removed when the configuration is loaded into the NetScreen, but can be very useful for documenting configurations in a noninterfering manner. Full-line comments start with a hash mark followed by a space (#). Half-line comments start with a space followed by a hash mark followed by a space (#).

From the Web Interface:

1. Go to **Network** | **Interfaces** | **Edit** | **ethernet3**.
2. Enter **1.1.1.1/24** for the IP address.
3. Click **OK**.
4. Go to **Network** | **Interfaces** | **Edit** | **ethernet2**.
5. Enter **172.16.32.2/30** as the IP Address and Netmask.
6. Click **OK** to save.
7. Go to **Network** | **PPPoE** | **New**.
8. Name this instance **adsl**.
9. Enter the username and password for the ISP.
10. Bind it to interface **ethernet3**.
11. Select **PAP** as the Authentication type.
12. Check the **Static IP** option.
13. Select the **Clear on Disconnect** option.
14. Set **Auto-connect** to **20 seconds**.
15. Click **OK** to save.
16. Go to **Network** | **PPPoE** | **PPP**.
17. Set LCP Echo Retries to **20**.
18. Set LCP Echo Timeout to **600**.
19. Click **OK**.
20. Go to **Network** | **Routing** | **Routing entries**.
21. Select **trust-vr**.

22. Click **New** to add a new route.
23. Enter **0.0.0.0/0** as the IP Address and Netmask.
24. Select **ethernet2** as the gateway interface.
25. Enter **172.16.32.1** as the gateway IP address.
26. Click **OK** to save the new route.
27. Go to **Network** | **Untrust Failover** | **Automatic Failover** and click **OK**.

Falling Back to Dial-Up

If you do not have two Ethernet-capable connections to the Internet, you can use a dial-up connection as the backup path. The modem is connected to the NetScreen's serial port, and the dial-up is configured using the modem settings and the serial interface. It is also possible to specify multiple phone numbers to dial in sequence, in effect giving you more fall-back possibilities; if the first ISP does not answer, the next in line can be dialed.

NOTE

The serial interface is only available when *trust-untrust* or *home-work* mode is used as the port mode. It is not possible to use the serial interface in any other mode.

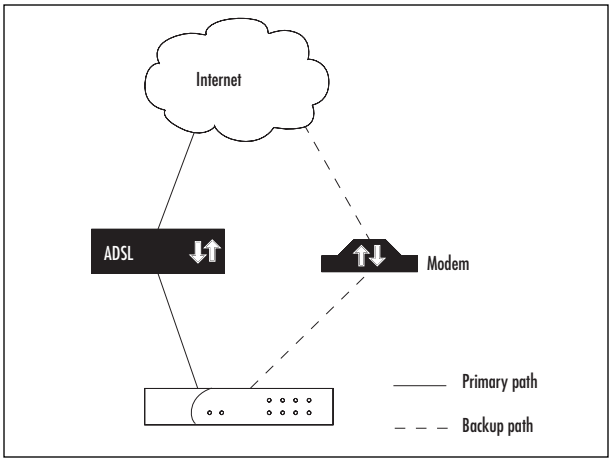
Following are the steps needed to configure this:

1. Move the serial interface to the Untrust zone to enable it.
2. Configure the modem settings.
3. Enter the ISP account details.
4. If using the CLI, add a default route for the serial interface (done automatically when using the Web interface).
5. Enable **Automatic Failover** (if desired).

Example: A Simple Backup Dial-up Configuration

Using an unspecified generic modem, we configure a backup dial-up setup for an ISP that has a single dial-up number. Figure 12.2 shows the setup.

Figure 12.2 Fall-Back to the Dial-Up Internet Connection



From the CLI:

```
set interface serial zone untrust
set modem settings "generic" init-strings AT&FS7=255S32=6
set modem settings "generic" active
set modem isp "myisp" account login user1 password abc123
set modem isp "myisp" primary-number 5550123
set modem isp "myisp" priority 1
set vrouter trust-vr route 0.0.0.0/0 interface serial
set failover auto
```

From the Web interface:

1. Go to **Network** | **Interfaces** and click **Edit** for the serial interface.
2. Select the **Untrust** zone.
3. Click **OK**.
4. Return to **Network** | **Interfaces** | **Serial Edit** and click **Modem**.
5. Name the modem **generic**.
6. Specify an Init-string of **AT&FS7=255S32=6**.
7. Select it as **Active**.
8. Click **OK** when done.
9. Go to **Network** | **Interfaces** | **Serial Edit** | **ISP**.
10. Name the ISP **myisp**.
11. Provide the **Login Name** and **Password** details as well as the **Primary telephone number**.

12. Set the Priority to **1**.
13. Click **OK**.
14. Go to **Network | Untrust Failover | Automatic Failover** and click **Apply**.

Example: An Advanced Backup Dial-up Configuration

To demonstrate more of the flexibility provided with the dial-up ability, let's consider a more complex example. Here, we configure support for two different ISPs, each with dual dial-up numbers. A Rockwell-based 56K modem will primarily be used; the settings on an old US Robotics 28800 are configured as a backup. Since we are more concerned with network availability than the phone bill, we will disconnect the call after 45 minutes of inactivity (inactivity being defined as a period where no packets are permitted by a policy through the firewall). We also want to dial rapidly until we successfully connect to an ISP, so the dial interval is lowered to four seconds (the default is 10 seconds). Be aware that you need to allow for sufficient time for the phone line to actually disconnect before the next dial attempt is made. Also, since we will have four different numbers that we can potentially connect through, there is little to be gained by retrying the same number many times before continuing to the next number (or ISP if there was no answer on either of the first ISP's numbers). Setting the number of retry attempts to one causes the NetScreen to only attempt to redial each number once before continuing on to the next.

From the CLI:

```
set interface serial zone untrust
set modem settings "rockwell" init-strings AT&FV1W1X4&C1&D3&K3&N3%C3S7=60
set modem settings "usr28800" init-strings AT&F&A3&B1&H1&R2&D0&C1X4S7=90
set modem settings "rockwell" active
set modem idle-time 45
set modem interval 4
set modem retry 1
set modem isp "isp1" account login user1 password abc123
set modem isp "isp1" primary-number 5550123 alternative-number 5550124
set modem isp "isp1" priority 1
set modem isp "isp2" account login user2 password bcd234
set modem isp "isp2" primary-number 5550198 alternative-number 5550199
set modem isp "isp2" priority 2
set vrouter trust-vr route 0.0.0.0/0 interface serial
set failover auto
```

From the Web interface:

1. Go to **Network** | **Interfaces** and click **Edit** for the serial interface.
2. Select the **Untrust** zone.

3. Click **OK**.
4. Still under **Network** | **Interfaces** | **Serial Edit**, click **Modem**.
5. Name the modem **rockwell**.
6. Specify an Init-string of **AT&FV1W1X4&C1&D3&K3&N3%C3S7=60**.
7. Select it as **Active**.
8. Set the Idle-time to **45 minutes**.
9. Set the Retry Interval to **4 seconds**.
10. Specify the number of Retries as **1**.
11. Add a second modem named **usr28800**.
12. Set the Init-string to **AT&F&A3&B1&H1&R2&D0&C1X4S7=90**.
13. Click **OK** when done.
14. Go to **Network** | **Interfaces** | **Serial Edit** | **ISP**.
15. Name the ISP **isp1**.
16. Provide the **Login Name** and **Password** details as well as the dial-up **Primary** and **Alternative telephone numbers**.
17. Set the Priority to **1** (the highest priority).
18. Click **OK**.
19. Add a second ISP named **isp2**.
20. Enter the **Login Name** and **Password** for this ISP, followed by the **Primary** and **Alternative phone numbers**.
21. Specify a priority of **2**.
22. Click **OK** to save these settings.
23. Navigate to **Network** | **Untrust Failover** | **Automatic Failover** and click **Apply**.

NOTE

You can send AT commands to the attached modem using the *exec modem command* command in the CLI (for example, *exec modem command ATZ* to reset the modem).

It is also possible to examine the state of the modem with the *get modem state* command (for instance, is it waiting, dialing, connected, and so forth).

Restricting Policies to a Subset When Using the Serial Interface

Since a dial-up link inherently has less bandwidth and more latency than a DSL line or similar, it is easy to end up with a seriously congested uplink after having failed over to the dial-up path. Fortunately, the wise NetScreen designers foresaw this problem and provided a means to avoid it. By tagging individual policies, it is possible to choose which policies should or should not be active while traffic is moving via the backup serial interface. Which policies you choose to disable, if any, are entirely up to you, but some common examples include File Transfer protocol (FTP), Voice over IP (VoIP), and audio-streaming services, since they are capable of chewing up a lot of bandwidth very easily.

Example: Marking FTP as Not Allowed When Using the Serial Interface

Assume we want to add a policy allowing FTP traffic to move from the Trust to the Untrust zone, which is automatically disabled when traffic is failed over to the serial interface.

From the CLI:

```
set policy from trust to untrust any any ftp permit no-session-backup
```

From the Web interface:

1. Go to **Policies** and create a new policy from the Trust zone to the Untrust zone.
2. Select **Any** as the source and destination address.
3. Select **FTP** as the service.
4. Set **Permit** as the action for the policy.
5. Click **Advanced**.
6. Deselect the **Valid For Serial** option, and click **Return**.
7. Click **OK** to save the new policy.

As can be seen, the keyword "no-session-backup" is added to the policy to indicate that the policy should not be active when the serial interface is in use. Note that this has no impact if you are using *dual-untrust* mode as the port mode, since the serial interface is not used in that case.

Using IP Tracking to Determine Failover

Relying on link monitoring is insufficient, especially when using tunneled protocols such as PPPoE. Usually when an outage occurs, it is not because the Ethernet cable between the NetScreen and the ADSL modem failed, but because the PPPoE connection could not be established. If using a default configuration, this scenario would not result in a failover because the Ethernet link is working and that is all that is being monitored. Nevertheless,

there is no disputing the fact that the Internet connectivity is nonexistent in this case, and that a failover should happen.

To address this problem, NetScreen has introduced the concept of *IP tracking*. IP tracking works by regularly pinging specified IP addresses; if they do not reply within the defined bounds, the link is considered down and a failover can be triggered. Note that I say, "can be triggered" not "will be triggered." That is because it is possible to have IP tracking configured and in use without actually having it cause failovers. The value of doing so is questionable, however.

IP tracking is considered an interface-level setting. This means that full routing is not done for the ping packets that are sent; they are always sent out via the interface that the IP tracking is configured on. You have the option of either specifying the next hop (gateway) to use for the pings explicitly, or letting the NetScreen determine it automatically. In an environment where the IP address and the default gateway are dynamically assigned, it is not feasible to explicitly set the gateway address.

The IP addresses that can be tracked fall into two categories: those specified explicitly, and those referred to by function. The latter includes only one type: the default gateway for the interface. This is very useful when the interface dynamically receives an IP address as well as the default gateway, as is the case with PPPoE and Dynamic Host Control Protocol (DHCP). In situations where you do not know the IP address you need to ping beforehand, being able to refer to it as a dynamic entity and let the NetScreen worry about the specifics of which IP it refers to is very handy.

Just being able to ping a few IP addresses and fail over if any single one of them does not respond is not particularly helpful, and NetScreen has recognized this. A great deal of flexibility is provided by the *weighting* system offered by the IP tracking. Fine-grained control is possible by assigning different *weights* (importance) to different IP addresses and setting a failover *threshold*. For example, you could say something like, "If both Web Server A and Web Server B or Mail Server M are unreachable, then fail over." It is generally a good idea to keep it as simple as possible.

The weighting system consists of two main components: the individual weights for the tracked IP addresses and the failover threshold value. If at any point the combined sum of the failed IP addresses weights equal or exceed the threshold, failover can be triggered. An IP address is considered to have failed if a certain number of pings have gone unanswered. It is possible to configure both the frequency of the pings and the number of missing replies needed for it to be classified as "failed."

Example: Tracking the Default Gateway

This example shows what is perhaps the most common scenario; tracking a dynamically assigned default gateway on interface ethernet3. Automatic failover is enabled and set to use IP tracking as the determining factor.

From the CLI:

```
set interface ethernet3 track-ip
```

```
set interface ethernet3 track-ip dynamic
set failover auto
set failover type track-ip
```

From the Web interface:

1. Go to **Network** | **Interfaces** | **Edit** | **Track IP**.
2. Select **Enable Track IP** to enable IP tracking on this interface.
3. Press **Apply**.
4. Go to **Network** | **Interfaces** | **ethernet3 Edit** | **Track IP**.
5. Select the **Dynamic** option to track the default gateway.
6. Select **Add** to add dynamic monitoring to the list.
7. Go to **Network** | **Untrust Failover** and enable the **Automatic Failover** option.
8. Select **IP Tracking** as the Failover Type.
9. Click **Apply** to confirm these settings.

Example: A More Complex IP Tracking Scenario

To expand on the previous example, consider a scenario where we always want to be able to reach at least one of two Web servers, as well as our mail server. We also know that due to the amount of junk mail received these days, the mail server might be slow to respond to pings, so we make adjustments for this, pinging once every five seconds and allowing for six missed responses (the default is to ping every second and only allow three missed replies).

For this exercise, we assume that the Web server's IP addresses are 1.1.1.1 and 1.1.1.2, and that the mail server resides at 2.2.2.2. The weights are allocated thus: 1 for each of the Web servers, 2 (or greater) for the mail-server, and a failover threshold of 2. Thus, if both Web servers fail, the sum of the weight for the failed IP addresses will reach 2, which is the failover threshold.

From the CLI:

```
set interface ethernet3 track-ip
set interface ethernet3 track-ip 1.1.1.1 weight 1
set interface ethernet3 track-ip 1.1.1.2 weight 1
set interface ethernet3 track-ip 2.2.2.2 interval 5 threshold 6 weight 2
set interface ethernet3 track-ip threshold 2
set failover auto
set failover type track-ip
```

From the Web interface:

1. Go to **Network** | **Interfaces** | **ethernet3 Edit** | **Track IP**.
2. Select **Enable Track IP** to enable **IP tracking** on this interface.
3. Set the Failover Threshold to **2**.

4. Click **OK**.
5. Return to **Network** | **Interfaces** | **ethernet3 Edit** | **Track IP**.
6. Add IP address **1.1.1.1** with a Weight of **1** and click **Apply**.
7. Add IP address **1.1.1.2** with a Weight of **1** and click **Apply**.
8. Add IP address **2.2.2.2**, with an Interval of **5**, a Threshold of **6**, and a Weight of **2** and click **Apply**.
9. Go to **Network** | **Untrust Failover** and enable the **Automatic Failover** option.
10. Select **IP tracking** as the Failover Type.
11. Click **Apply** to confirm these settings.

Monitoring VPNs to Determine Failover

If IP tracking does not provide you with sufficient control and you are reliant on VPN tunnels, you will be pleased to know that the status of VPN tunnels can also be used as a basis for initiating failovers. This feature works similar to IP tracking in that it adds up the weights for the VPN tunnels that are down, and if they reach the failover threshold, a failover is initiated. There are some differences, such as the threshold is always 100 (think of it as 100 percent), and that in addition to the *working* and *failed* states, there is also a *halfway failed* state, in which half the weight of the tunnel is counted towards the threshold.

Such is the case for VPN tunnels that are in *inactive*, *ready*, and *indeterminate* states. For example, if the VPN weight is 60, it would be counted as 30 when that VPN tunnel was in an *inactive* state, and hence the NetScreen would be 30 percent towards failing over to the backup interface (assuming automatic failover is enabled).

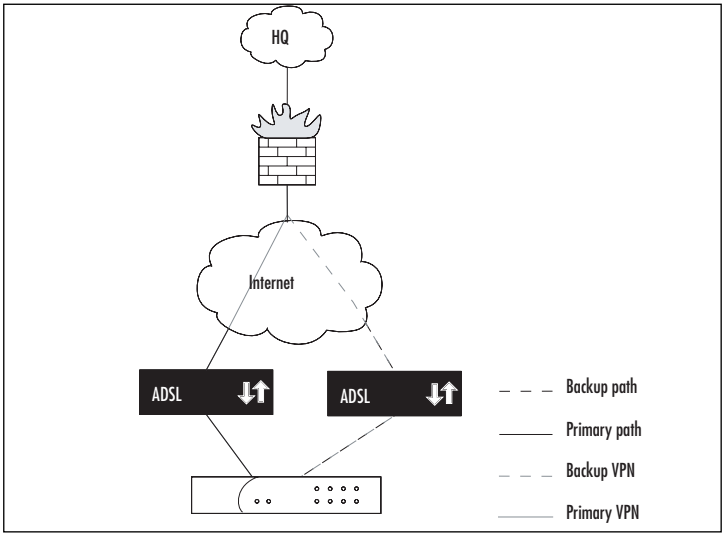
Not all VPNs are monitored for failure. You have to enable monitoring on the VPN tunnels you wish to monitor, which enables you to monitor only the most important ones.

Be aware that once a failover is triggered due to failed VPN tunnels, it *will not revert back* to the primary interface by default. Once the failover has taken place, no attempts will be made to reestablish the failed VPN tunnels, and therefore the failover threshold will stay exceeded until you manually intervene. To avoid this, you must enable *re-keying* as well as monitoring. By doing this, the NetScreen will attempt to regularly restore the failed VPN tunnels and, if successful (and the failure value falls below the threshold), will revert traffic back to the primary interface. This is not enabled by default because if there were a continual attempt to bring the tunnels up, the second path would be open all the time, which is generally not desired. It is safer to have the firewall administrator explicitly enable re-keying where wanted, instead of hoping they will remember to disable it where appropriate.

Example: Monitoring One VPN Tunnel, with Fall-Back to a Second Unmonitored Tunnel

In this example, under normal circumstances a single VPN tunnel is up and in use, which goes to corporate headquarters (HQ) and is therefore very important. If it fails for any reason, the NetScreen will fail over to the backup interface where a second VPN tunnel is also ready to connect to HQ. To achieve this, we use two tunnel interfaces and add routes to HQ through both of those tunnel interfaces. We also enable re-keying on the primary VPN so that the NetScreen will revert back to the primary interface once that VPN tunnel is up again. Figure 12.3 shows the setup for this scenario.

Figure 12.3 Redundant VPN Tunnels to HQ



From the CLI:

```
# prepare tunnel interfaces
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
set interface tunnel.2 zone untrust
set interface tunnel.2 ip unnumbered interface ethernet2
# use two IKE gateways, bound to ethernet3 and ethernet2, respectively
set ike gateway hq-gw-eth3 ip 1.2.3.4 aggressive outgoing-interface
  ethernet3 preshare key123 sec-level standard
set ike gateway hq-gw-eth2 ip 1.2.3.4 aggressive outgoing-interface
  ethernet2 preshare key123 sec-level standard
# create the actual VPNs
set vpn to-hq-primary gateway hq-gw-eth3 sec-level standard
```

```
set vpn to-hq-primary bind interface tunnel.1
set vpn to-hq-primary monitor rekey
set vpn to-hq-primary failover-weight 100
set vpn to-hq-backup gateway hq-gw-eth2 sec-level standard
set vpn to-hq-backup bind interface tunnel.2
# routes to corporate HQ, via VPN tunnels
set vrouter trust-vr route 10.10.0.0/16 interface tunnel.1
set vrouter trust-vr route 10.10.0.0/16 interface tunnel.2
# enable automatic failover, based on VPN monitoring
set failover auto
set failover type tunnel-if
```

From the Web Interface:

1. Go to **Network** | **Interfaces** | **New Tunnel IF**.
2. Name the new interface **tunnel.1**.
3. Set the Zone (VR) to **Untrust** (trust-vr).
4. Select **Unnumbered** and set the interface to **ethernet3**.
5. Click **OK**.
6. Return to **Network** | **Interfaces** | **New Tunnel IF**.
7. Name this interface **tunnel.2**.
8. Set the Zone (VR) to **Untrust** (trust-vr).
9. Select **Unnumbered** and specify **ethernet2**.
10. Click **OK**.
11. Go to **VPNs** | **Autokey Advanced** | **Gateway** | **New**.
12. Name this interface **hq-gw-eth3**.
13. Choose **standard** as the Security Level.
14. Select **Static IP Address** as the Remote Gateway Type.
15. Enter **1.2.3.4** as the Address/Hostname.
16. Specify **key123** as the Preshared Key.
17. Select **ethernet3** as the Outgoing Interface.
18. Click **OK**.
19. Return to **VPNs** | **Autokey Advanced** | **Gateway** | **New**.
20. Name this interface **hq-gw-eth2**.
21. Choose **standard** as the Security Level.
22. Select **Static IP Address** as the Remote Gateway Type.

23. Enter **1.2.3.4** as the Address/Hostname.
24. Enter **key123** as the Preshared key.
25. Select **ethernet2** as the Interface.
26. Click **OK** to save.
27. Go to **VPNs** | **Autokey IKE** | **New** to create a new VPN.
28. Name this VPN **to-hq-primary**.
29. Choose **standard** as the Security Level.
30. Select **Predefined** as the Remote Gateway and **hq-gw-eth3** as the actual gateway.
31. Click **Advanced** to enter additional settings for this VPN.
32. Select **tunnel.1** as the Interface.
33. Select **VPN Monitor** to enable monitoring of this VPN.
34. Enable the **Rekey** option.
35. Click **Return**, followed by **OK** to save these settings.
36. Return to **VPNs** | **Autokey IKE** | **New** to create a new VPN.
37. Name this VPN **to-hq-backup**.
38. Choose **standard** as the Security Level.
39. Select **Predefined** as the Remote Gateway, and choose **hq-gw-eth2** from the list.
40. Click **Advanced**.
41. Bind this VPN to **tunnel.2**.
42. Click **Return** and then **OK** to save.
43. Go to **Network** | **Routing** | **Routing Entries** | **trust-vr** | **New** to add a new route.
44. Enter **10.10.0.0/16** as the Network Address/Netmask.
45. Select **tunnel.1** as the Interface.
46. Set the Gateway IP Address to **0.0.0.0**.
47. Click **OK**.
48. Return to **Network** | **Routing** | **Routing Entries** | **trust-vr** | **New**.
49. Enter **10.10.0.0/16** as the Network Address/Netmask.
50. Select **tunnel.2** as the Interface.
51. Set the Gateway IP Address to **0.0.0.0**.
52. Click **OK**.

53. Go to **Network** | **Untrust Failover**.
54. Set the Failover Type to **Tunnel Interface**.
55. Enable **Automatic Failover**.
56. Click **Apply**.
57. Return to **Network** | **Untrust Failover** and click **Edit Weights**.
58. Set the Weight to **100** for the VPN named **to-hq-primary**, if not done already.
59. Click **OK**.

Introducing the NetScreen Redundancy Protocol

It is time to focus on the more advanced features of NetScreen. Because the standard features are not strong enough for the more demanding environments, the mid- to high-range NetScreens provide support for the NetScreen Redundancy Protocol (NSRP). This protocol is the heart of all of the HA options covered here. NSRP has been available for several years, and was originally referred to as *HA*. With the introduction of NSRP version 2 in the Screen OS 3.1 versions, this changed. HA setups are now commonly referred to as NSRP setups, or just "running NSRP."

NSRP is the protocol that redundant NetScreen devices use to talk to each other when running in various HA configurations. It is the language that allows them to exchange state information and make decisions. Before we detail the specifics of what type of information is exchanged over NSRP, we need to cover a bit more theory.

One of the main goals of HA is to have multiple redundant systems, where a second system can take over in case the first one fails. This is commonly achieved by duplicating the hardware. As with the NetScreen firewalls, any HA setup using NSRP implies at least two firewalls of the same model are working together. This group of firewalls is called an *NSRP cluster*, or simply, a *cluster* (see Figure 12.4). While, conceptually, NSRP has been engineered to allow for future expansion into clusters containing more than two members, this is not yet implemented and may never be.

Virtualizing the Firewall

To minimize the amount of downtime caused when the first system in a cluster fails, it is important to ensure that the second system knows precisely what the first system is doing, so that it can pick up without any interruption. In effect, what you want to do is shift the entire running firewall onto new hardware. When looked at this way, it makes sense to turn the actual firewall into a virtual firewall that has some hardware associated with it. This is precisely what is done in NetScreen firewalls. When NSRP is enabled, a Virtual Security Device (VSD) is created, and the configuration for the physical interfaces changes to apply to

virtual interfaces called Virtual Security Interfaces (VSI). The fact that these virtual interfaces are in turn associated with actual hardware is not important; all of the configurations are done on the VSI. NSRP then takes care of associating the VSI to the correct physical interface (see Figure 12.5).

Figure 12.4 A NetScreen Cluster

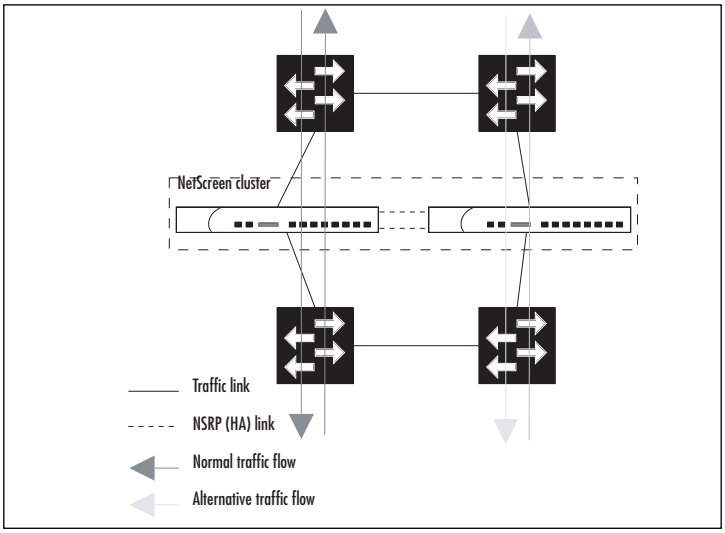
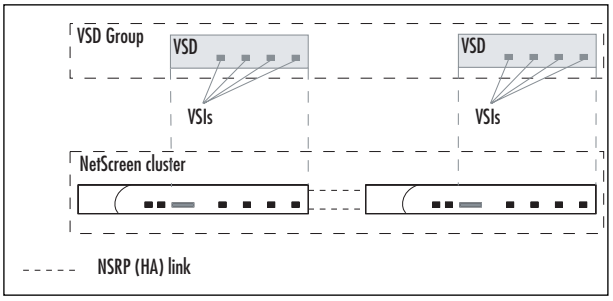


Figure 12.5 VSDs and VSIs



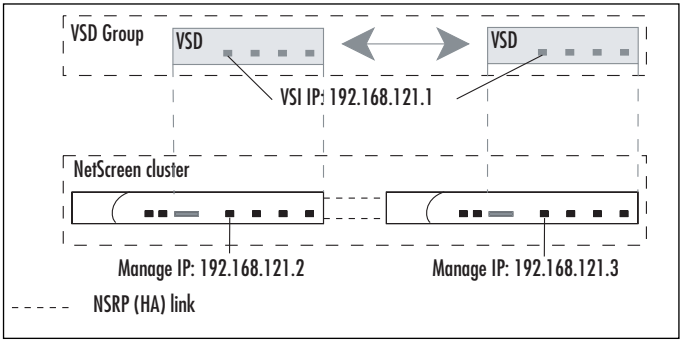
By abstracting the firewall in this manner, it becomes relatively easy to move the firewall between different hardware units as necessary. It also becomes possible to have more than one VSD per physical firewall. A VSD is not a stand-alone entity, but rather it is always part of a *VSD group*, which spans both of the NetScreens in the cluster. There is one VSD per VSD group on each cluster member, and the VSD configuration is identical everywhere. This sets the scene for when a failover is needed, since the configuration is already set up and ready to go. Note that the VSD only acts as a container for the VSIs. Other configuration items, such as policies and routing, apply across all VSDs on a NetScreen.

Within a VSD group, one VSD is the designated *master VSD*. This VSD is the currently active VSD that is processing and forwarding traffic. The other VSD in the VSD group is the *backup*, which is located on the other NetScreen. The backup VSD is not processing traffic, which means that only one of the firewalls is active and processing traffic at any given time. This is known as an *Active/Passive setup*.

It is important to note that IP addresses assigned to a VSI follow the master VSD. This is slightly different from how Virtual Router Redundancy Protocol (VRRP) works. In VRRP, the backup unit has its own IP address and simply acquires the primary IP address upon failover. With NSRP, there is only one interface IP address that floats between the NetScreens as necessary.

The *manage-ip* settings stay bound to the physical interface and do not follow the VSD. It would not be very useful if the IP addresses were also moved across to the active firewall since that would render the backup firewall unreachable for management purposes. Thus, when the backup firewall becomes the master, it already has the manage IP address and then simply adds the VSI address (see Figure 12.6).

Figure 12.6 VSI IP Addresses vs. Managed IP Addresses



When talking about NSRP clusters, the opposite of a VSI is a *local interface*. A local interface is one that is not tied to a VSD, and therefore will not move across in case of a failover. There are a few select cases when this can be useful, which are covered later. For now, all you need to be aware of is that it is possible, though uncommon, to have local interfaces that do not participate in the VSD.

Understanding NSRP States

As mentioned, the fundamental concept of NSRP is duplicating hardware—to be able to move the firewall functionality around as necessary using VSDs. As a consequence of this, at any given time each VSD is in one of six states, which determines the current role of the VSD. The possible states are

- Master
- Primary Backup
- Backup
- Initial
- Ineligible
- Inoperable

Understanding which state is used for what purpose is central to monitoring and controlling your NSRP cluster. Instead of simply explaining what each state is, let's look at the order in which the VSD transitions between the states.

When a VSD is first created, either due to a reboot or a configuration change, it is put in the *initial* state. While in this state, the VSD learns which other devices are participating in the VSD group, synchronizes that state with the other VSDs if needed, and possibly partakes in the election process for which VSD should become the master.

From the *initial* state, the VSD can move into either the *master* or *backup* state. If it wins the election process, this VSD takes on the task of processing traffic. If it does not win, it transitions into the *backup* state.

The election process used to determine the master VSD is reasonably straightforward. First, if there is no other VSD available, this VSD automatically wins the election. Second, if two VSDs are starting up at the same time, the winner is determined based on the configured priorities (*set nsrp vsd-group id X priority N*). The unit with the lowest priority value is the preferred VSD. If both VSDs have the same priority, or the priority is not configured, the VSD with the lowest Media Access Control (MAC) address wins.

Normally, an election is only held if there is no master VSD in the VSD group. However, if the starting VSD has preemption enabled (*set nsrp vsd-group X preempt*), it can force an election, which it would probably win due to having a better priority than the old master VSD.

A VSD in the *backup* state checks to see if there is already a primary backup VSD, and if there isn't, makes itself the primary backup for the VSD group. As the primary backup, it is responsible for taking over the traffic processing should the master fail or step down. From the primary *backup* state there are generally two directions the VSD can take; it either ends up promoted to master due to the old master VSD disappearing, or it goes into the *inoperable* state.

A VSD puts itself into the *inoperable* state if it detects a failure that would prevent it from processing traffic. If this VSD were the master, any failure that resulted in a failover would result in this VSD becoming *inoperable*. In this state, the VSD does not participate in elections for the position of master VSD; however, it does continue to check for the failure condition. If that condition is remedied, as can be the case if the failure was caused by a monitored interface going down, which was then subsequently brought back up, the VSD will progress from the *inoperable* state back into the *initial* state again.

The *ineligible* state is only entered by manual intervention. It is the *administratively down* state of the VSD. If for any reason you want to prevent the VSD from participating in the master election, thereby preventing it from processing traffic, you can put the VSD into the *ineligible* state by using the *set nsrp vsd-group id X mode ineligible* command. The VSD group stays in that state until you use the corresponding *unset* command, or the NetScreen is rebooted without having saved the configuration (the *ineligible* state can be kept across reboots if you save the configuration after entering the command).

This explains the various NSRP states that a VSD can be in. If you are confident in this knowledge, you will have no problem understanding what the VSDs in your cluster are doing.

The Value of Dual HA Links

Only a single link is needed for the NSRP traffic in an NSRP cluster; however, there are significant advantages to using dual HA links. To better understand these advantages, let's examine what types of packets are exchanged between the NetScreens in an NSRP cluster.

Two categories of packets are exchanged: *control messages* and *data packets*. Control messages are what enable NSRP to function. Data packets are simply normal user data packets that are passed on from one firewall to the other for processing. This packet forwarding occurs in certain cases, but it is not the norm and should be avoided if possible. This can occur if an Active/Active setup is used.

The control messages consist of various heartbeats and synchronization messages, such as physical link probes, VSD state information, and session synchronization information. These messages are always sent via HA link #1. If you have dual HA interfaces, #1 has the lower interface number. For example, if ethernet7 and ethernet8 are bound to the HA zone, ethernet7 would carry the control messages unless it becomes unavailable, in which case the control messages would be sent on ethernet8 instead.

Due to the bandwidth requirements of the data-forwarding function, data forwarding is not always available. Table 12.3 shows when and which HA link is used for data or control messages in the different scenarios.

Table 12.3 NSRP Control and Data Messages HA Link Usage

Message Type	Interfaces			
	Single 100MB	**Dual 100MB**	**Single 1GB**	**Dual 1GB**
Control	Yes (#1)	Yes (#1)	Yes (#1)	Yes (#1)
Data	No	Yes (#2)	Yes (#1)	Yes (#2)

On NetScreen models that do not contain dual dedicated HA interfaces, one or two interfaces can be bound to the HA zone to enable them to be used as HA links. This is done by assigning the interface to the HA zone just as you would with any other zone.

NOTE

If you find yourself in a situation where you do not have any unused interfaces on the firewall and you want to turn it into an NSRP cluster, you can still do so. NetScreens have the option of allowing NSRP traffic to coexist with normal traffic on an interface. You will need to use an interface that is connected to a switch that provides a layer 2 broadcast domain that is common to both firewalls. By using the *set nsrp interface* command, you can direct NSRP to send the control messages on that particular interface. Be aware that the NSRP packets can use up a significant amount of bandwidth. Also, remember that in this scenario you should enable authentication and encryption of the NSRP traffic.

Building an NSRP Cluster

Before you can configure the NetScreens to be used in your NSRP cluster, you must do the cabling. Since there are a few options available, this section covers the advantages and disadvantages of the most common ones. What is presented here is not an exhaustive list, but it should be enough for you to properly evaluate your own proposed setup, and then make an informed decision based on that evaluation.

The five different ways of cabling discussed here are grouped into two categories: *traffic links* and *HA links*. For the traffic links, the three main choices are to connect the firewalls directly to the routers, connect the firewalls to the routers via switches, or connect the firewalls in a full mesh. The HA links can either be directly connected between the NetScreens or connected via switches (see Figure 12.7).

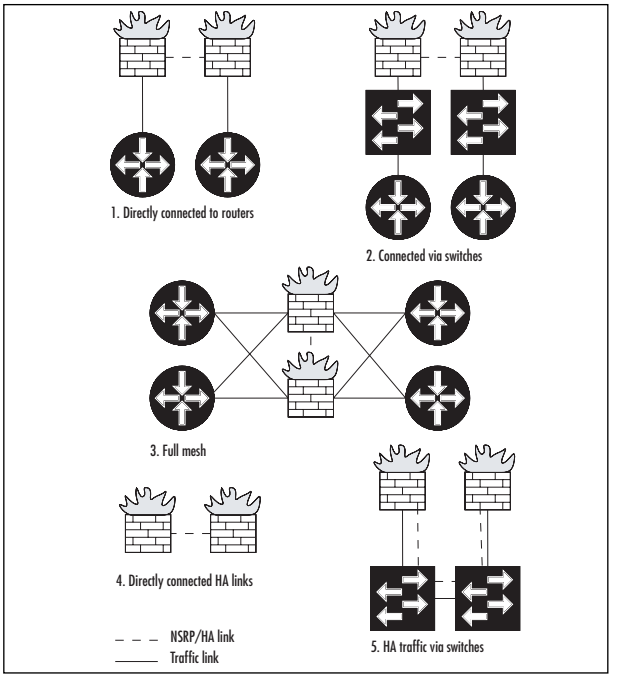
Connecting the Firewalls Directly to the Routers

This option reviews the advantages and disadvantages of cabling by connecting the NetScreens directly to the next hop routers.

Advantages

- Interface failure on the router is immediately detected, resulting in faster failover.
- There is less risk of failure (one less point of failure) without a switch in between.

Figure 12.7 Different Approaches to Cabling NetScreen Clusters



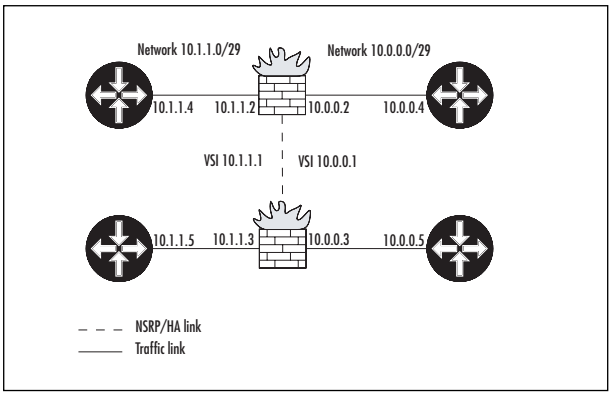
Disadvantages

- It is not possible to have a secondary HA path.
- Tracking a VRRP primary IP address is slightly more complicated.

Most of these advantages are self-explanatory, but the disadvantages require some elaboration. A secondary HA path can be configured to use in case the HA links fail. It is useful in certain scenarios, and a full explanation of its value is included in the "Avoiding the Split-Brain Problem" section later in this chapter. The issue of tracking VRRP IP addresses is a minor point. To negate this disadvantage, you only need to remember to use the Address Resolution Protocol (ARP) tracking method instead of the default method.

The typical use for this way of cabling is when firewalls act as transit nodes—that is, when there is no LAN directly behind the firewalls and you are using very small subnets for all interfaces (such as /30 or /29 subnets). (See Figure 12.8.) Note that you cannot use IP tracking if you use a /30 subnet for a VSI, because it does not leave room for a managed IP, which is necessary in order for IP tracking to work.

Figure 12.8 Directly Connected Routers



Connecting the Firewalls to Routers via Switches

This option covers connecting the NetScreens via Layer 2 switches, which in turn connect to the routers. The main pros and cons of using this approach include the following:

Advantages

- It is possible to use a secondary HA path.
- It is possible to avoid the need for another router if the firewalls are directly protecting a LAN, because the firewall can be the default gateway for the LAN hosts.

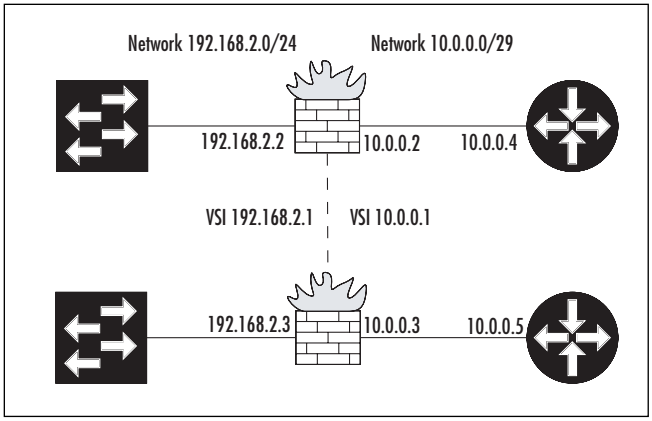
Disadvantages

- The interface failure on the router is not immediately detected—IP tracking must be used.
- Switches are additional points of failure that must be factored into availability calculations.

As must be expected, the advantages and disadvantages for this approach are almost the opposite of when you connect the firewalls directly to the routers.

This way of cabling is very common, simply because NetScreens are more commonly used for protecting LANs than for protecting transit-type traffic. A highly common scenario is where the NetScreens are cabled to switches on the inside, and directly to routers on the outside, typically Internet border routers (see Figure 12.9).

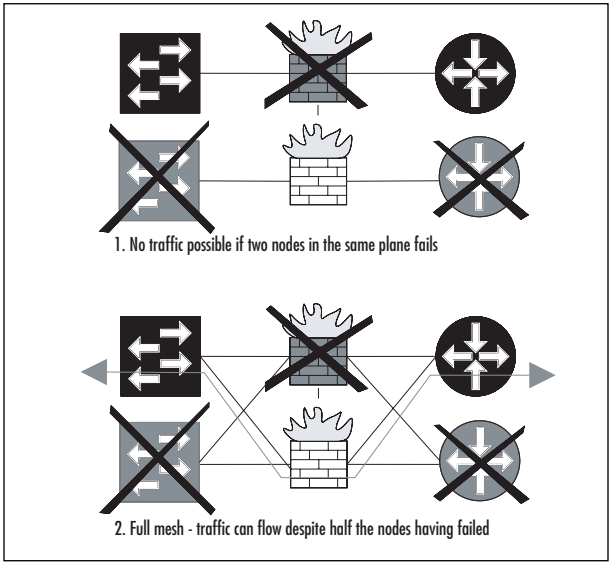
Figure 12.9 Firewalls Connected to Both Routers and Switches



Cabling for a Full-Mesh Configuration

Full-mesh cabling is for those who want very, very high levels of availability. A full mesh implies that each NetScreen has dual links to each of the neighboring nodes, be it routers or switches. This makes the network highly resilient not only to one device failure, but to two or even three simultaneous failures (see Figure 12.10).

Figure 12.10 Comparison between a Non-Full-Mesh and a Full-Mesh Network



The main pros and cons of a full-mesh setup are as follows:

Advantages

- Can survive multiple points of failure (as long as it is not both of the firewalls).
- It is possible to use a secondary HA path.
- "Full-mesh" is a good buzzword to use for making management happy (and convince them of approving the expenditure involved).

Disadvantages

- Requires substantially more resources—twice as many interfaces are used on all nodes.
- Configuring all of the nodes correctly is difficult, because there are many interactions to take into account.
- Testing all of the possible traffic paths is time-consuming.

In many cases, a full mesh is unwarranted and simply overkill. Before deciding to implement a full mesh, do the math for the availability required, and compare it to the expenses involved. Do not forget the hidden items such as the time needed for proper design, the configuration, and the verification testing. Remember, if you have not tested and verified that your HA setup is working as intended, you do not have a highly available network—it is as simple as that.

Using Directly Connected HA Links

When deciding how to cable the HA links, there are two choices: either you connect them directly using crossover cables, or you connect them via switches. The advantages and disadvantages of connecting them directly are summarized next.

Advantages

- There is a minimum risk of link failure.
- Link failure is detected immediately on both firewalls.
- NSRP data is not open for interception.
- Uses less resources (no ports needed to be allocated on switches).

Disadvantages

- None

Connecting HA Links via Switches

I do not recommend connecting HA links via switches since there are significant drawbacks to doing it this way.

Advantages

- Can be done, if necessary.

Disadvantages

- Must configure separate virtual local area networks (VLANs) on the switches to prevent NSRP traffic from escaping into the LAN.
- Must use port-based VLANs on the switches, because forwarded data packets may contain VLAN tags that could otherwise conflict with the switch configuration.
- Authentication and encryption of NSRP packets should be enabled.
- Must use HA link probes to detect link failures consistently.
- Depending on cabling, a single switch failure could disrupt both HA links.

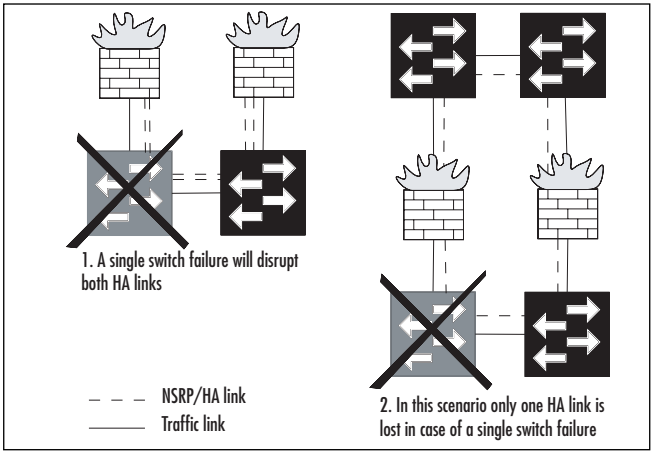
If there is more than one NetScreen cluster on the same network, the NSRP traffic could end up colliding if the same cluster ID was used for both clusters. NSRP packets are broadcast packets, and therefore easily use up significant bandwidth if not confined to their own VLAN. Depending on your setup, the NetScreens may need to forward data packets between each other, which is done over the HA data link. The packets are forwarded as is, and therefore can contain VLAN tags. Hence, the switches must be configured to use a port-based VLAN for the HA data link, and to accept and forward both tagged and untagged frames. Depending on your brand of switches, this may or may not be possible.

In addition, when sending NSRP traffic across switches it is good practice to both authenticate and encrypt the traffic, even if it is separated onto its own VLAN. Since a lot of sensitive information is contained in the NSRP packets, you wouldn't want someone to eavesdrop on them.

Finally, when cabling in this manner, the NetScreens are not able to detect a HA link failure directly; they have to rely on HA link probes to determine whether the link is still up or not. You have to explicitly enable these probes, and if you forget, you may find yourself in some "interesting" situations. (See "Avoiding the Split-Brain Problem" later in this chapter.)

Figure 12.11 shows two ways of cabling the HA links. In the first scenario, a single switch failure would disrupt both HA links, whereas in the second scenario, only one HA link would be affected.

Figure 12.11 Two Ways of Cabling HA Links through Switches



Adding a NetScreen to an NSRP Cluster

To add a NetScreen to an NSRP cluster, additional configuration is needed. The good news is that this is easy to do for a simple NSRP cluster. Once you have made it part of the cluster, you will probably want to add a few more configuration settings to make it fail over when appropriate. For now, let's focus on how to turn a stand-alone NetScreen into an NSRP cluster member.

The theory for this is simple. A NetScreen that has an NSRP cluster ID *greater* than zero is considered part of a cluster; valid cluster IDs range from 1 to 7. The following example shows you how to set the cluster ID.

Example: Setting the Cluster ID

In this example, we make the NetScreen part of NSRP cluster 1. By following these instructions for both firewalls when they are cabled for HA, you will see that they detect each other by looking at the output from the *get nsrp* command.

From the CLI:

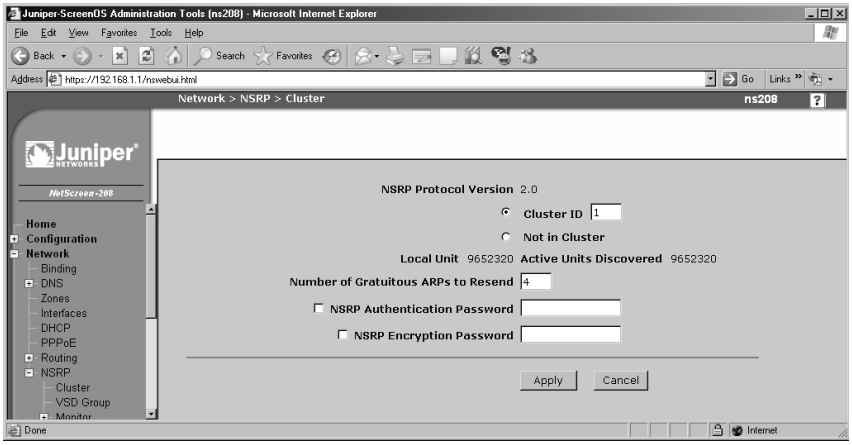
```
set nsrp cluster id 1
```

From the Web Interface:

1. Go to **Network** | **NSRP** | **Cluster** (see Figure 12.12).

2. Enter **1** as the Cluster ID.
3. Press **OK**.

Figure 12.12 Setting the NSRP Cluster ID



Example: Setting Both Cluster ID and Cluster Name

In addition to a cluster ID, each cluster can also be assigned a name, which is needed if you are using certificates and/or Simple Network Management Protocol (SNMP). The hostname used for those items should be the cluster name instead of the individual hostnames (which differ between the firewalls). If you do not do this, the certificates will only be valid on one of the firewalls, and the SNMP management system will most likely also refuse to acknowledge the second firewall. In this example, we set the cluster name to **beowulf** and the cluster ID to **7**.

From the CLI:

```
set nsrp cluster id 7
set nsrp cluster name beowulf
```

From the Web Interface:

It is not possible to set the cluster name from the Web interface.

NOTE

You can only set the cluster name through CLI.

Synchronizing the Configuration

Once you have cabled the NSRP cluster the way you want it, you must address the configuration side of things. Cluster members must have *near-identical* configurations in order to operate properly. The reason for near-identical and not *identical* is that some aspects must be unique to each NetScreen, including things such as the hostname and the management IP addresses.

While it is possible to copy the cluster configuration from another cluster member by using the *exec nsrp sync global-config* command, personal experience leads me to recommend a different approach to synchronizing the configurations initially. Unless you are highly familiar with the precise effects of this command, it will not do what you want or expect it to do. Take my advice on this one. If you want to experiment and find out for yourself how it works, by all means, go right ahead. If on the other hand you want to get your cluster set up as quickly and smoothly as possible, perform the following sequence of steps.

Initial Synchronization Procedure #1

1. Configure the first firewall fully, including the cluster configuration.
2. Back up the configuration to a Trivial File Transfer Protocol (TFTP) server using the *save config to tftp x.x.x.x netscreen1.cnf* command.
3. Open up the resulting file (*netscreen1.cnf*) in your favorite text editor.
4. Change the few things that should be different between the two firewalls. This typically includes hostname, management IP addresses (*mgt* interface and/or *manage-ip*), VSD group priority and preemption, and physical interface settings.
5. Save the changed file under a new filename (for example, *netscreen2.cnf*).
6. Download the new configuration file to the second NetScreen using the *save config from tftp x.x.x.x netscreen2.cnf to flash* command.
7. Once the configuration has been saved, reset the second firewall (remember to answer **n** when asked if you want to save the configuration—if you save it at this point, you will overwrite the recently downloaded configuration).
8. When the firewall has rebooted, log in and issue the *exec nsrp sync file* command, followed by *save all*. This will copy the various files (such as Public Key Infrastructure [PKI] information and Secure Shell [SSH] keys) from the first NetScreen and store it on the second NetScreen.
9. Reboot the second NetScreen to ensure it is using the new information that was just synchronized.
10. You should now have a fully working NSRP cluster. You can make further configuration changes at this point, which will automatically be propagated to all of the

members of the clusters. See Table 12.4 for a list of commands that do not propagate within the cluster.

Table 12.4 Commands that Do Not Propagate to Other Cluster Members

Interface Commands	NSRP Commands
(un)set interface <int> manage-ip <ip>	(un)set nsrp cluster ...
(un)set interface <int> phy ...	(un)set nsrp auth password <password>
(un)set interface <int> bandwidth ...	(un)set nsrp encrypt password <password>
(un)set interface redundant<X> phy primary <int>	(un)set nsrp monitor interface <int> (don't use this generally)
Console Commands	(un)set nsrp vsd-group id <X> preempt
(un)set console ...	(un)set nsrp vsd-group id <X> priority <prio>
Hostname	(un)set nsrp vsd-group id <X> monitor track-ip ...
(un)set hostname <hostname>	(un)set nsrp monitor ... (another one that's not generally used.)
SNMP	**Virtual Router Commands**
(un)set snmp name <sysname>	(un)set vrouter <name> router-id (this isn't right – they all sync unless you've issued *unset vrouter foo nsrp-config-sync*)

NOTE

Both the *clear...* and *debug...* commands do not, by default, propagate to other cluster members. To execute one of these categories of commands on all cluster members, use the form *clear cluster...* and *debug cluster...* commands.

The advantages of synchronizing the configuration using the preceding method are that you know precisely what is going on at all times, and there is no real risk of duplicate IP addresses conflicting with each other between the firewalls. This makes it possible to use this procedure safely when logged in remotely. Also, having the configuration files stored side-by-side makes it easy to compare them and see the differences (using the *diff* command for Unix and the *WinDiff* command for Windows).

If you do not have a TFTP server available, or for some other reason do not want to use the procedure previously outlined, the following is the more official "cold start" approach.

Initial Synchronization Procedure #2

1. Configure the first firewall fully, including the cluster configuration.
2. Run the *unset all* command on the second firewall to clear any existing configuration, and to confirm the action when prompted to do so.
3. Reset the second firewall; do not save the configuration when prompted.
4. When it has rebooted, issue the *set nsrp cluster id X* command to make it part of the same cluster as the first firewall (*X* being the same cluster ID used on the first firewall). Alternatively, do this via the Web interface.
5. Synchronize the files between the firewalls with the *exec nsrp sync file* command, and run on the second firewalls.
6. Synchronize the cluster configuration using the *exec nsrp sycn global-config run* command, and run on the second firewall.
7. Configure all of the settings that were not synchronized automatically. Refer to Table 12.4 for a list of commands you will have to enter manually. Also enter them on the second firewall.
8. Save the newly built configuration with the *save all* command.
9. Reset the second firewall.
10. After it has rebooted, run the *exec nsrp sync global-config checksum* command and verify that the configurations are in sync. If they are not, manually inspect the two configurations and discern any differences that must be corrected.

As you can see, this procedure leaves more room for error. However, this does not necessarily mean that errors *will* occur. If you know the setup and configuration details intimately, you will be able to use this procedure and have it run smoothly, or at least be able to correct any problems quickly. Personally, I prefer to take the safer route and use procedure #1. The choice, however, is yours.

NOTE

If you are using the Network Time Protocol (NTP) for keeping time on the firewalls in the cluster, you should add the *set ntp no-ha-sync* command, which will enable both firewalls to clock off of the NTP server independently. If you do not use this command, the clocks will be synchronized using the built-in NSRP time synchronization feature, which has much lower accuracy than NTP.

Determining When to Fail Over: The NSRP Ways

Similar to the options provided on the low-end range of NetScreen firewalls, NSRP provides a number of different methods that can be used to determine when a failover should be initiated. While the options in some cases may seem identical to their low-end cousins, do not confuse them—they are distinctly different, albeit subtly so. Also, if you recall from the earlier discussion, the low-end range of NetScreen firewalls provided VPN monitoring as one of the many ways to determine the failover point. This particular feature is not present when using NSRP because it is not considered necessary or appropriate at this level; it is only really useful on the small firewalls.

If you really like that feature, you can achieve almost the same thing using IP tracking towards one or many hosts that are reachable only through the VPN. For cases where there are no known hosts behind the VPN, simply tracking the VPN gateway may be sufficient.

Before going into detail about how to detect the need for a failover, let's look at a list of things that are already reason enough to fail over:

- Software crashes
- Hardware or power failure
- Link failure on monitored interfaces or zones
- Unavailability of one or more tracked IP addresses

The first two items—software and hardware failure—are detected automatically without any need for explicit configuration. The latter two items are available to provide flexibility in determining whether to fail over or not, and must be explicitly enabled to be in effect.

Using NSRP Heartbeats

Heartbeat monitoring is an integral part of NSRP, and is always active. It provides the mechanism to detect when a firewall becomes unresponsive for any reason, such as a software or

hardware failure or cable faults. There are only two user-configurable settings for this feature: the frequency of the heartbeats and the number of missed heartbeats allowed before a failover is triggered.

By default, heartbeats are sent once every second, and the threshold for lost heartbeats is set to three, implying that a failover would happen in less than four seconds in case the active firewall stops responding. That's quite impressive by most standards. It is however not the most aggressive setting—it is possible to lower the heartbeat frequency to a mere 200 milliseconds, giving us a sub-second failover time, in exchange for some added processing overhead due to the rapid heartbeats. In demanding environments it can be well worth the trade-off though. However, you will need to be very careful in selecting the equipment that the firewalls connect to—many, or even most, routers and switches are not capable of keeping up with such a quick failover, and may introduce additional delays before the network has stabilized.

If you do lower the heartbeat interval, it is also recommended that you increase the initial hold-down value. The reason for this is that the total time spent in the *initial* NSRP state should be of a sufficient period to allow synchronization of the state information before the unit makes itself available in the cluster. The actual hold-down time is calculated as *init-hold* x *hb-interval* milliseconds, which by default resolves as 5 ? 1000 = 5000 milliseconds, or five seconds. To compensate for lowering the heartbeat interval from 1000 to 200 milliseconds, it would be appropriate to set the *init-hold* value to 25, which would keep the actual time spent in the *initial* state at five seconds. You may need to adjust this to your particular setup, but this is a good starting point. Generally, it doesn't hurt to have a high initial hold-down time, since the only time this will have an impact on the network is if both firewalls power on at the same time, in which case the initial hold-down time to some extent determines how long it takes before the firewalls can start passing traffic.

Example: Configuring More Aggressive Heartbeats

To show how the heartbeat settings are configured, we will use the preceding example with the exception of allowing for four lost heartbeats before triggering the failover. This still provides sub-second failover, and also shows how to adjust the threshold value (since the lower value, three, is the default value).

Using the CLI:

```
set nsrp vsd-group hb-interval 200
set nsrp vsd-group hb-threshold 4
set nsrp vsd-group init-hold 25
```

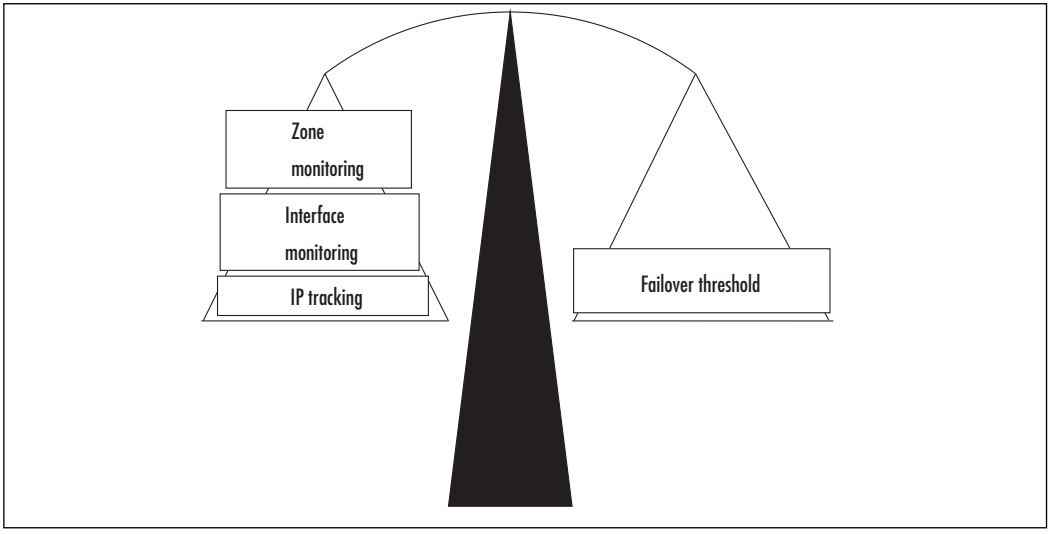
Using the Web Interface:

It is not possible to adjust the heartbeat settings from the Web interface.

Using Optional NSRP Monitoring

NSRP provides options for monitoring other things besides NSRP heartbeats—specifically, interface link status (interface monitoring), zone availability (zone monitoring), and IP address reachability (IP tracking). The IP tracking provided by NSRP is basically a superset of the IP tracking feature provided for low-end NetScreen firewalls. It also ties in with the other monitoring methods in that a common threshold value is used. This means that the total weight of all failed objects (interfaces, zones, and IP addresses) is compared against a single value, and the outcome of that comparison determines whether a failover is triggered (see Figure 12.13). If you are using only a single monitoring method, this is of no significance, but once you begin mixing different methods, it can become tricky to keep track of all of the interactions. Proper planning is the key to being successful in those circumstances. Difficult as it may be to get it right, there is no doubt that it provides a lot of flexibility.

Figure 12.13 The Relationship between the Monitoring Objects and the Failover Threshold



Two levels of monitoring are offered: *device level* and *VSD level*. Objects monitored on a device level affect all of the VSDs on that firewall, whereas objects monitored on a VSD level only affect that particular VSD. If the device level monitoring reaches the failover threshold, all VSDs on that device become inoperable. On the other hand, if a VSD failover threshold is reached, only that VSD becomes *inoperable*—other VSDs on that firewall may still be operable.

Example: Lowering the Failover Threshold

For this example, imagine that we want to work in percentages instead of the default failover threshold of 255. Setting the failover threshold to 100 makes it easier to think conceptually and thus makes it easier to explain to others, as in the following: "The loss of interface *X* counts as a 50 percent loss," rather than "Losing the link on interface *X* adds 127 to the weighted sum, and the failover threshold is at 255."

Using the CLI:

```
set nsrp monitor threshold 100
```

Using the Web Interface:

It is not possible to adjust this setting from the Web interface.

Using NSRP Interface Monitoring

The easiest optional monitoring method is interface monitoring. By marking an interface as monitored, its link state becomes a deciding factor for triggering failovers. By default, any single monitored interface can cause a failover. This can be changed by adjusting the weight attached to the interface. Each failed interface's weight is added to the failover threshold (that is, if there are two monitored interfaces, both with a weight of 50, and both fail, it will be counted as 100 towards the failover threshold).

Example: A Simple Interface Monitoring Setup

Consider a simple setup where interfaces ethernet1 and ethernet3 are used. If either of them fail, we want them to fail over to the backup firewall. Device level monitoring is used here.

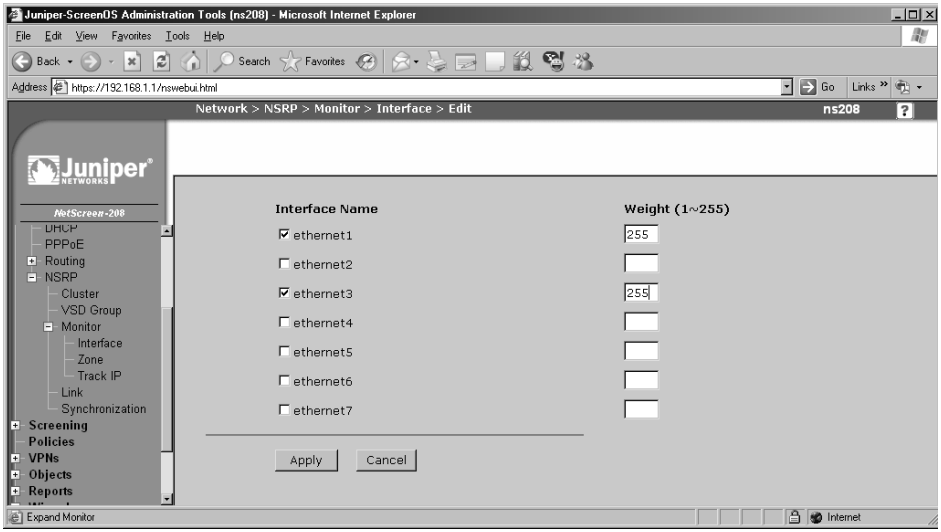
From the CLI:

```
set nsrp monitor interface ethernet1
set nsrp monitor interface ethernet3
```

From the Web Interface:

1. Go to **Network | NSRP | Monitor | Interface | VSD ID: Device** | **Edit interface** (see Figure 12.14).
2. Select **ethernet1**.
3. Set the Weight to **255**.
4. Select **ethernet3**.
5. Set the Weight to **255**.
6. Click **Apply**.

Figure 12.14 Configuring Interface Monitoring



Example: A More Complex Interface Monitoring Setup

In this example, we configure the monitoring of three interfaces: ethernet1, ethernet2, and ethernet3. Due to the setup used, it is acceptable to lose either ethernet2 or ethernet3, but not both. If the ethernet1 link goes down at any time, a failover should be triggered. Since the monitor threshold cannot be changed via the Web interface, we use the default threshold value of 255 for this example.

From the CLI:

```
set nsrp monitor interface ethernet1 weight 255
set nsrp monitor interface ethernet2 weight 128
set nsrp monitor interface ethernet3 weight 128
```

From the Web Interface:

1. Go to **Network** | **NSRP** | **Monitor** | **Interface** | **VSD ID: Device** | **Edit interface**.
2. Select **ethernet1**.
3. Set the Weight to **255**.
4. Select **ethernet2**.
5. Set the Weight to **128**.
6. Select **ethernet3**.
7. Set the Weight to **128**.
8. Click **Apply**.

Using NSRP Zone Monitoring

Instead of monitoring individual links and juggling the weight of the interfaces, it is sometimes useful to monitor an entire *zone*. When monitoring a zone, the zone is not considered to have failed until there are no working interfaces left in that zone. The definition of "working" in this case is that its link state is "up." As such, zone monitoring provides a slightly more high-level approach to monitoring, compared to interface monitoring. Its main advantage shines through when you are working on firewalls where you have lots of interfaces (say 10+) bound to the same zone. It saves a lot of typing and clicking when you can ask the firewall to monitor the zone instead of every interface. Alternatively, if a monitored zone contains no interfaces at all, that zone will never fail.

Just as with interface monitoring, each zone has an assigned weight. If the zone fails, its weight is counted towards the failover threshold; once the failover threshold is reached, a failover is triggered. Keep in mind that the failover threshold is shared between all of the monitoring methods, and that it is possible to have a failover triggered as a result of a combination of failed interfaces and failed zones. Care must be taken to ensure that the interactions are what you want them to be.

Example: Monitoring the Untrust Zone

For this example, only a single zone is monitored and the default weight is used. This scenario means that the failure of the zone will cause a failover.

From the CLI:

```
set nsrp monitor zone untrust
```

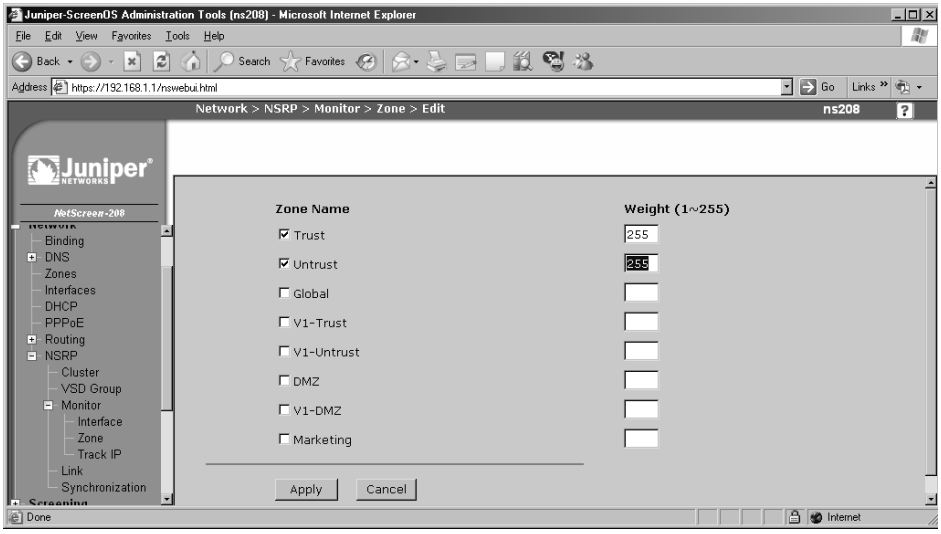
From the Web Interface:

1. Go to **Network** | **NSRP** | **Monitor** | **Zone** | **VSD ID: Device** | **Edit Zone** (see Figure 12.15).
2. Select the **Untrust** zone.
3. Set the Weight to **255**.
4. Click **Apply**.

Example: Using Combined Interface and Zone Monitoring

For this example, consider a fictional scenario where we do not want to cause a failover unless all of the interfaces in custom zone **Internet** as well as interface **ethernet6** have failed. To achieve this, we divide the total weight needed to trigger a failover (255) between the two objects, giving us a weight of 128 for the Internet zone, and the same for the ethernet6 interface. It would also have worked equally well with the weights set to 128 + 127 or 1 + 254 for the zone and interface, respectively, and variations thereof.

Figure 12.15 Configuring Zone Monitoring



From the CLI:

```
set nsrp monitor zone Internet weight 128
set nsrp monitor interface ethernet6 weight 128
```

From the Web Interface:

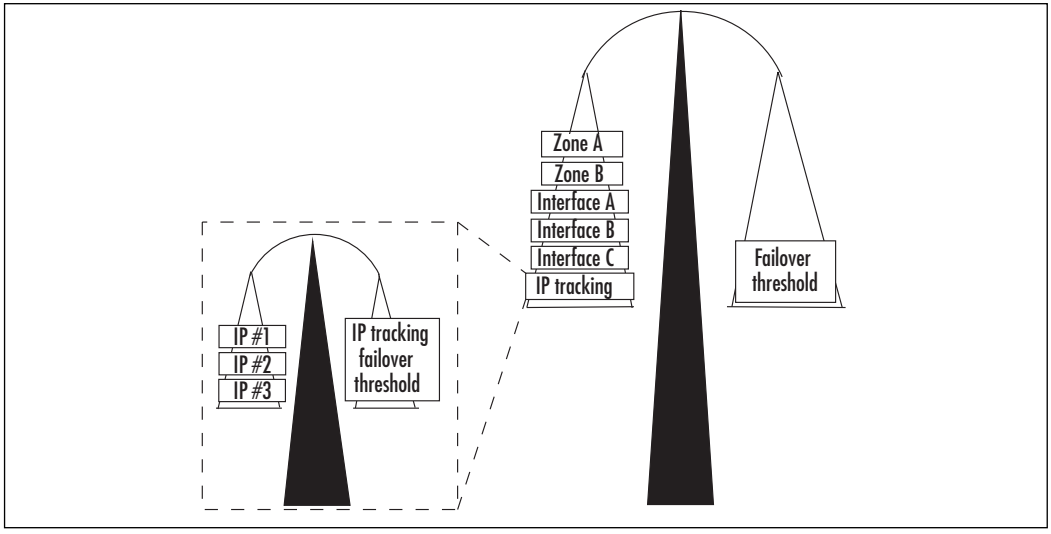
1. Go to **Network | NSRP | Monitor | Zone | VSD ID: Device | Edit Zone**.
2. Select the **Internet** zone.
3. Set the Weight to **128**.
4. Click **Apply**.
5. Go to **Network | NSRP | Monitor | Interface | VSD ID: Device | Edit Interface**.
6. Select **ethernet6**.
7. Set the Weight to **128**.
8. Click **Apply**.

Using NSRP IP Tracking

The third monitoring option is tracking IP address reachability. By regularly contacting the tracked IP address, its availability is determined. This is similar to the functionality provided by the low-end NetScreen firewalls, but with a few pointed differences such as the way that the weight associated with the IP address is used, and the method used to contact the IP address.

If you recall, all three optional NSRP monitoring features (interface monitoring, zone monitoring, and IP tracking) share the same failure threshold and counter. With IP tracking, another layer is introduced. While each monitored interface and zone counts as its own separate object with associated weight, IP tracking is only ever a single monitored object with one weight attached. The entire IP tracking object can either be failed or not. There is no in-between value indicating that some IP addresses have failed while others are still responding. To determine whether the IP tracking object is flagged as failed or not, another failure threshold is used internally. Each monitored IP address has an associated internal weight, which is added towards the internal IP tracking failover threshold. When enough IP addresses have failed, the internal threshold value is reached or exceeded and the IP tracking object will count as failed. With that, the weight associated to the IP tracking object will be counted towards the device failover threshold (see Figure 12.16).

Figure 12.16 The Relationship between All Monitored Objects



In addition to carrying its own internal failover threshold, IP tracking provides two different methods of contacting the tracked IP address: the traditional Internet Control Message Protocol (ICMP) ping approach and ARP requests. The latter are intended to be used mainly in instances where the tracked IP address is the virtual IP of a directly connected cluster of routers using the VRRP. Due to the workings of VRRP, it is not generally possible to use pings to determine whether it is or is not reachable. ARP request packets, however, always solicit a response from the router. The restriction when using ARP packets is that the tracked IP address must be in the same physical subnet as the NetScreen since no IP routing is done for ARP packets.

It is possible to specify the interval for the probe packets as well as the failure threshold for each tracked IP address. The failure threshold determines how many responses can be lost before that IP address is considered down.

When using IP tracking, there are restrictions on which interface can be used as the source for the tracking packets. Of particular note is that a plain VSI cannot be used since that IP address might move between different hardware units. Think about it. To be able to track an IP address, you must send the requests from an IP address that is available on the firewall regardless of whether it is the master of the cluster or a backup. Since the IP addresses assigned to VSIs follow the master unit, they are unsuitable for IP tracking purposes. In order to work around this problem, any interface that is the source of IP tracking packets must have a managed IP address specified. Since the managed IP does not move between units, it can be used.

Example: Using IP Tracking to Determine VPN Availability

As mentioned earlier, NSRP does not provide any explicit methods for tracking VPN states and triggering failovers based on those states. We can, however, simulate it by tracking IP addresses that are only available as long as the VPN tunnel is up (in other words, IP addresses that can only be reached *through* the tunnel). In this example, we configure things to do precisely that. We work on the assumption that IP addresses 172.16.5.1 and 172.16.5.2 are only reachable through the tunnel, and that they are both likely to always be available (that is, they belong to routers, compared to a PC workstation which may or may not be powered on). Furthermore, we assume that the VPN tunnel is already configured and bound to the ethernet3 interface, which has an IP address of 1.1.1.1.

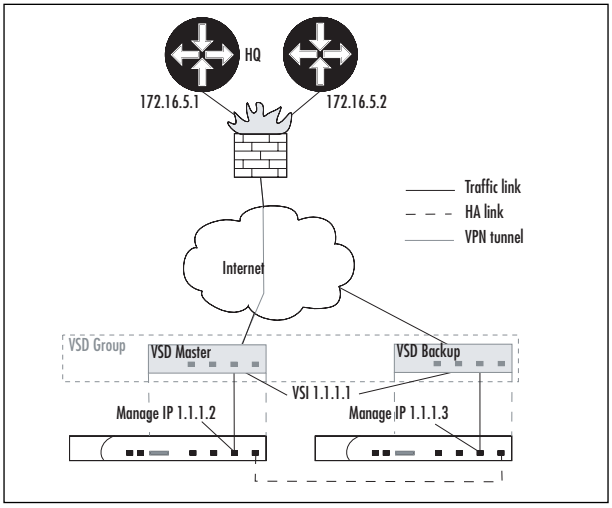
To enable IP tracking packets to be sent from the ethernet3 interface, a managed IP address is added onto each firewall (they must be different from each other and from the VSI IP address). We have decided that pinging each IP address every five seconds is sufficient, and we allow for four lost responses before flagging the IP address as down. Each IP address is assigned a weight that is half of the IP tracking failover threshold, meaning that both IP addresses must be unreachable before the IP tracking object itself is flagged as down. The default weight of 255 is used on the IP tracking object, which means that if it is marked as failed, a failover will be triggered. Figure 12.17 shows this network layout.

From the CLI:

```
# on firewall A, use this command
set interface ethernet3 manage-ip 1.1.1.2
# on firewall B, use this instead
set interface ethernet3 manage-ip 1.1.1.3
#
set nsrp monitor track-ip ip 172.16.5.1
set nsrp monitor track-ip ip 172.16.5.1 interval 5
set nsrp monitor track-ip ip 172.16.5.1 threshold 4
```

```
set nsrp monitor track-ip ip 172.16.5.1 weight 50
set nsrp monitor track-ip ip 172.16.5.2
set nsrp monitor track-ip ip 172.16.5.2 interval 5
set nsrp monitor track-ip ip 172.16.5.2 threshold 4
set nsrp monitor track-ip ip 172.16.5.2 weight 50
set nsrp monitor track-ip threshold 100
set nsrp monitor track-ip
# this command uses the default value, and is only shown for clarity's sake
set nsrp monitor track-ip weight 255
```

Figure 12.17 A VPN to Corporate HQ



From the Web Interface:

1. Go to **Network** | **Interfaces** | **ethernet3** | **Edit**.
2. Enter **1.1.1.2** or **1.1.1.3** as the Managed IP on Firewalls A and B, respectively.
3. Click **OK**.
4. Go to **Network** | **NSRP** | **Monitor** | **Track IP**.
5. Click **New**.
6. Enter **172.16.5.1** as the Track IP.
7. Set the Weight to **50**.
8. Specify an Interval of **5**.
9. Specify the Threshold as **4**.
10. Click **OK** to add this IP address to the list of tracked addresses.

11. Click **New** to add a second IP address.
12. Enter **172.16.5.2** as the Track IP.
13. Set the Weight to **50**.
14. Specify an Interval of **5**.
15. Specify the Threshold as **4**.
16. Click **OK**.
17. Go to **Network** | **NSRP** | **Monitor** | **Track IP** | **VSD: Device** | **Edit**.
18. Select **Enable Track IP**.
19. Designate the Failover Threshold as **100**.
20. Click **Apply**.

NOTE

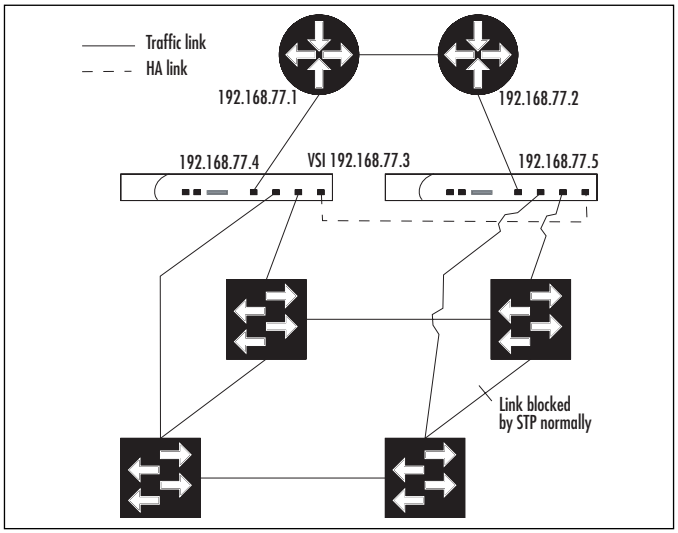
If you use the *set nsrp monitor track-ip...* commands, the IP tracking applies to all VSDs on this firewall. If you want per-VSD IP tracking, use the *set nsrp vsd-group id X monitor track-ip...* commands.

Example: Combining Interface, Zone, and IP Tracking Monitoring

If you are working in high-end environments, you will probably wind up using multiple tracking methods within the same NSRP cluster. To give you a head start, let's review such an example.

In this scenario, we have two NetScreens directly connected to two VRRP routers on one side and four switches on the other. We use interface monitoring towards the routers, because that is the quickest way to determine a severe failure of a router, and also IP tracking using ARP. Either of these options should trigger a failover to minimize downtime. We use zone monitoring towards the switches because we have dual interfaces in the *inside* zone. Spanning Tree Protocol (STP) is used within the inside network to prevent network loops. It's normally partitioned in two halves, each with a direct connection to the firewalls. This achieves the best use of bandwidth, but in the case of a switch failure, all inside traffic can be sent via one of the links to the firewall. While we could achieve the same monitoring effect by monitoring each of the links to the switch and giving that link a weight of 50 percent of the failover threshold, zone monitoring is the more elegant way to implement this (see Figure 12.18).

Figure 12.18 Network Using STP



Interface ethernet1 is connected to the routers and bound to the *outside* zone. Interfaces ethernet2 and ethernet3 are connected to the two sets of switches and are bound to the *inside* zone. The routers' primary address is 192.168.77.1, and the backup IP address is 192.168.77.2. On the firewall side, 192.168.77.3 is the VSI address, and 192.168.77.4 and 192.168.77.5 are the managed IP addresses used for originating the IP tracking packets from. We use 100 as the failover threshold (for its value of enabling us to think in percentages), except for the internal IP tracking threshold where we use 101 instead (to make it clear which values are related).

Since the monitor threshold cannot be adjusted via the Web interface, weights of 255 are used throughout the configuration when using the Web interface.

From the CLI:

```
set zone name outside
set zone name inside
set interface ethernet1 zone outside
set interface ethernet1 ip 192.168.77.3/29
set interface ethernet1 route
# on firewall A, use this command
set interface ethernet1 manage-ip 192.168.77.4
# on firewall B, use this command instead
set interface ethernet1 manage-ip 192.168.77.5
#
set interface ethernet2 zone inside
set interface ethernet3 zone inside
# IP address configuration for ethernet2 & 3 omitted
```

```
set arp always-on-dest
set nsrp cluster id 1
set nsrp monitor threshold 100
set nsrp monitor interface ethernet1 weight 100
set nsrp monitor zone inside weight 100
set nsrp monitor track-ip ip 192.168.77.1
set nsrp monitor track-ip ip 192.168.77.1 weight 101
set nsrp monitor track-ip threshold 101
set nsrp monitor track-ip weight 100
set nsrp monitor track-ip
```

From the Web Interface:

1. Go to **Network | Zones**.
2. Click **New**.
3. Enter **outside** as the Zone Name.
4. Click **OK**.
5. Click **New**.
6. Enter **inside** as the Zone Name.
7. Click **OK**.
8. Go to **Network | Interfaces | ethernet1 | Edit**.
9. Enter **outside** as the Zone Name.
10. Enter **192.168.77.3** as the IP Address and **29** as the Netmask.
11. Click **Apply**.
12. Use **192.168.77.4** as the Managed IP on Firewall A, and **192.168.77.5** as the Managed IP on Firewall B.
13. Enter **Route** as the Interface Mode.
14. Click **OK**.
15. Go to **Network | Interfaces | ethernet2 | Edit**.
16. Enter **inside** as the Zone Name. (The IP address configuration for this interface is not shown in this example.)
17. Click **OK**.
18. Go to **Network | Interfaces | ethernet3 | Edit**.
19. Select **inside** as the Zone Name. (The IP address configuration for this interface is not shown in this example.)
20. Click **OK**.
21. Go to **Network | NSRP | Cluster**.

22. Select **Cluster ID** and enter an ID value of **1**.
23. Click **Apply**.
24. Go to **Network | NSRP | Monitor | Interface | VSD ID: Device | Edit Interface**.
25. Select **ethernet1**.
26. Set the Weight to **255**.
27. Click **Apply**.
28. Go to **Network | NSRP | Monitor | Zone | VSD ID: Device | Edit Zone**.
29. Enter **inside** as the Zone Name.
30. Set the Weight to **255**.
31. Click **Apply**.
32. Go to **Network | NSRP | Monitor | Track IP**.
33. Click **New**.
34. Enter **192.168.77.1** as the Track IP.
35. Set the Weight to **101**.
36. Click **OK**.
37. Go to **Network | NSRP | Monitor | Track IP | VSD: Device | Edit**.
38. Select **Enable Track IP**.
39. Designate the Failover Threshold as **101**.
40. Click **Apply**.

One command used in this example that we have not discussed is the *set arp always-on-dest* command. Whenever you connect your NetScreens to a load balancer or router running VRRP or similar protocol, it is wise to enable this option. What it does is force an ARP lookup for incoming sessions, instead of relying on the MAC address in the incoming frame. This can alleviate problems in cases where the load balancer or router sends packets from its virtual IP address with its physical MAC. Without this option enabled, you might end up with sessions "stuck to" one particular router, even after a failover has occurred. Be aware that this option can only be adjusted via the CLI, not via the Web interface. Note that enabling this option will not cause an ARP request for every single session. The ARP table is consulted first; only if an entry is not found there will it send an ARP query.

NOTE

The *set arp always-on-dest* option is available only from the CLI.

Reading the Output from *get nsrp*

Before we continue exploring the NSRP features, let's take a look at what sort of feedback to expect from the NetScreen in regards to the NSRP configuration. You will undoubtedly find yourself frequently using the *get nsrp* command whenever you are working with NSRP clusters. Therefore, being familiar with its output will make you more efficient at controlling your cluster.

In addition to the *get nsrp* command and all of its sub-commands, one other command that you should add to your arsenal is a simple *get config* filtered to only show NSRP settings (for instance, **get config | include nsrp** or, in its abbreviated form, **get conf | i nsrp**). This shows you all of the NSRP commands that have been entered into the configuration, and can often be easier to read than the *get nsrp* output. No state information is provided when you use this command, therefore it is not a replacement for *get nsrp*, but it is a very useful complement.

Looking into an NSRP Cluster

It would be impossible to provide listings of all of the possible outputs from *get nsrp*, and there is no attempt to do so here. Instead, we look at a single example that shows some of the NSRP features enabled, but not all. Also, there is a wealth of information available from the NSRP sub-command printouts; do a *get nsrp ?* to see the options.

Example

In this example, we examine a numbered output from one NS-500 firewall participating in an NSRP cluster.

From the CLI:

```
1. ns(B)-> get nsrp
2. nsrp version: 2.0

   cluster info:
3. cluster id: 2, no name
4. local unit id: 2071408
   active units discovered:
5. index: 0, unit id:   2071408, ctrl mac: 0010db1f9b75,
     data mac: 0010db1f9b76
6. index: 1, unit id:   2070976, ctrl mac: 0010db1f99c5,
     data mac: 0010db1f99c6
7. total number of units: 2

   VSD group info:
8. init hold time: 5
9. heartbeat lost threshold: 3
```

```
10. heartbeat interval: 200(ms)
    group priority preempt holddown inelig   master       PB other members
11.    0       50 no             3 no      2070976   myself
12. total number of vsd groups: 1
13. Total iteration=144,time=926517,max=16862,min=1355,average=6434

    RTO mirror info:
14. run time object sync:   enabled
15. ping session sync: enabled
16. nsrp data packet forwarding is enabled

    nsrp link info:
17. control   channel: ha1 (ifnum: 5)  mac: 0010db1f9b75 state: up
18. data      channel: ha2 (ifnum: 6)  mac: 0010db1f9b76 state: up
19. ha secondary path link not available

20. NSRP encryption: disabled
21. NSRP authentication: disabled
22. device based nsrp monitoring threshold: 255, weighted sum: 0, not failed
23. device based nsrp monitor interface: ethernet1/1(weight 255, UP)
      ethernet3/1(weight 255, UP)
24. device based nsrp monitor zone:
25. device based nsrp track ip: (weight: 255, disabled)
26. number of gratuitous arps: 4 (default)
27. config sync: enabled

28. track ip: disabled
```

The following is an explanation of each numbered line of code.

1. The command being issued. As can be inferred from the prompt, it is done from the backup firewall.
2. The version of NSRP.
3. This line tells us that this firewall is participating in NSRP cluster 2, and does not have a cluster name assigned.
4. Here we learn what the ID number of this firewall is. This ID refers to the hardware itself, not a VSD. It can be used for correlation on other cluster members (see numbered line 11 in the printout).
5. This line states that this firewall has been found in the NSRP cluster. The unit ID here matches the ID printed on line 4. You can also see the virtual MAC addresses used by this firewall for its NSRP messages and forwarded data. This is different

from the interface MACs, which you can obtain by running the *get interface* command.

6. On this line we learn that another firewall has been discovered in the cluster, this one with an ID of 2070976. If you were to log on to that firewall and run *get nsrp*, you would find that this is the same ID that is printed on line 4.
7. A count of the number of discovered cluster members. If this says 1, you know something is missing.
8. The *init-hold* value (discussed in the section "Determining When to Fail Over: The NSRP Ways" earlier in this chapter).
9. The number of lost heartbeats before a failover is initiated.
10. The heartbeat interval, which in this case has been lowered to its minimum value. Worth noting is that the *init-hold* value on line 8 has not been adjusted to compensate for this.
11. This line presents a summary of VSD group 0. You can see that this firewall has a priority of 50 for this VSD (100 is the default), preemption is not enabled, but if it were, a three-second hold-down would be in effect, preventing rapid failovers back and forth. You can also see that the VSD is not marked as ineligible, and that the firewall with ID 2070976 has the master VSD for this group, and that this firewall is the Primary Backup. If there are multiple VSD groups defined, you will get multiple lines printed, each with this summary.
12. This line shows the number of VSD groups defined on this firewall. Here it tells us that only one VSD group exists.
13. The output here is only intended for debugging purposes by Juniper/NetScreen, and can safely be ignored.
14. This line shows us that RTO mirroring is enabled (see the "Taking Advantage of the Full NSRP" section later in this chapter).
15. Here is more debug output. Again, you can simply ignore this line.
16. Here we learn that data forwarding across the HA data link is enabled.
17. This line shows us that the HA control link is up, and that interface ha1 is used for control messages.
18. Similar to the line before it, this shows that the HA data link is up and available, and that interface ha2 is the designated HA data link.
19. This shows us that no secondary NSRP path is configured on this firewall. Even if it were configured on the other firewall in the cluster, it would not help, because this firewall is not expecting NSRP messages on any interfaces other than ha1 and ha2.

20. Because NSRP encryption is not enabled, it likely means that the HA links are direct crossover cables between the two firewalls.
21. Just as with the encryption, authentication of NSRP messages is disabled.
22. This line is very useful; it not only tells us what the failover threshold is, but also shows us how far towards that threshold we have progressed. This can be valuable information when you are troubleshooting. If you do not like having to compare the weighted sum to the threshold, you can refer to the end of the line to see if enough things have failed to warrant a failover or not. Note that what is shown here is the *device* monitor settings. If you have set monitoring on VSD level, you need to use the *get nsrp vsd-group id X monitor* command to see the corresponding information for the VSD.
23. On this line, we find all of the device level interface monitoring that is configured, and the weight and state of those links. Again, this only shows interface monitoring configured on the device level, not on the VSD level.
24. Device level zone monitoring is shown on this line. In this case, no zones are monitored. To see VSD level zone monitoring, use the *get nsrp vsd-group id X monitor* command instead.
25. Similar to the preceding lines, this line shows whether IP tracking is enabled on device level, and the weight of that IP tracking. To get the details for the actual IP addresses tracked, you need to use the *get nsrp monitor track-ip* command. Similarly, if you have IP tracking on VSD level, use the *get nsrp vsd-group id X monitor track-ip* command.
26. This line shows the number of gratuitous ARPs that will be sent right after a failover.
27. Automatic configuration synchronization is enabled according to this line. This is almost always what you want. If you know that you do not want it, use the *unset nsrp config sync* command to change it.
28. This line is mainly a legacy leftover from older days. Line number 25 provides more information, so you can safely ignore this line, if you want.

Using NSRP-Lite on Midrange Appliances

With all that theory under our belt, it's time to see how we can implement a highly available network by using the NSRP-Lite feature available on some of the midrange NetScreen appliances. NSRP-Lite is a slimmed-down variant of NSRP that does not support the full feature-set of NSRP. All of the features discussed so far are available, however, which makes NSRP-Lite a very formidable feature in and of itself.

The two main things that NSRP-Lite cannot do are the Active/Active setup and synchronization of Run-Time Objects (RTOs). The lack of RTO synchronization means that in case of a failover, any existing sessions and VPNs will be lost and must be reestablished. If you are using VPNs with NSRP-Lite, remember to enable VPN monitoring with the *rekey* option to ensure that the VPNs are reestablished after a failover.

Since the midrange NetScreen appliances are targeted towards small and medium enterprises (SMEs), we go through example setups fitting for that category. We start off with a simple but still fully usable example, followed by a more advanced setup where we make good use of local interfaces to provide redundant outgoing paths.

NOTE

By default, the NetScreen firewalls do not inspect TCP packets to verify that they are part of an existing TCP session; only source and destination information is matched against the policies. This is very helpful if you have asymmetric routing or are using NSRP-Lite, since it allows sessions to survive asymmetric routing as well as failovers.

It does, however, mean that an attacker can introduce packets into your network easier. If you are concerned about that, you can force the NetScreen to verify each TCP session by using the *set flow tcp-syn-check* command. When TCP SYN-flag checking is active, only TCP packets with the SYN flag set can create session entries, and only TCP traffic that is part of that session will be allowed through.

Basic NSRP-Lite Usage

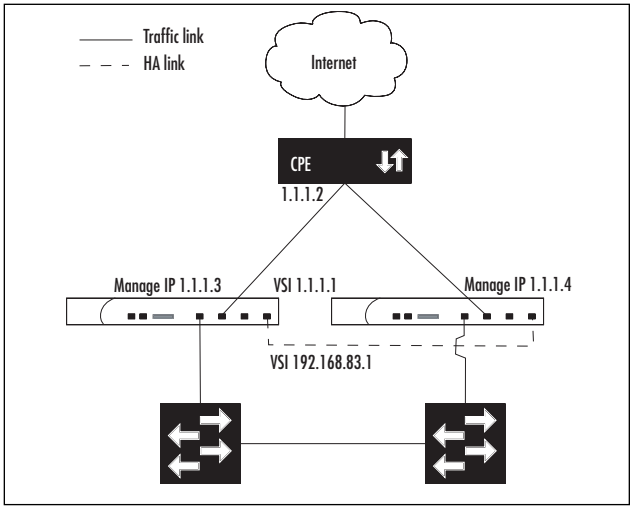
Using only the features covered up to this point, let's see how we can provide an office network with highly available Internet access.

Example: Providing HA Internet Access

For this example, imagine that we have an office network in need of Internet access. The Internet access is delivered to the premises by a third party, which provides the Customer Premises Equipment (CPE) and is also responsible for said CPE. This particular CPE contains a built-in Ethernet switch with a few ports (say four) for good measure. Upper management has decreed that Internet access is of utmost importance, and we should make it as highly available as we can, given our equipment. The equipment at our disposal includes a number of switches and two midrange NetScreen firewalls capable only of NSRP-Lite.

After working through the possible options, the network layout shown in Figure 12.19 is decided upon and needs to be implemented. Based on this design, we produce the following configuration for the NetScreens.

Figure 12.19 Office Network with HA Internet Access



From the CLI:

```
set zone name office
set zone name internet
set zone office vrouter trust-vr
set zone internet vrouter trust-vr
set interface ethernet1 zone office
set interface ethernet2 zone internet
set interface ethernet3 zone ha
set interface ethernet4 zone ha
set interface ethernet1 ip 192.168.83.1/24
set interface ethernet2 ip 1.1.1.1/29
# on firewall A, use these commands
set interface ethernet2 manage-ip 1.1.1.3
set hostname sme-fwA
# on firewall B, use these commands instead
set interface ethernet2 manage-ip 1.1.1.4
set hostname sme-fwB
#
set vrouter trust-vr route 0.0.0.0/0 interface ethernet2 gateway 1.1.1.2
set nsrp cluster id 1
set nsrp cluster name sme
```

```
set nsrp monitor interface ethernet1
set nsrp monitor interface ethernet2
# configure aggressive IP tracking
set nsrp monitor track-ip ip 1.1.1.2
set nsrp monitor track-ip ip 1.1.1.2 weight 255
set nsrp monitor track-ip ip 1.1.1.2 threshold 2
set nsrp monitor track-ip
# prefer to have firewall #1 as the master, but do not preempt
# on firewall A, use this command
set nsrp vsd-group id 0 priority 1
# on firewall B, use this instead
set nsrp vsd-group id 0 priority 2
#
# configure for sub-second failover on lost heart-beats
set nsrp vsd-group hb-interval 200
set nsrp vsd-group hb-threshold 4
set nsrp vsd-group init-hold 25
```

From the Web Interface:

1. Go to **Network | Zones | New** to create a new zone.
2. Enter **office** as the Zone Name.
3. Select **trust-vr** as the Virtual Router Name.
4. Click **OK**.
5. Click **New**.
6. Enter **internet** as the Zone Name.
7. Select **trust-vr** as the Virtual Router Name.
8. Click **OK**.
9. Go to **Network | Interfaces | ethernet1 | Edit**.
10. Enter **office** as the Zone Name.
11. Enter **192.168.83.1** as the IP Address and **24** as the Netmask.
12. Click **OK**.
13. Go to **Network | Interfaces | ethernet2 | Edit**.
14. Enter **internet** as the Zone Name.
15. Enter **1.1.1.1** as the IP Address and **29** as the Netmask.
16. Click **Apply**.
17. Use **1.1.1.3** as the Managed IP on Firewall A, and **1.1.1.4** on Firewall B.
18. Select **Route** as the Interface Mode.

19. Click **OK**.
20. Go to **Network | Interfaces | ethernet3 | Edit**.
21. Select **HA** as the Zone Name.
22. Click **OK**.
23. Go to **Network | Interfaces | ethernet4 | Edit**.
24. Select **HA** as the Zone Name.
25. Click **OK**.
26. Go to **Network | DNS**.
27. Enter **sme-fwA** as the hostname on Firewall A, and **sme-fwB** on Firewall B.
28. Click **Apply**.
29. Go to **Network | Routing | Routing Entries**.
30. Click **New** after selecting **trust-vr** as the virtual router to create the new routing entry in.
31. Enter **0.0.0.0** as the IP Address, and **0.0.0.0** as the Netmask.
32. Select **Gateway** instead of Next Hop Virtual Router Name.
33. Set the Interface to **ethernet2**.
34. Enter **1.1.1.2** as the Gateway IP Address.
35. Click **OK**.
36. Go to **Network | NSRP | Cluster**.
37. Select **Cluster ID** and enter **1** as the ID.
38. Click **Apply**.
39. Go to **Network | NSRP | Monitor | Interface | VSD ID: Device | Edit Interface**.
40. Select **ethernet1**.
41. Set the Weight to **255**.
42. Select **ethernet2**.
43. Set the Weight to **255**.
44. Click **Apply**.
45. Go to **Network | NSRP | Monitor | Track IP**.
46. Click **New**.
47. Enter **1.1.1.2** as the Track IP.
48. Set the Weight to **255**.

49. Set the Threshold to **2**.
50. Click **OK**.
51. Go to **Network | NSRP | Monitor | Track IP | VSD ID: Device | Edit**.
52. Select **Enable Track IP**.
53. Click **Apply**.
54. Go to **Network | NSRP | VSD Group | Group ID 0 | Edit**.
55. Set the Priority to **1** on Firewall A, and to **2** on Firewall B.
56. Click **OK**.

NOTE

The cluster name and heartbeat settings must be entered via the CLI even if you are using the Web interface because they cannot be adjusted via the Web interface.

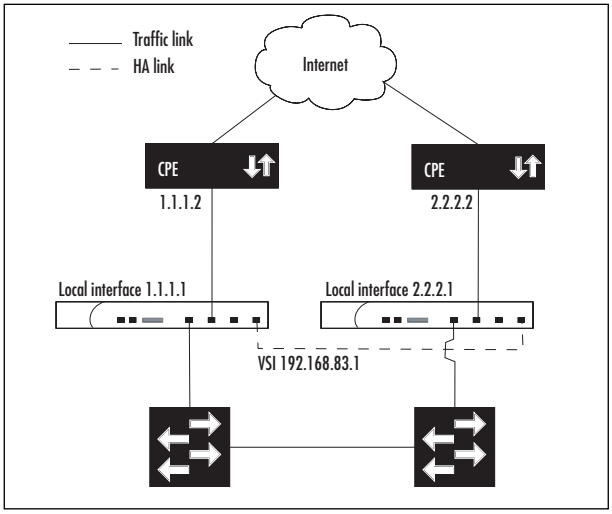
Working with Local Interfaces in an NSRP-Lite Setup

If we adjust one of the inputs to the scenario to only have a single Internet link, we end up with a more interesting example that is worth examining. In the case where we have two separate Internet links provided to the premises, each with its own CPE and only providing a single Ethernet port, our NetScreen setup becomes quite different. Let's take a closer look at how to deal with this new scenario.

Example: HA Internet via Dual Providers

While this example is similar to the previous one, the resulting configuration is quite different because in this case it is not possible to connect both of the NetScreens to each of the CPE routers (see Figure 12.20). Instead, the individual firewall must have a local configuration for talking to its connected CPE. To achieve this when running NSRP (or NSRP-Lite), you must use local interfaces so that the configuration of those interfaces does not move between the firewalls. You can create a local interface by removing the default VSD once NSRP is enabled, and then create a new VSD and only bind select interfaces to it. Any interfaces not bound to the new VSD remain as local interfaces.

Figure 12.20 Network Layout Using Local Interfaces



For clarity's sake, we omit the configuration for sub-second failover on lost heartbeats as well as the more aggressive settings for the IP tracking. Also note that because the interfaces towards the CPE are local interfaces, there is no need to set a Managed IP for the IP tracking.

From the CLI:

```
set zone name office
set zone name internet
set zone office vrouter trust-vr
set zone internet vrouter trust-vr
set interface ethernet1 zone office
set interface ethernet2 zone internet
set interface ethernet3 zone ha
set interface ethernet4 zone ha
set nsrp cluster id 1
set nsrp cluster name sme
set nsrp monitor interface ethernet1
set nsrp monitor interface ethernet2
unset nsrp vsd-group id 0
set nsrp vsd-group id 1
# only ethernet1:1 is a VSI now
set interface ethernet1:1 ip 192.168.83.1/24
# on firewall A, use these commands
set interface ethernet2 ip 1.1.1.1/30
set hostname sme-fwA
set vrouter trust-vr route 0.0.0.0/0 interface ethernet2 gateway 1.1.1.2
```

```
set nsrp monitor track-ip ip 1.1.1.2
set nsrp monitor track-ip ip 1.1.1.2 weight 255
set nsrp monitor track-ip
set nsrp vsd-group id 1 priority 1
# on firewall B, use these commands instead
set interface ethernet2 ip 2.2.2.1/30
set hostname sme-fwB
set vrouter trust-vr route 0.0.0.0/0 interface ethernet2 gateway 2.2.2.2
set nsrp monitor track-ip ip 2.2.2.2
set nsrp monitor track-ip ip 2.2.2.2 weight 255
set nsrp monitor track-ip
set nsrp vsd-group id 1 priority 2
#
```

From the Web Interface:

1. Go to **Network | Zones**.
2. Click **New** to create a new zone.
3. Enter **office** as the Zone Name.
4. Select **trust-vr** as the Virtual Router Name.
5. Click **OK**.
6. Click **New**.
7. Enter **internet** as the Zone Name.
8. Select **trust-vr** as the Virtual Router Name.
9. Click **OK**.
10. Go to **Network | Interfaces | ethernet1 | Edit**.
11. Enter **office** as the Zone Name.
12. Click **OK**.
13. Go to **Network | Interfaces | ethernet2 | Edit**.
14. Enter **internet** as the Zone Name.
15. Click **OK**.
16. Go to **Network | Interfaces | ethernet3 | Edit**.
17. Enter **HA** as the Zone Name.
18. Click **OK**.
19. Go to **Network | Interfaces | ethernet4 | Edit**.
20. Select **HA** as the Zone Name.
21. Click **OK**.

22. Go to **Network | NSRP | Cluster**.
23. Select **Cluster ID** and enter **1** as the ID.
24. Click **Apply**.
25. Go to **Network | NSRP | Monitor | Interface | VSD ID: Device | Edit Interface**.
26. Select **ethernet1**.
27. Set the Weight to **255**.
28. Select **ethernet2**.
29. Set the Weight to **255**.
30. Click **Apply**.
31. Go to **Network | NSRP | VSD Group**.
32. Click **Remove** for Group ID 0.
33. Confirm the prompt to delete it.
34. Click **New**.
35. Set the Group ID to **1**.
36. Click **OK**.
37. Go to **Network | Interfaces**.
38. Click **New** after selecting **VSI IF**.
39. Select **ethernet1** as the VSI Base.
40. Select **VSD Group 1**.
41. Assign it an IP Address and Netmask of **192.168.83.1/24**.
42. Click **Apply**.

Do the following on Firewall A:

1. Go to **Network | Interfaces | ethernet2 | Edit**.
2. Enter **1.1.1.1** as the IP Address, and **30** for the Netmask.
3. Click **OK**.
4. Go to **Network | DNS**.
5. Enter **sme-fwA** as the hostname.
6. Click **Apply**.
7. Go to **Network | Routing | Routing Entries**.
8. Click **New** after selected **trust-vr** as the virtual router.
9. Enter **0.0.0.0/0.0.0.0** as the Network Address/Netmask.

10. Select **Gateway** instead of Next Hop Virtual Router Name.
11. Pick **ethernet2** as the Interface.
12. Enter **1.1.1.2** as the Gateway IP Address.
13. Click **OK**.
14. Go to **Network | NSRP | Monitor | Track IP**.
15. Click **New**.
16. Enter **1.1.1.2** as the Track IP.
17. Set the Weight to **255**.
18. Click **OK**.
19. Go to **Network | NSRP | Monitor | Track IP | VSD ID: Device | Edit**.
20. Select **Enable Track IP**.
21. Click **Apply**.
22. Go to **Network | NSRP | VSD Group | Group ID 1 | Edit**.
23. Set the Priority to **1**.
24. Click **OK**.

Do the following on Firewall B:

1. Go to **Network | Interfaces | ethernet2 | Edit**.
2. Enter **2.2.2.1** as the IP Address, and **30** for the Netmask.
3. Click **OK**.
4. Go to **Network | DNS**.
5. Enter **sme-fwB** for the hostname.
6. Click **Apply**.
7. Go to **Network | Routing | Routing Entries**.
8. Click **New**.
9. Select **trust-vr** as the Virtual Router Name.
10. Enter **0.0.0.0/0.0.0.0** as the Network Address/Netmask.
11. Select **Gateway** instead of Next Hop Virtual Router Name.
12. Select **ethernet2**.
13. Enter **2.2.2.2** as the Gateway IP Address.
14. Click **OK**.
15. Go to **Network | NSRP | Monitor | Track IP**.
16. Click **New**.

17. Enter **2.2.2.2** as the Track IP.
18. Set the Weight to **255**.
19. Click **OK**.
20. Go to **Network | NSRP | Monitor | Track IP | VSD ID: Device | Edit**.
21. Select **Enable Track IP**.
22. Click **Apply**.
23. Go to **Network | NSRP | VSD Group | Group ID 1| Edit**.
24. Set the Priority to **2**.
25. Click **OK**.

Notes from the Underground…

Forcing Links Down on the Backup Firewall

In certain rare situations, you might find that having all of the links up (but inactive) on the backup firewall causes problems. This is sometimes seen in cases where the NetScreens connect directly to a router instead of going via a switch, and the router uses only link state to adjust its routing, thereby sending packets to the inactive interface, which will simply be dropped.

If you find yourself in this situation, the recommendation is to rethink the network design to avoid this scenario. If this is not possible, try rethinking it again. If you still can't avoid it, be prepared for some headaches as you work around it on the firewall. This can be done with the unset nsrp link-up-on-backup command.

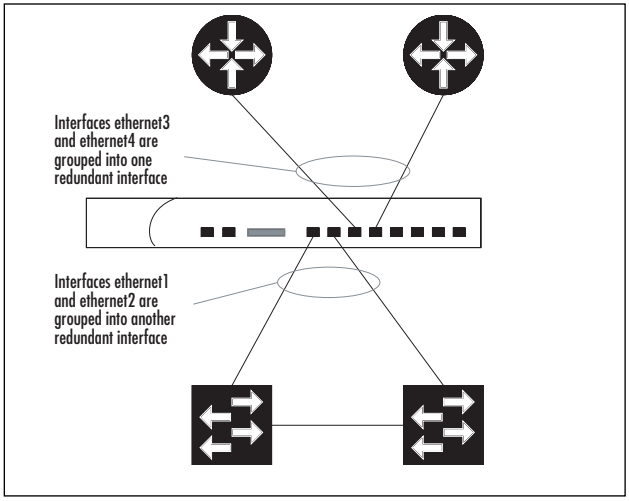
As the name implies, when this is unset, the links are brought down on the backup unit (except for the HA interfaces and the Management interface, if applicable). Until recently, this command was a hidden, undocumented command (it also existed in NSRPv1 as unset ha link-up-on-slave). The reasons for it being hidden and undocumented are many, including such things as: management via traffic interfaces becomes impossible; secondary NSRP paths are not available; failover time is greatly increased (especially in the presence of STP); and unless you are using a model that has a dedicated management interface, it is not possible to use NTP for clock synchronization (or any management features, for that matter).

Where at all possible, avoid using this feature, even if it is now a documented option.

Creating Redundant Interfaces

In addition to providing HA using redundant firewalls, it is also possible to use *redundant interfaces*. The idea behind a redundant interface is that failing over to a different firewall is disruptive and should be avoided. By cabling the firewalls so that each one has redundant links to each network segment (such as the case of a full-mesh setup), interface redundancy can be used. Instead of failing over all traffic onto a different firewall, it is only failed over onto a different interface on the same firewall (see Figure 12.21).

Figure 12.21 Cabling for Redundant Interfaces



Redundant interfaces are not a part of NSRP, but are commonly used together with NSRP to build full-mesh setups. However, it is possible to create redundant interfaces without enabling NSRP.

If you are using redundant interfaces with NSRP, you need to know that you can bind VSIs to the redundant interfaces, just as you would with a physical interface. By the same token, you can also keep a redundant interface as a local interface instead of a VSI.

Grouping Physical Interfaces into a Redundant Interface

To create a redundant interface, two or more physical interfaces are grouped together. Within a redundant interface group, one interface is considered the primary interface and will be used for sending and receiving traffic unless its link goes down. The secondary interface has its link up at all times, but is not active; traffic sent to it is simply discarded.

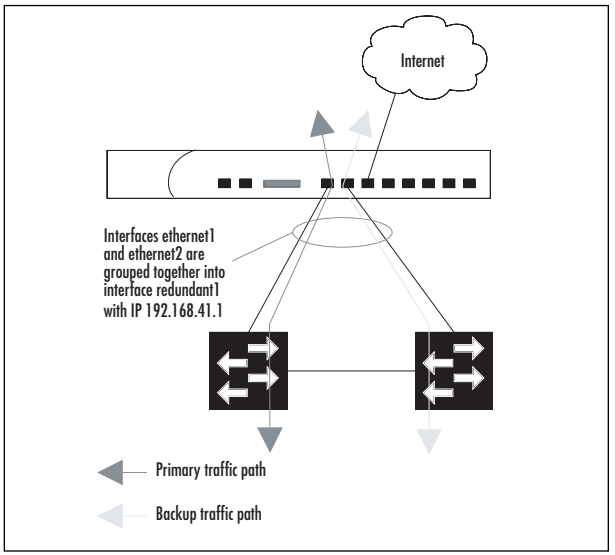
Once the redundant interface has been created, it can be configured like any other interface. It is assigned to a zone, given an IP address, and can be referred to in unnumbered

VPN tunnels. An interesting aspect worth noting is that physical interfaces do not need to be assigned to the same zone as the redundant interface.

Example: A Simple Redundant Interface Setup

To keep matters simple, consider only a partial setup in which redundant interfaces are used (see Figure 12.22). We will group the interfaces ethernet1 and ethernet2 together as a redundant interface, and bind the redundant interface to the *inside* zone.

Figure 12.22 Firewall with a Redundant Interface



With this configuration, all traffic normally passes over ethernet1. If that link fails, traffic is instantly moved to ethernet2.

From the CLI:

```
set interface redundant1 zone inside
set interface redundant1 ip 192.168.41.1/24
set interface ethernet1 group redundant1
set interface ethernet2 group redundant1
```

From the Web Interface:

1. Go to **Network | Interfaces | New Redundant IF**.
2. Enter **redundant1** as the interface name.
3. Enter **inside** as the Zone Name.
4. Specify **192.168.41.1/24** as the IP Address and Netmask.
5. Press **OK** to create the redundant interface.

6. Go to **Network | Interfaces | ethernet1 | Edit**.
7. Set As Member Of to **redundant1**.
8. Press **OK**.
9. Go to **Network | Interfaces | ethernet2 | Edit**.
10. Set As Member Of to **redundant1**.
11. Press **OK**.

Example: Changing the Primary Interface of a Redundant Interface

By default, the first physical interface added to the redundant interface group becomes the primary interface. It is possible to change this afterwards, if desired. For this example, we assume that interface redundant1 consists of physical interfaces ethernet1 and ethernet2, and that ethernet1 is currently the primary interface. To change the primary interface to ethernet2, use the following.

From the CLI:

```
set interface redundant1 phy primary ethernet2
```

From the Web Interface:

It is not possible to change the primary interface of a redundant interface from the Web interface.

Taking Advantage of the Full NSRP

By now we have explored most aspects of NSRP, but there are still a few important areas remaining. One of the shortcomings of NSRP-Lite is that in the case of failover, any existing session and VPN information (among other things) is lost. Juniper/NetScreen has an answer to this problem as well. It is called RTO mirroring, and is only available with the full NSRP.

A second shortcoming of NSRP-Lite is that in an NSRP cluster, one firewall ends up sitting unutilized for most of the time. It can be hard to justify the purchase of an additional firewall when management counters with the argument, "It says here that it will not actually be used. Why do you expect me to spend $$$ on something that will not be used?" The setups we have looked at so far are what are referred to as Active/Passive setups—one firewall is active and the other is passive. NSRP provides the ability to create Active/Active configurations, in which both firewalls actively handle traffic. Designing the network for Active/Active setups requires careful consideration, but once you have it set up and working, your network will run very nicely.

In this section, we also examine a full-mesh setup, where we combine just about everything that has been discussed in this chapter into one concrete example. Let's start by looking at RTO mirroring.

Synchronizing State Using RTO Mirroring

As mentioned throughout this chapter, a failover between two firewalls can be very disruptive—session information is lost, VPNs are brought down, DHCP leases go missing, and so on. The solution to this problem is RTO mirroring, which means that all of this dynamic information (RTOs) is actively mirrored between the VSDs. In case of a failover, all of this state information is already available on the new master VSD, and can resume operations with a minimum of interference to the traffic flow.

Enabling RTO mirroring is easy to do, but depends on the configurations being in sync between the firewalls. Before you attempt to enable RTO mirroring, it is useful to run the *exec nsrp sync global-config checksum* command to verify that the configurations indeed are in sync.

Example: Enabling RTO Mirroring in an NSRP Cluster

To create a cluster that is using RTO mirroring, the following is needed.

From the CLI:

```
set nsrp cluster id 1
set nsrp rto-mirror sync
```

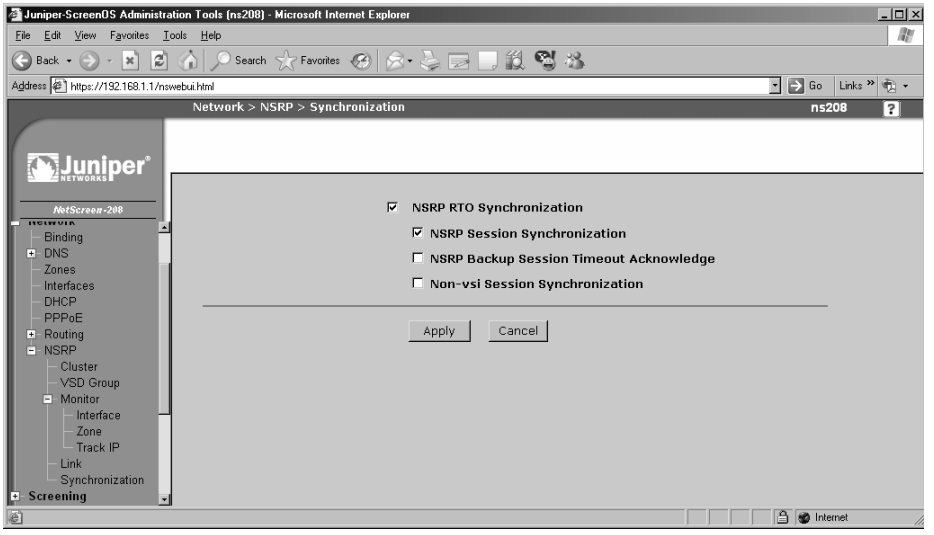
From the Web Interface:

1. Go to **Network | NSRP | Cluster**.
2. Enter **1** as the Cluster ID.
3. Click **Apply**.
4. Go to **Network | NSRP | Synchronization** (as shown in Figure 12.23).
5. Select **NSRP RTO Synchronization**.
6. Select **NSRP Session Synchronization**.
7. Click **Apply**.

Example: Preventing Certain Sessions from Being Backed Up

Earlier in this chapter, when discussing the SOHO range of NetScreens and their HA features, we mentioned the *no-session-backup* option that a policy can be tagged with. If you recall, we mentioned that this option has a different meaning when NSRP is used.

Figure 12.23 Enabling RTO Synchronization



Instead of the entire policy being inactive after a failover, under NSRP, this option indicates that any sessions created as a result of this particular policy should *not* be mirrored by the other VSD. Hence, if a failover occurs, those unmirrored sessions are dropped and must be reestablished after the failover. This can sometimes be useful when dealing with Denial-of-Service (DOS) attacks against insecure protocols.

Let's reuse the same policy; however, here it implies that any existing FTP sessions are dropped in case of a failover.

From the CLI:

```
set policy from trust to untrust any any ftp permit no-session-backup
```

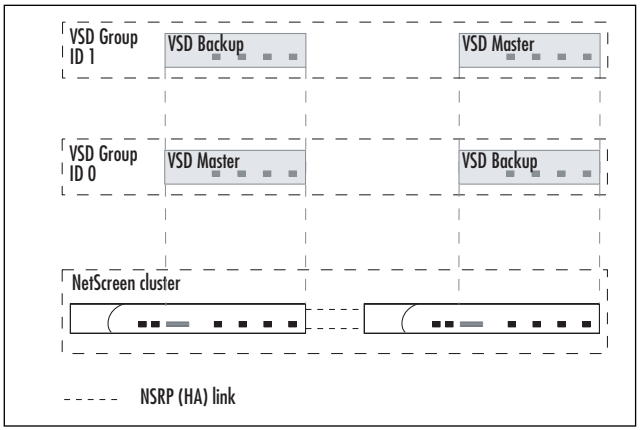
From the Web Interface:

1. Go to **Policies** and create a new policy from the **Trust** zone to the **Untrust** zone.
2. Select **any** as the source and destination address.
3. Select **FTP** as the service.
4. Set **Permit** as the action for the policy.
5. Click **Advanced**.
6. Deselect the **HA Session Backup** option, and click **Return**.
7. Click **OK** to save the new policy.

Setting Up an Active/Active Cluster

Setting up an Active/Active NSRP cluster is not that different from setting up an Active/Passive cluster. The difference is that in an Active/Active cluster you have more than one VSD group. You configure the VSD groups so that under normal circumstances the first firewall has the master VSD for the first VSD group and the backup VSD for the second VSD group, and vice versa. Spreading the VSDs out like this is a simple matter of setting their priorities correctly (see Figure 12.24).

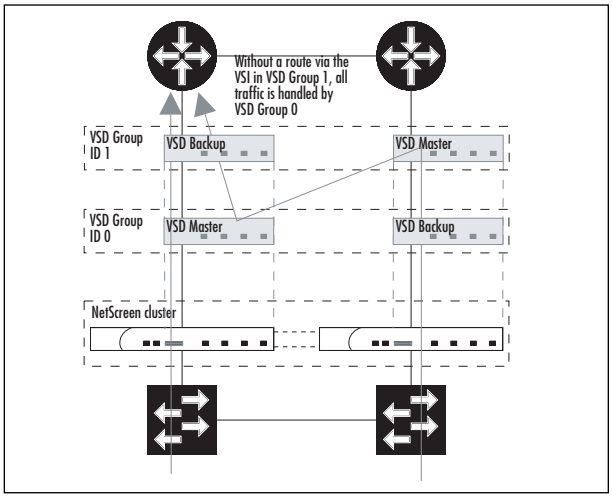
Figure 12.24 Multiple VSD Groups



Since each VSD contains its own set of VSIs, you end up with multiple IP addresses in each network, compared to a single IP address per network (not counting managed IP addresses) if you are using a single VSD. To benefit from this setup, you must configure the neighboring network nodes to load-balance between these IP addresses. With routers, this is commonly done by having two routes with identical cost, each pointing to one of the VSI IP addresses. If the NetScreens are providing the default gateway for a LAN, you can configure half the hosts on the LAN to use the IP address in the first VSD, and the second half to use the other VSD's IP address as their default gateway. This is the same approach as using a pair of VRRP routers as the default gateway.

An important thing to remember when setting up Active/Active configuration is that you must duplicate the routes as well. If you do not have routes for the second VSD, it is forced to forward all traffic to the first VSD across the HA data link, which should be avoided (see Figure 12.25).

Figure 12.25 Network with Missing Route in VSD Group 1

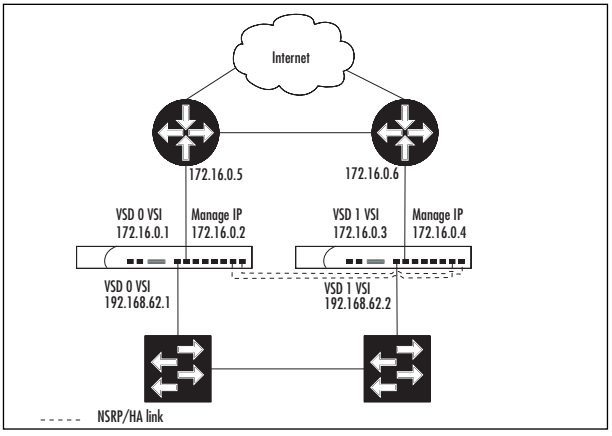


In addition to this, you must also take into account that the load you are putting on the firewalls may need to be handled by a single firewall. In other words, each firewall should run at 50-percent capacity (at the most) under normal conditions because in a failover scenario, everything goes through the remaining firewall; it cannot be expected to operate at more than 100 percent. If you attempt to put more traffic through it, be prepared to have some packets dropped.

Example: A Typical Active/Active Setup

Consider a scenario where you have two NetScreens providing a default gateway towards the Internet for an office LAN. The NetScreens are connected to redundant switches, as well as redundant Internet border routers (see Figure 12.26).

Figure 12.26 Network Layout for an Active/Active Setup



The client hosts in the office LAN have been configured so that half of them use the internal IP address of the first VSD, and the rest use the internal IP address of the second VSD as the default gateway. The routers run VRRP, and the NetScreens use ARP-based IP tracking of their respective router's IP address.

From the CLI:

```
set zone name office
set zone name internet
set zone office vrouter trust-vr
set zone internet vrouter trust-vr
set interface ethernet1 zone office
set interface ethernet1 ip 192.168.62.1/24
set interface ethernet1 route
set interface ethernet2 zone internet
set interface ethernet2 ip 172.16.0.1/29
set interface ethernet2 route
# on firewall A, use this command
set interface ethernet2 manage-ip 172.16.0.2
# on firewall B, use this command instead
set interface ethernet2 manage-ip 172.16.0.4
#
set interface ethernet7 zone ha
set interface ethernet8 zone ha
set nsrp cluster id 1
set nsrp cluster name deepthought
set nsrp vsd-group id 1
set interface ethernet1:1 ip 192.168.62.2/24
set interface ethernet2:1 ip 172.16.0.3/29
set vrouter trust-vr route 0.0.0.0/0 interface ethernet2
   gateway 172.16.0.5
set vrouter trust-vr route 0.0.0.0/0 interface ethernet2:1
   gateway 172.16.0.6
set nsrp monitor interface ethernet1
set nsrp monitor interface ethernet2
set nsrp vsd-group id 0 monitor track-ip ip 172.16.0.5
set nsrp vsd-group id 0 monitor track-ip ip 172.16.0.5 method arp
set nsrp vsd-group id 0 monitor track-ip ip 172.16.0.5 weight 255
set nsrp vsd-group id 0 monitor track-ip
set nsrp vsd-group id 1 monitor track-ip ip 172.16.0.6
set nsrp vsd-group id 1 monitor track-ip ip 172.16.0.6 method arp
set nsrp vsd-group id 1 monitor track-ip ip 172.16.0.5 weight 255
set nsrp vsd-group id 1 monitor track-ip
set nsrp rto-mirror sync
```

```
set nsrp secondary-path ethernet1
set nsrp vsd-group master-always-exists
set arp always-on-dest
# on firewall A, use these commands
set hostname deepthoughtA
set nsrp vsd-group id 0 priority 1
set nsrp vsd-group id 0 preempt
set nsrp vsd-group id 0 preempt hold-down 45
set nsrp vsd-group id 1 priority 2
# on firewall B, use these commands instead
set hostname deepthoughtB
set nsrp vsd-group id 0 priority 2
set nsrp vsd-group id 1 priority 1
set nsrp vsd-group id 1 preempt
set nsrp vsd-group id 1 preempt hold-down 45
#
```

From the Web Interface:

1. Go to **Network | Zones | New** to create a new zone.
2. Enter **office** as the Zone Name.
3. Select **trust-vr** as the Virtual Router Name.
4. Click **OK**.
5. Click **New** again.
6. Enter **internet** as the Zone Name.
7. Select **trust-vr** as the Virtual Router Name.
8. Click **OK**.
9. Go to **Network | Interfaces | ethernet1 | Edit**.
10. Enter **office** as the Zone Name.
11. Assign **192.168.62.1/24** to the IP Address and Netmask.
12. Select **Route** as the Interface Mode.
13. Click **OK**.
14. Go to **Network | Interfaces | ethernet2 | Edit**.
15. Enter **internet** as the Zone Name.
16. Assign **172.16.0.1/29** to the IP Address and Netmask.
17. Select **Route** as the Interface Mode.
18. Click **Apply**.

19. Enter **172.16.0.2** as the Managed IP on Firewall A, and **172.16.0.4** on Firewall B.
20. Click **OK**.
21. Go to **Network | Interfaces | ethernet7 | Edit**.
22. Select **HA** as the Zone Name.
23. Click **OK**.
24. Go to **Network | Interfaces | ethernet8 | Edit**.
25. Select **HA** as the Zone Name.
26. Click **OK**.
27. Go to **Network | NSRP | Cluster**.
28. Select **Cluster ID** and enter **1** as the ID.
29. Click **Apply**.
30. Go to **Network | NSRP | VSD Group**.
31. Click **New**.
32. Set the Group ID to **1**.
33. Click **OK**.
34. Go to **Network | Interfaces**.
35. Click **New** after selecting **VSI IF**.
36. Select **ethernet1** as the VSI Base.
37. Set the VSD Group to **1**.
38. Assign **192.168.62.2/24** to the IP Address and Netmask.
39. Click **Apply**.
40. Click **New** after selecting **VSI IF**.
41. Select **ethernet2** as the VSI Base.
42. Select VSD Group **1**.
43. Assign **172.16.0.3/29** to the IP Address and Netmask.
44. Click **Apply**.
45. Go to **Network | Routing | Routing Entries**.
46. Click **New**.
47. Select **trust-vr** as the Virtual Router Name.
48. Enter **0.0.0.0/0.0.0.0** as the Network Address and Netmask.
49. Select **Gateway** instead of Next Hop Virtual Router Name.
50. Select **ethernet2**.

51. Enter **172.16.0.5** as the Gateway IP Address.
52. Click **OK**.
53. Click **New**.
54. Select **trust-vr** as the Virtual Router Name.
55. Enter **0.0.0.0/0.0.0.0** as the Network Address and Netmask.
56. Select **Gateway** instead of Next Hop Virtual Router Name.
57. Select **ethernet2:1**.
58. Enter **172.16.0.6** as the Gateway IP Address.
59. Click **OK**.
60. Go to **Network | NSRP | Monitor | Interface | VSD ID: Device | Edit Interface**.
61. Select **ethernet1**.
62. Set the Weight to **255**.
63. Select **ethernet2**.
64. Set the Weight to **255**.
65. Click **Apply**.
66. Go to **Network | NSRP | Monitor | Track IP**.
67. Click **New**.
68. Enter **172.16.0.5** as the Track IP.
69. Set the Weight to **255**.
70. Change the Method to **ARP**.
71. Set the VSD Group ID to **0**.
72. Click **OK**.
73. Click **New**.
74. Enter **172.16.0.6** as the Track IP.
75. Set the Weight to **255**.
76. Change the Method to **ARP**.
77. Set the VSD Group ID to **1**.
78. Click **OK**.
79. Go to **Network | NSRP | Monitor | Track IP | VSD ID: 0 | Edit**.
80. Select **Enable Track IP**.
81. Click **Apply**.

82. Go to **Network | NSRP | Monitor | Track IP | VSD ID: 1 | Edit**.
83. Select **Enable Track IP**.
84. Click **Apply**.
85. Go to **Network | NSRP | Synchronization**.
86. Select **NSRP RTO Synchronization**.
87. Select **NSRP Session Synchronization**.
88. Press **Apply**.
89. Go to **Network | NSRP | Link**.
90. Select **ethernet1** as the Secondary Link.
91. Click **Apply**.

On Firewall A, do the following:

1. Go to **Network | DNS**.
2. Enter **deepthoughtA** as the hostname.
3. Click **Apply**.
4. Go to **Network | NSRP | VSD Group | Group ID 0 | Edit**.
5. Set the Priority to **1**.
6. Select **Enable Preempt**.
7. Set the Preempt Hold-Down Time to **45**.
8. Click **OK**.
9. Go to **Network | NSRP | VSD Group | Group ID 1 | Edit**.
10. Set the Priority to **2**.
11. Click **OK**.

Do the following on Firewall B:

1. Go to **Network | DNS**.
2. Enter **deepthoughtB** as the hostname.
3. Click **Apply**.
4. Go to **Network | NSRP | VSD Group | Group ID 0 | Edit**.
5. Set the Priority to **2**.
6. Click **OK**.
7. Go to **Network | NSRP | VSD Group | Group ID 1 | Edit**.
8. Set the Priority to **1**.

9. Select **Enable Preempt**.
10. Set the Preempt Hold-Down Time to **45**.
11. Click **OK**.

Not all settings can be adjusted from the Web interface. In this example, you need to enter the cluster name manually via the CLI, as well as enable the *nsrp master-always-exists* and *arp always-on-dest* options.

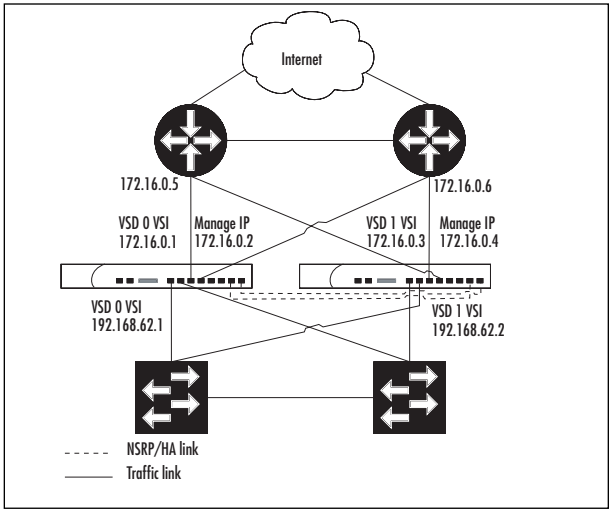
Implementing a Full-Mesh Active/Active Setup

The grand masterpiece of HA with NetScreen firewalls is the full-mesh Active/Active setup. Such a setup typically includes redundant interfaces, RTO mirroring, and multiple VSD groups. As such, it can be a handful to set up, and even more of a challenge to test and verify properly. To give you a place to start, we go through a sample full-mesh setup, which will give you a good outline of what is required. Please remember that every network is different; you must adjust the settings so that they are right for your network. Once you are done, remember to test, test, and further test your setup. You will not know if it all works until you have tested all possible scenarios. Tedious work for sure, but if you are on a pager for network support, it is work that will pay off quite well.

Example: A Full-Mesh Active/Active Setup

For this full-mesh example, we build on the previous example. The difference is that this time we have cabled the nodes in a full mesh, and therefore need to make configuration changes on the NetScreens. In particular, redundant interfaces must be introduced. (See Figure 12.27.)

Figure 12.27 A Fully Meshed Active/Active Setup



From the CLI:

```
set zone name office
set zone name internet
set zone office vrouter trust-vr
set zone internet vrouter trust-vr
set interface redundant1 zone office
set interface redundant1 ip 192.168.62.1/24
set interface redundant1 route
set interface ethernet1 zone office
set interface ethernet1 group redundant1
set interface ethernet2 zone office
set interface ethernet2 group redundant1
set interface redundant2 zone internet
set interface redundant2 ip 172.16.0.1/29
set interface redundant2 route
set interface ethernet3 zone internet
set interface ethernet3 group redundant2
set interface ethernet4 zone internet
set interface ethernet4 group redundant2
# on firewall A, use this command
set interface redundant2 manage-ip 172.16.0.2
# on firewall B, use this command instead
set interface redundant2 manage-ip 172.16.0.4
#
set interface ethernet7 zone ha
set interface ethernet8 zone ha
set nsrp cluster id 1
set nsrp cluster name deepthought
set nsrp vsd-group id 1
set interface redundant1:1 ip 192.168.62.2/24
set interface redundant2:1 ip 172.16.0.3/29
set vrouter trust-vr route 0.0.0.0/0 interface redundant2
   gateway 172.16.0.5
set vrouter trust-vr route 0.0.0.0/0 interface redundant2:1
   gateway 172.16.0.6
set nsrp monitor interface redundant1
set nsrp monitor interface redundant2
set nsrp vsd-group id 0 monitor track-ip ip 172.16.0.5
set nsrp vsd-group id 0 monitor track-ip ip 172.16.0.5 method arp
set nsrp vsd-group id 0 monitor track-ip ip 172.16.0.5 weight 255
set nsrp vsd-group id 0 monitor track-ip
set nsrp vsd-group id 1 monitor track-ip ip 172.16.0.6
```

```
set nsrp vsd-group id 1 monitor track-ip ip 172.16.0.6 method arp
set nsrp vsd-group id 1 monitor track-ip ip 172.16.0.5 weight 255
set nsrp vsd-group id 1 monitor track-ip
set nsrp rto-mirror sync
set nsrp secondary-path redundant1
set nsrp vsd-group master-always-exists
set arp always-on-dest
# on firewall A, use these commands
set hostname deepthoughtA
set nsrp vsd-group id 0 priority 1
set nsrp vsd-group id 0 preempt
set nsrp vsd-group id 0 preempt hold-down 45
set nsrp vsd-group id 1 priority 2
# on firewall B, use these commands instead
set hostname deepthoughtB
set nsrp vsd-group id 0 priority 2
set nsrp vsd-group id 1 priority 1
set nsrp vsd-group id 1 preempt
set nsrp vsd-group id 1 preempt hold-down 45
#
```

From the Web Interface:

1. Go to **Network | Zones | New** to create a new zone.
2. Enter **office** as the Zone Name.
3. Select **trust-vr** as the Virtual Router Name.
4. Click **OK**.
5. Click **New**.
6. Enter **internet** as the Zone Name.
7. Select **trust-vr** as the Virtual Router Name.
8. Click **OK**.
9. Go to **Network | Interfaces| New** after selecting **Redundant IF**.
10. Enter **office** as the Zone Name.
11. Assign **192.168.62.1/24** as the IP Address and Netmask.
12. Select **Route** as the Interface Mode.
13. Click **OK**.
14. Click **New** after selecting **Redundant IF**.
15. Enter **internet** as the Zone Name.
16. Assign **172.16.0.1/29** as the IP Address and Netmask.

17. Select **Route** as the Interface Mode.
18. Click **Apply**.
19. Enter **172.16.0.2** as the managed IP on Firewall A, and **172.16.0.4** on Firewall B.
20. Click **OK**.
21. Go to **Network | Interfaces | ethernet1 | Edit**.
22. Enter **office** as the Zone Name.
23. Set As Member Of to **redundant1**.
24. Click **OK**.
25. Go to **Network | Interfaces | ethernet2 | Edit**.
26. Enter **office** as the Zone Name.
27. Set As Member Of to **redundant1**.
28. Click **OK**.
29. Go to **Network | Interfaces | ethernet3 | Edit**.
30. Enter **internet** as the Zone Name.
31. Set As Member Of to **redundant2**.
32. Click **OK**.
33. Go to **Network | Interfaces | ethernet4 | Edit**.
34. Enter **internet** as the Zone Name.
35. Set As Member Of to **redundant2**.
36. Click **OK**.
37. Go to **Network | Interfaces | ethernet7 | Edit**.
38. Select **HA** as the Zone Name.
39. Click **OK**.
40. Go to **Network | Interfaces | ethernet8 | Edit**.
41. Select **HA** as the Zone Name.
42. Click **OK**.
43. Go to **Network | NSRP | Cluster**.
44. Select **Cluster ID** and enter **1** as the ID.
45. Click **Apply**.
46. Go to **Network | NSRP | VSD Group | New**.
47. Set the Group ID to **1**.
48. Click **OK**.

49. Go to **Network | Interfaces | New**.
50. Select **VSI IF**.
51. Select **redundant1** as the VSI Base.
52. Select VSD Group **1**.
53. Assign **192.168.62.2/24** as the IP Address and Netmask.
54. Click **Apply**.
55. Click **New** after selecting **VSI IF**.
56. Select **redundant2** as the VSI Base.
57. Select VSD Group **1**.
58. Assign **172.16.0.3/29** as the IP Address and Netmask.
59. Click **Apply**.
60. Go to **Network | Routing | Routing Entries| New**.
61. Select **trust-vr** as the Virtual Router Name.
62. Enter **0.0.0.0/0.0.0.0** as the IP Address and Netmask.
63. Select **Gateway** instead of Next Hop Virtual Router Name.
64. Select **redundant2**.
65. Enter **172.16.0.5** as the Gateway IP Address.
66. Click **OK**.
67. Select **trust-vr** as the Virtual Router Name.
68. Click **New**.
69. Enter **0.0.0.0/0.0.0.0** as the IP Address and Netmask.
70. Select **Gateway** instead of Next Hop Virtual Router Name.
71. Select **redundant2:1**.
72. Enter **172.16.0.6** as the Gateway IP Address.
73. Click **OK**.
74. Go to **Network | NSRP | Monitor | Interface | VSD ID: Device | Edit interface**.
75. Select **redudant1**.
76. Set the Weight to **255**.
77. Select **redundant2**.
78. Set the Weight to **255**.
79. Click **Apply**.

80. Go to **Network | NSRP | Monitor | Track IP | New**.
81. Enter **172.16.0.5** as the Track IP.
82. Set the Weight to **255**.
83. Change the Method to **ARP**.
84. Set VSD Group ID to **0**.
85. Click **OK**.
86. Click **New**.
87. Enter **172.16.0.6** as the Track IP.
88. Set the Weight to **255**.
89. Change the Method to **ARP**.
90. Set VSD Group ID to **1**.
91. Click **OK**.
92. Go to **Network | NSRP | Monitor | Track IP | VSD ID: 0 | Edit**.
93. Select **Enable Track IP**.
94. Click **Apply**.
95. Go to **Network | NSRP | Monitor | Track IP | VSD ID: 1 | Edit**.
96. Select **Enable Track IP**.
97. Click **Apply**.
98. Go to **Network | NSRP | Synchronization**.
99. Select **NSRP RTO Synchronization**.
100. Select **NSRP Session Synchronization**.
101. Press **Apply**.
102. Go to **Network | NSRP | Link**.
103. Select **redundant1** as the Secondary Link.
104. Click **Apply**.

Do the following on Firewall A:

1. Go to **Network | DNS**.
2. Enter **deepthoughtA** as the hostname.
3. Click **Apply**.
4. Go to **Network | NSRP | VSD Group | Group ID 0 | Edit**.
5. Set the Priority to **1**.
6. Select **Enable Preempt**.

7. Set the Preempt Hold-Down Time to **45**.
8. Click **OK**.
9. Go to **Network | NSRP | VSD Group | Group ID 1 | Edit**.
10. Set the Priority to **2**.
11. Click **OK**.

Do the following on Firewall B:

1. Go to **Network | DNS**.
2. Enter **deepthoughtB** as the hostname.
3. Click **Apply**.
4. Go to **Network | NSRP | VSD Group | Group ID 0 | Edit**.
5. Set the Priority to **2**.
6. Click **OK**.
7. Go to **Network | NSRP | VSD Group | Group ID 1 | Edit**.
8. Set the Priority to **1**.
9. Select **Enable Preempt**.
10. Set the Preempt Hold-Down Time to **45**.
11. Click **OK**.

Not all settings can be adjusted from the Web interface. You would need to enter the cluster name and enable the *nsrp master-always-exists* and *arp always-on-dest* options from the CLI.

Failing Over

A chapter on HA and NSRP would not be complete without a more in-depth dissection of what happens when a failover occurs. Things that can cause a failover are

- Software crashes (resulting in lost heartbeats)
- Hardware or power failure (resulting in lost heartbeats)
- Link failure on monitored interfaces or zones
- Unavailability of one or more tracked IP addresses
- Manually requested failover

Once the primary backup VSD has determined that it must become the master VSD, a few things happen. Firstly, the VSD promotes itself to master to prevent any other VSDs from doing the same thing. Second, if the VSD has any links down, an attempt is made to

bring them up. If a monitored link cannot be brought up, the VSD relinquishes its role as master and puts itself in the *inoperable* state. (See the "Avoiding the No-Brain Problem" section later in this chapter.)

Assuming the VSD is the newly promoted master VSD with all relevant links up, it proceeds to send out gratuitous ARP requests. This is a very important aspect of the failover. These ARP requests tell the neighboring network nodes that the IP addresses configured on the VSIs are now reachable via a different path than before. This will cause switches to update their forwarding tables, and routers to update their ARP tables. By default, four ARP packets are sent out on each interface, but this can be adjusted if needed (see the following example).

As soon as the neighboring nodes have adjusted to this change, traffic is sent to this VSD instead of the old one. If RTO mirroring was enabled before the failover, this VSD already has a copy of the *all run-time* state, and proceeds to handle traffic with no further disruption. Note that some packets may have been lost during the time it takes for the neighboring nodes to reroute their traffic flows to the second NetScreen.

Example: Adjusting the Number of ARP Packets Sent after Failover

If you discover that some network elements do not notice that the VSD has failed over, you might want to try increasing the number of ARP requests sent out after the VSD has failed over. In the following we increase it from the default four to nine.

From the CLI:

```
set nsrp arp 9
```

From the Web Interface:

1. Go to Network | NSRP | Cluster.
2. Enter **9** for Number Of Gratuitous ARPs To Resend.
3. Press **Apply** to save these settings.

Failing Over Virtual Systems

Ensuring that virtual systems (VSYSs) fail over takes some consideration and careful configuration. The key to making sure that a VSYS fails over correctly lies in remembering that only VSIs are moved across to the other VSD. Therefore, for a VSYS to fail over, its interfaces must be VSIs, not local interfaces (or variations built on a local interface).

Example: Binding a VSYS to VSD Group 1

To illustrate the concept of using VSIs for the VSYS, consider a scenario where an NSRP cluster has been configured, and VSD group 5 is in use. To set up the VSYS in a way that ensures that it fails over as expected (VLAN 42 on ethernet1/1 and VLAN 3 on ethernet2/1), we use the following configuration. The important thing to note here is that the IP addresses are assigned to the VSI, not to the sub-interface.

From the CLI:

```
set vsys vsys4
set interface ethernet1/1.4 tag 42 zone untrust    # Sub-interface
set interface ethernet1/1.4:5 ip 1.1.1.42/24       # VSI
set interface ethernet2/1.3 tag 3 zone trust-vsys4 # Sub-interface
set interface ethernet2/1.3:5 ip 192.168.102.1/24  # VSI
set interface ethernet2/1.3:5 route                # VSI
# more vsys configuration...
save
exit
```

From the Web Interface:

1. Go to **VSYS | New**.
2. Enter **vsys4** as the VSYS Name.
3. Press **OK**.
4. Go to **VSYS | vsys4 | Enter | Network | Interface**.
5. Press **New Sub-IF**.
6. Enter **ethernet1/1.4**.
7. Select the **untrust** Zone.
8. Enter **42** as the VLAN Tag.
9. Press **OK** to create this interface.
10. Return to **VSYS | vsys4 | Enter | Network | Interface**.
11. Press **New VSI IF**.
12. Enter **ethernet1/1.4**.
13. Choose VSD Group **5**.
14. Enter **1.1.1.42/24** as the IP Address and Netmask.
15. Press **OK**.
16. Return to **VSYS | vsys4 | Enter | Network | Interface**.
17. Press **New Sub-IF**.
18. Enter **ethernet2/1.3** as the Interface Name.
19. Select the **trust-vsys4** Zone.
20. Enter **3** as the VLAN Tag.
21. Press **OK**.
22. Return to **VSYS | vsys4 | Enter | Network | Interface**.
23. Press **New VSI IF**.

24. Enter **ethernet2/1.3** as the VSI Base.
25. Choose VSD Group **5**.
26. Use **192.168.102.1/24** as the IP Address and Netmask.
27. Select **Route** as the Interface Mode.
28. Press **OK**.

Avoiding the Split-Brain Problem

Consider a scenario where two NetScreens are cabled together in a HA setup with single or dual HA links. Ordinarily, these two units send heartbeats back and forth, verifying each other's state and agreeing on who should be the master unit. When this is happening, all is well, but what if for some reason the HA link(s) were disconnected? The NetScreens would no longer be able to talk to each other, and even though the loss of the HA links would not have a direct impact on the surrounding network, the resulting events will have an impact on the rest of the network—a very bad one.

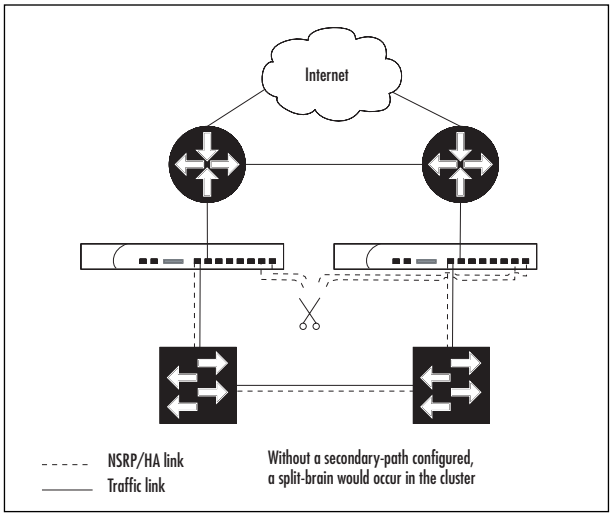
What happens in this scenario is that each of the NetScreens thinks the other unit fell over, and therefore there is only one NetScreen left, and hence this is the master. As such, the previous primary backup NetScreen promotes itself to master, and all of a sudden both firewalls are trying to persuade the surrounding network that it is the only place to send packets. Untold grief ensues, and sporadic connectivity through the firewalls is the best to be hoped for until at least one HA link is restored. This is commonly referred to as "split-brain."

Now that we know how and why it happens, let's look at ways we can mitigate the risk of this actually happening. The first option is to use dual HA links instead of a single link; it is less likely that both links fail simultaneously compared to a single link.

Second, if the HA links are connected via one of more switches, consider changing them to direct cross-over cables to avoid being dependent on the switch(es). Also, make sure you have enabled the HA link probes with the *set nsrp ha-link probe* command. Without the link probes, the firewalls may not be able to detect the link failure.

The third alternative is perhaps the most interesting, and warrants a more thorough discussion. In addition to one or two dedicated HA links, it is possible to specify a secondary path for the heartbeats. This secondary path is simply one of the traffic interfaces, which will be used as a fall-back option in case of total HA link failure (see Figure 12.28). Heartbeat packets will coexist with normal traffic on this interface. The important thing here is to ensure that the interface chosen is in the same Layer 2 broadcast domain as the corresponding interface on the second NetScreen. Or in plain English, the two interfaces must be part of the same VLAN. The reason for this is that the heartbeat packets are not IP packets, and are therefore not routable through the network. Generally, this works out well, since it is almost always the case for interfaces in the *Trust* (or similar) zone. This is also good for security reasons. You would not want to send the heartbeats via the *Untrust* (or similar) zone, since that part of the network is, by definition, not trusted.

Figure 12.28 HA Link Failure



Example: Configuring a Secondary NSRP Path

Adding a secondary path to an existing NSRP setup is very easy. For this example, if you have confirmed that interface ethernet1 on both firewalls meets the criteria for being used as a secondary path, the following settings will achieve your goal.

From the CLI:

```
set nsrp secondary-path ethernet1
```

From the Web Interface:

1. Go to **Network** | **NSRP** | **Link**.
2. Select **ethernet1** as the Secondary Link.
3. Press **Apply** to save these settings.

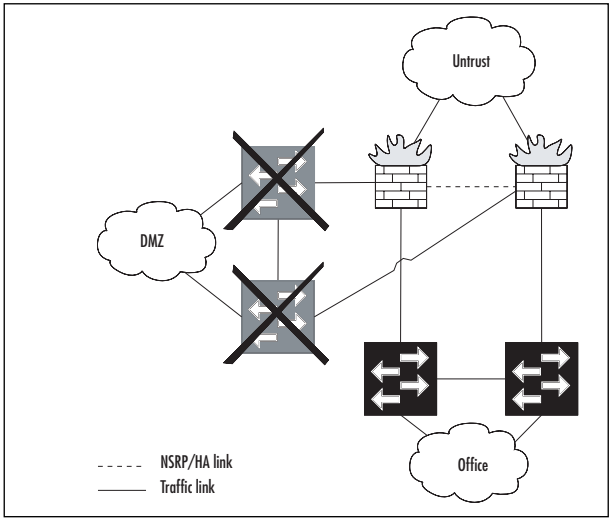
Avoiding the No-Brain Problem

The opposite problem to "split-brain" is "no-brain," where neither of the firewalls wants to be the master. This can be just as problematic (and infuriating) as having them both want to be the master. Let's first take a look at how we can find ourselves in this situation, and then work out how to avoid it from happening. The good news is that there is a definite way around this problem. The bad news is that it might not always yield the best possible result. This will become apparent as we work through the cause and the solution.

To illustrate how this problem can occur, consider the following scenario (Figure 12.29) where two NetScreens in an NSRP cluster are used to provide the office with Internet connectivity and to make services available from the demilitarized zone (DMZ). Aiming for a

highly available network, each NetScreen connects to a separate switch, both in the DMZ and in the *Trust* zone (we are not interested in how things look on the *untrusted* side for this scenario). To ensure that a failover occurs, you need the firewalls to monitor the appropriate interfaces. This is a decent setup that will handle most failure situations quite well. However, there is always Murphy's Law, and in this particular case it could manifest itself as a power failure to both of the DMZ switches. A bit contrived? Perhaps—but if you are a seasoned network engineer, you already know that the most unlikely things happen at the most inopportune moments.

Figure 12.29 Network with Redundant DMZ and Office Switches



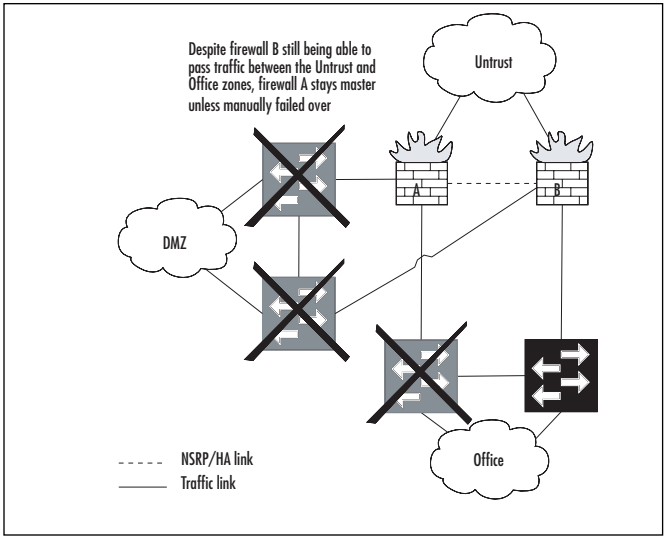
The failure of the DMZ switch that the current master firewall is connected to is detected by the firewall (thanks to interface monitoring), and goes into the *inoperable* state. This in turn prompts the backup NetScreen to promote itself to master, except for the fact that it just detected a link failure, and went into the *inoperable* state instead.

By now, you see the problem. Because of a failure that originally affected only the DMZ, the entire office is left without Internet connectivity. Not quite what you had in mind when you set up the HA network.

Fear not, for there is a way to avoid this. You can force an NSRP cluster to always have a master by using the *set nsrp vsd-group master-always-exists* command. The good news is that there will always be a master with this set. The bad news is the way that the master is elected. If both NetScreens have failed, the master is elected based on the preempt settings and the priority values, which means that while a master is elected, it may not be the best one for the job. As a point in case, consider the scenario (Figure 12.30) where one of the switches in the *Trusted* zone also lost power (in addition to the DMZ switches). If that happened to be the switch that the resulting master firewall is connected to, the office is still

without Internet connectivity, despite the fact that the second firewall would be able to pass the traffic. In this scenario, there is nothing that can be done other than manually failing over to the other NetScreen.

Figure 12.30 Network with Three Failed Switches



HA, like security itself, always boils down to the same thing—risk management and mitigation. There is no such thing as perfection, only a decreasing likelihood of things going wrong in a service-impacting manner. However, that should not discourage you from grabbing the cheap gains where possible. So if you find yourself with a network that runs a risk of running across this particular problem, then adding the *master-always-exists* option makes sense.

Configuring HA through NSM

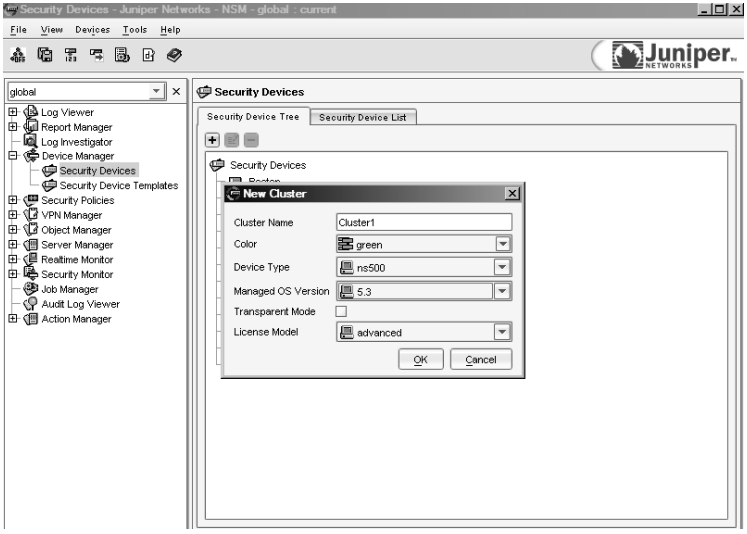
NetScreen Security Manager (NSM) provides intuitive and easy-to-use graphical user interfaces to manage a large number of firewall and IDP appliances from Juniper Networks. Let's have a brief look at how to configure HA using NSM. Refer to the documentation for the versions of Screen OS supported by a specific version of NSM. Prior to configuring the NetScreen devices through NSM, configure the devices to allow the specific management server to access them. To configure HA you need to create a cluster, add members to the cluster, configure NSRP parameters, and configure VSD.

Creating a Cluster

To create a cluster:

1. Start the NSM by double-clicking the icon on the desktop or by selecting from the Programs menu.
2. Provide the administrator name, password, and the IP address of the management server.
3. From the left pane, select **Device Manager** and click Security Devices. A list of security devices on your network appears on the right pane (Figure 12.31).

Figure 12.31 Adding an NSRP Cluster



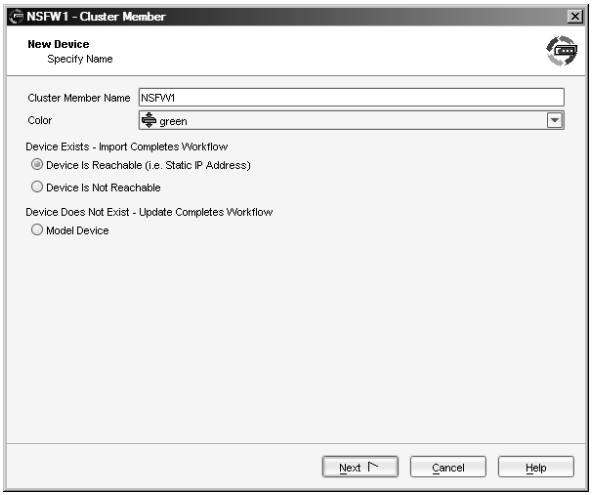
4. Click the + symbol to view a drop-down list. Select **Cluster**.
5. Provide the cluster name, the color to identify this object uniquely in the GUI, the device type, managed OS version, and the licensed model. Select the check box if your cluster runs on Transparent mode.
6. You can now right-click and select **Edit** to edit the cluster object you have created.

Adding Members to the Cluster

To add members to the cluster:

1. Click the + symbol to view a drop-down with the list of devices. Select **Cluster member**.
2. Provide the cluster member name, the color of the object on the GUI, and device reachability information (Figure 12.32).

Figure 12.32 Adding a Cluster Member



3. Click **Next**.
4. Choose the device type, OS version, and operating mode. Provide the IP address and the admin password (as shown in Figure 12.33). Do not modify the default SSH connection settings. Note that if the device is not reachable, the options are grayed-out (Figure 12.34).

Figure 12.33 Cluster Member Properties

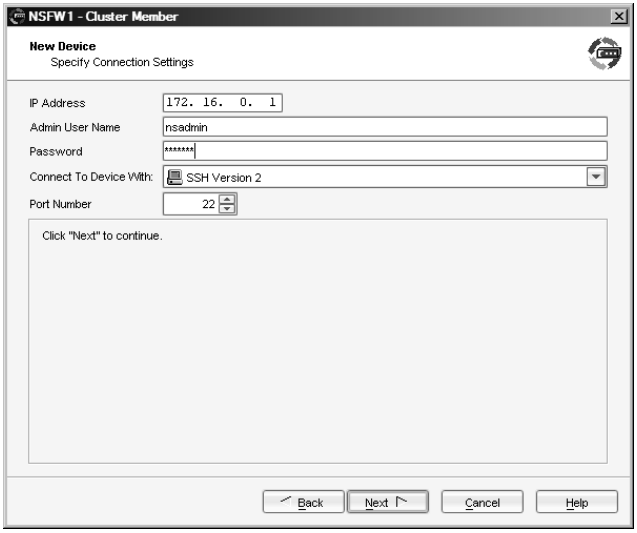
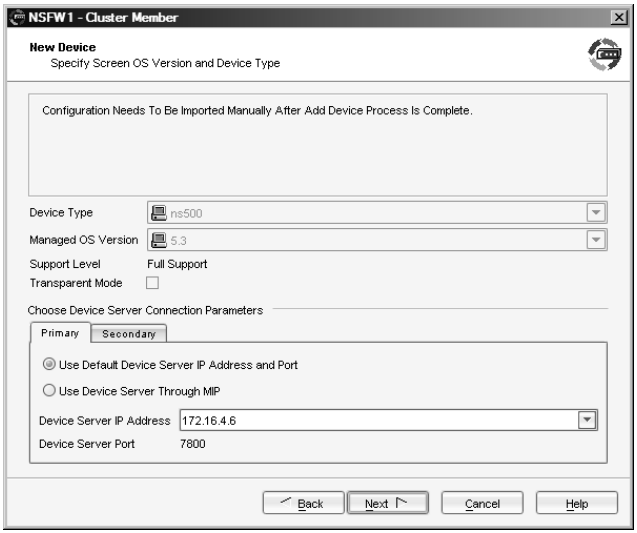
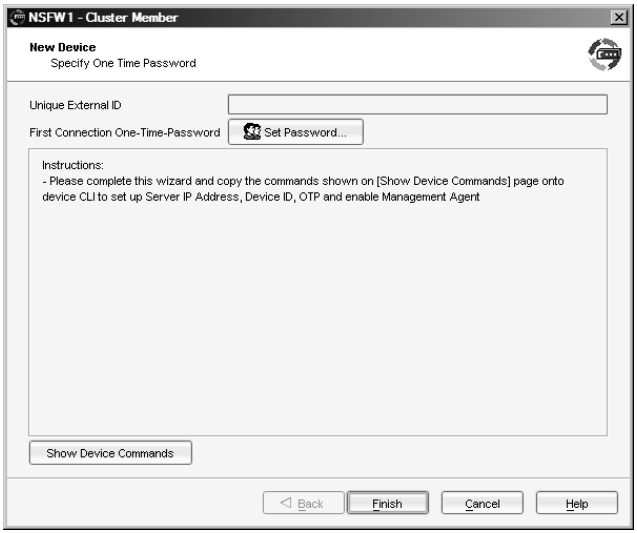


Figure 12.34 Configuring the Screen OS and Device Type



5. Click **Next**. You will find the Unique External ID and an option to set the one-time password. Set the password and make a note of it (Figure 12.35).

Figure 12.35 Setting the One-Time Password



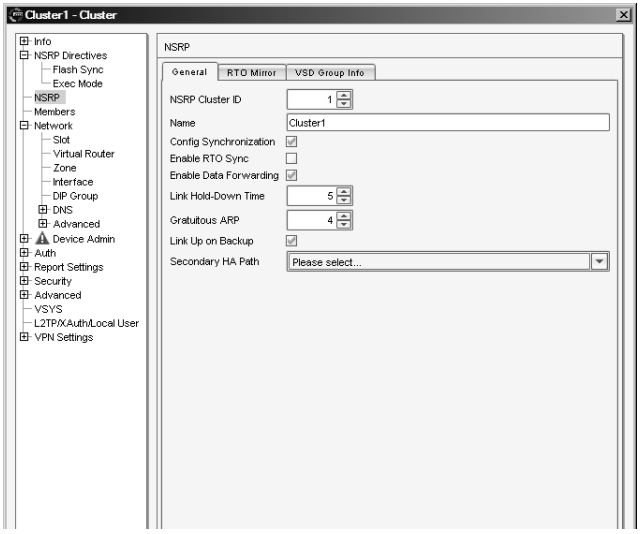
6. Repeat the preceding steps to add another member to the cluster.

Configuring NSRP Parameters

To configure NSRP parameters:

1. Under the Security Devices tree on the right pane, right-click and select **Edit** to edit the Cluster Object.
2. Click **NSRP**. On the General Tab, provide the cluster name, modify parameters such as Enable RTO Sync, Link Hold-Down Time, and the Gratuitous ARP parameters (see Figure 12.36).

Figure 12.36 Configuring NSRP Parameters



3. On the RTO Mirror tab, you can modify Heartbeat Interval, Heartbeat Threshold, and optionally enable or disable session synchronization and other session parameters (Figure 12.37).
4. You can also modify VSD parameters on VSD Group Info tabs. Select the Master-always-exist box to avoid any no-brain problems, as discussed earlier (Figure 12.38).

Figure 12.37 Configuring RTO Parameters

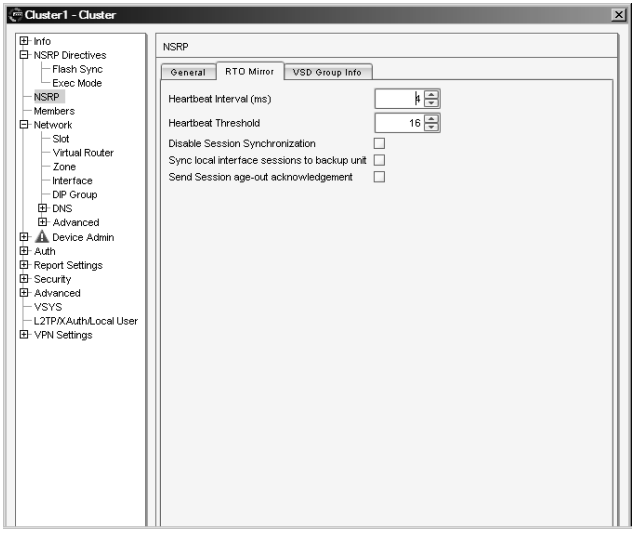
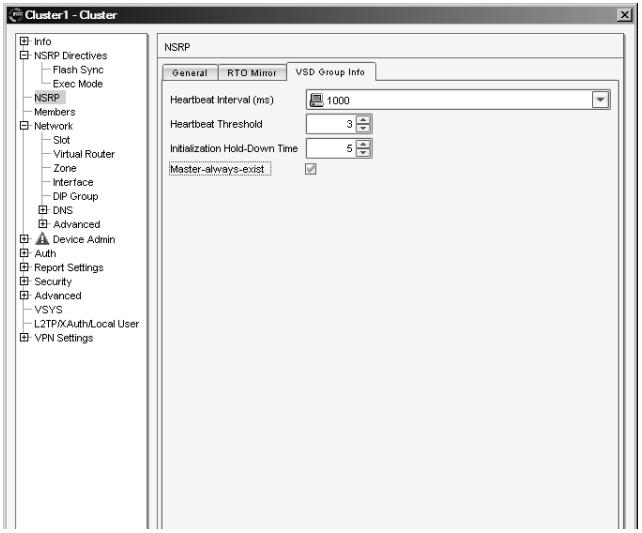


Figure 12.38 Configuring VSD Parameters



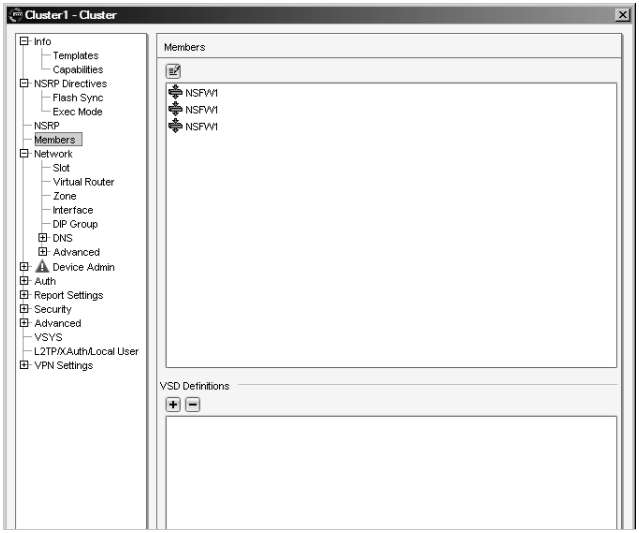
5. Click the Monitor option below NSRP on the left screen, and add Monitoring Interfaces by clicking +.

Configuring VSD

To configure VSD:

1. Under the Security Devices tree in the right pane, right-click and select **Edit** to edit the Cluster Object.
2. Click **Members** in the Cluster properties screen.
3. Click **+** to add VSD definitions (Figure 12.39).

Figure 12.39 Adding VSD Definitions



4. Modify VSD parameters as required.

Summary

HA is a hot topic, with more importance being placed on the availability of data networks across the globe. The NetScreen firewalls step up to the challenge by offering a wide range of different features designed for improving the overall availability of the networks in which they are deployed.

Many of the low-end SOHO firewalls come with features explicitly designed to make sure that Internet or other remote network connectivity is always available. While not spouting the same impressive range of features as the high-end NetScreen firewalls, the features provided are nevertheless highly useful and appropriate for the intended task of these firewalls. With the ability to automatically detect VPN failures and network host unavailability, and in that case automatically establish an alternative network path, the SOHO range leaves very little to be desired in this area.

On the midrange and high-end firewalls, the NSRP is available, which provides plenty of different options for improving network availability. The concept of duplicating hardware to increase availability is present with the NetScreen firewalls, in which two firewalls are grouped into an NSRP cluster. Enabling NSRP on a NetScreen automatically creates an abstraction of some parts of the firewall, referred to as a VSD. This abstraction has the ability to move between the cluster members as needed. By also abstracting the Ethernet interfaces into VSIs, NSRP provides virtual IP addresses not unlike those you get when using the VRRP on routers.

NSRP also provides a range of monitoring features that enable you to precisely track the state of the network, and when determined to be necessary, automatically fail over to a second firewall or even just a second interface, if you are making use of redundant interfaces. Interface link state, zone availability, and IP address availability are all things that can be monitored. The ability to configure the monitoring either on a device level or on a VSD level gives you great flexibility and enables you to tailor the failover behavior to something that is suitable for your network.

If you find yourself with a network design that requires different configurations on the NSRP cluster members to cater for differing network paths connected to the firewalls, you can achieve this by using local interfaces instead of VSIs. An IP address assigned to a local interface will not move between the cluster members, making it the ideal candidate for such a scenario.

To provide a minimum amount of disruption when a failover occurs, you can use RTO mirroring, which means that the standby firewall synchronizes the run-time state of the active firewall, thus ensuring that the standby firewall can take over traffic processing without missing a beat. At most, a few packets will need to be re-sent, but all sessions are kept alive.

By creating multiple VSDs, you can have both firewalls in an NSRP cluster processing traffic at the same time in an Active/Active setup, making the most of your hardware. Combining all of the HA features, you can achieve some remarkably available networks, as is the case of a fully meshed Active/Active setup.

Solutions Fast Track

The Need for High Availability

- HA is all about risk mitigation.
- The level of availability should be dictated by your business strategy.
- Finding the right balance between availability and cost is not always easy.
- Active/Passive, Active/Active, and Active/Active-Full mesh are various HA options available.

Improving Availability Using NetScreen SOHO Appliances

- Many SOHO appliances provide the ability to fail over between interfaces.
- Certain low-end appliances require an extended license to offer NSRP-Lite features.
- You can use either manual or automatic failover.
- Depending on the model, failover between two Ethernet interfaces or one Ethernet and the serial interface (with attached modem) is possible.
- You can restrict the policies to a subset while failed over the serial link.
- IP tracking can be used to determine when to fail over.
- You can track either the default gateway or explicit IP addresses.
- The state of VPNs can also be used to determine when to fail over. Remember to use the *monitor rekey* option if you are using automatic failover.

Introducing the NetScreen Redundancy Protocol

- NSRP is the protocol used between firewalls configured in redundant clusters.
- VSDs are the logical containers used when configuring NSRP.
- VSIs are logical interfaces belonging to a VSD, which are configured with the IP addresses that should be possible to transfer between the firewalls when a failover occurs.
- A VSD can be in any of the following states: Master, Primary Backup, Backup, Initial, Ineligible, or Inoperable.
- Where possible, use dual HA links.

Building an NSRP Cluster

- Several choices are possible in regards to the cabling of the NSRP cluster.
- Using directly connected HA links is preferred.
- To make a NetScreen part of an NSRP cluster, assign it a cluster ID between 1 and 7.
- Before the cluster can work properly, the configuration must be synchronized between the cluster members.

Determining When to Fail Over: The NSRP Ways

- NSRP automatically uses heartbeats between the cluster members to monitor each other.
- The interface link state can be used to determine failover.
- The monitoring of zones provides a slightly higher level of interface monitoring.
- IP tracking provides a flexible way to determine when to fail over.
- Use the ARP method if tracking a VRRP IP address.
- Remember that the weighting involved in IP tracking is different than that of interface and zone monitoring.

Reading the Output from *get nsrp*

- In addition to the *get nsrp* command, **get config | include nsrp** is a useful command to remember.
- Remember to distinguish between monitoring on device level (which affects all VSDs) and monitoring on VSD level.
- Use the *get nsrp vsd-group id X monitor* command to examine per-VSD monitoring.

Using NSRP-Lite on Midrange Appliances

- NSRP-Lite does not support RTO mirroring.
- Only Active/Passive setups are possible with NSRP-Lite.
- A local interface is an interface that is not bound to a VSI, and its IP address cannot fail over to another firewall.
- Local interfaces can be highly useful in setups with redundant but different traffic paths.

Creating Redundant Interfaces

- A redundant interface is created by grouping two or more physical interfaces together.
- Using redundant interfaces results in less risk of failing over between firewalls.
- Redundant interfaces are not dependent on NSRP being enabled.
- VSIs can be bound to a redundant interface, just as with a physical interface.

- Redundant interfaces are commonly used in full mesh setups.

Taking Advantage of the Full NSRP

- The use of RTO synchronization means that sessions will not be dropped in case of a failover occurring.
- In an Active/Active setup, both firewalls are actively processing traffic.
- Two VSD groups are used to achieve Active/Active setups.
- Equal-cost routes are commonly used for load balancing from the routers.
- If a cluster is providing the default gateway for a LAN, configure half of the hosts on the LAN to use the IP address of the first VSD, and the other half to use the IP address of the second VSD, in order to achieve rudimentary load sharing.
- Remember to provide routes for the second VSD, or you will most likely have problems.
- A full-mesh, Active/Active setup provides the best availability.

Failing Over

- When a failover occurs, the newly elected master VSD sends out gratuitous ARPs to announce the topology change to neighboring network nodes.
- In order for a VSYS to fail over, it must be using VSIs, not local interfaces or derivatives thereof.

Avoiding the Split-Brain Problem

- A split-brain is where the firewalls in an NSRP cluster have lost the HA link(s) and both assume that it is the only firewall available in the cluster, and therefore both promote themselves to master.
- Use dual HA links where possible.
- Connect the HA link(s) directly via crossover cables instead of via a switch.
- Specify a secondary NSRP path, using a traffic interface, with the *set nsrp secondary-path Ethernet X* command.

Avoiding the No-Brain Problem

- A no-brain scenario is where both firewalls in an NSRP cluster have determined that they are ineligible to become master, and the cluster is left without a master.
- To force an NSRP cluster to always have a master, use the *set nsrp vsd-group master-always-exists* command.
- When a master is elected in this scenario, only the preempt and priority settings are used; IP tracking and monitoring settings are ignored.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: I'm working on setting up an NSRP cluster, but I'm having trouble getting it to work. Both the HA links are up on both sides and the NetScreens are members of the same cluster ID, but they still don't see each other. Each thinks it's the master unit, and there is no backup listed in the *get nsrp* output. Why would this be?

A: You have managed to cross the two HA links. HA link #1 is connected to port A on the first NetScreen, but to port B on the second NetScreen and vice versa. This causes both NetScreens to receive each other's heartbeats on the HA data link, where it is ignored. Make sure you connect the HA links from port A to port A and port B to port B. What A and B are depends, of course, on the model of NetScreen you are using.

Q: One of the firewalls in my NSRP cluster is complaining about the configuration being out of sync. How did this happen and how do I fix it?

A: The most probable cause of this is that you made a configuration change to the cluster when this firewall was not running or was otherwise not part of the cluster. Therefore, that configuration change was not propagated across to this firewall. To remedy this, use the *exec nsrp sync global-config run* command. Make sure you save the configuration after doing this. If you want to verify that the configurations are indeed in sync, run the *exec nsrp sync global-config checksum* command.

Q: How do I force a failover?

A: On the current master device, execute the *exec nsrp vsd-group id X mode backup* command where *X* is the ID number of the VSD group (0 is the default VSD group if none have been specified).

Q: I am trying to set up an Active/Active NSRP cluster using a single HA link. I get no error messages, but all traffic from the LAN to the outside world going via the second VSD group is dropped. What is wrong?

A: You have most likely forgotten to add a default route for the second VSD. Since you do not have a HA data link, the packets cannot be forwarded between the VSDs, and are therefore dropped. Add a default route for the VSI in the second VSD, and traffic should flow normally.

Q: I have configured NSRP IP tracking, but it is not working. I do not see any pings sent out and the VSD fails over very quickly. What is up with this? Do I have a faulty Screen OS version?

A: You forgot to set a Managed IP on the interface used for sending the IP tracking packets. Remember that IP tracking packets cannot be sent from a VSI; they need either a managed IP or a local interface address. If you use the *get nsrp monitor track-ip* or *get nsrp vsd-group id X monitor track-ip* commands, you can see the statistics for the IP tracking as well as any error messages.

Chapter 13

Troubleshooting the Juniper Firewall

Solutions in this chapter:

- Troubleshooting Methodology
- Troubleshooting Tools
- Network Troubleshooting
- Debugging the Juniper Firewall
- Debugging NAT
- Debugging VPNs
- Debugging NSRP
- Debugging Traffic Shaping
- NetScreen Logging

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

Troubleshooting is a fact of life in computer networking, so this chapter covers the different ways to track the status of packets going through your firewall. Juniper firewalls offer a selection of tools to assist with troubleshooting network access. If you're already familiar with the troubleshooting tools available on the Juniper firewalls, you won't find many surprises when working with the new SSG product line since new features in the SSG firewall have similar troubleshooting tools, as well as a similar structure, to their NetScreen predecessors.

When dealing with network firewalls, it's important to remember that they often change the content of the packets going through them. So it's our task to keep track of the changes and make sure they are what we intended. Most firewalls have four main functions: packet forwarding, stateful filtering, address translation, and encryption. We tackle each of these functions differently. For instance, troubleshooting packet forwarding can be as easy as inspecting the routing table. Address translation may require looking at a log of the traffic. Troubleshooting encryption may require analysis of a detailed packet dump. Juniper firewalls, on the other hand, offer specific troubleshooting tools built in to the ScreenOS operating system. Here, we cover the different troubleshooting facilities—from *ping* to firewall *debug* commands—to help you understand the full arsenal of troubleshooting capabilities the Juniper firewall provides you.

Remember that every firewall issue is resolvable. There is a reason behind every decision the firewall makes. Thus, we begin this chapter by going through a common troubleshooting methodology which provides a solid process to help solve problems in an efficient manner. Next, we cover the various processes affecting a packet as it makes its way through the firewall. Then, we go over the different tools available for troubleshooting, along with the various firewall commands pertinent to troubleshooting. Following this introduction to troubleshooting commands, we discuss troubleshooting methods for VPNs (virtual private networks), NSRP (NetScreen Redundancy Protocol), and traffic shaping. Finally, we explore the logs the firewall creates in order to help us determine what the firewall is doing with our packets.

Troubleshooting Methodology

When a computer network isn't functioning the way you expect it to, the most valuable first step is often a sanity check. Is this a firewall issue? Are the packets making it to the firewall? Many firewall issues may actually be internal routing issues, so follow your packets from your computer through the internal network hubs, switches, and routers. It may be a good idea to sniff the traffic just outside of your firewall as well to see what the packets look like before they reach the firewall.

If you have narrowed things down to the firewall, make sure you cover the basics before you start digging into it. Be certain the power or network cables are properly connected and aren't damaged. Spend a few minutes ruling out any potential hardware problems before you begin working with the software. After all, if you end up calling Juniper support for help with the device, you don't want to be "that guy" who missed the bad cable.

Every troubleshooting session begins with a plan of action, so let's outline our own plan of action to help us figure out what went wrong. The following are seven steps to take when troubleshooting issues.

1. Describe the problem.
2. Describe the environment.
3. Determine the location of the problem.
4. Identify the cause of the problem.
5. Solve the problem.
6. Test the solution.
7. Document the changes.

We next go over each of these steps and describe how they can help us in the troubleshooting process.

Step One: Describe the Problem

Before we can start troubleshooting the process, we need to be able to describe the problem. (It's also important to tackle each problem individually in order to solve them more easily.) When doing this, try to narrow the scope of the problem by asking yourself several questions. Is this problem affecting all users of the device, a small group, or only one user? Is the problem affecting a certain operating system? You should try to find some commonality between affected users, and start your work from there.

Step Two: Describe the Environment

Next, we must be able to describe which network devices we are dealing with. This step includes listing the hardware and software involved in the path of the network traffic.

Step Three: Determine the Location of the Problem

The location of the problem isn't always apparent, but we need to determine where the problem is occurring. Several troubleshooting tools are available to us to help locate the problem. This step can be tricky since the problem might not be occurring where we thought.

Step Four: Identify the Cause of the Problem

Once we determine where the problem exists, we must identify its cause. This is normally done by analyzing the output of certain troubleshooting tools. The Juniper firewall comes with several troubleshooting tools built in to the device (something we will discuss in detail shortly). Of course, you can, and should, use other troubleshooting tools to help discover the cause of any potential problem.

Step Five: Solve the Problem

Once the cause of the problem has been identified, you need to actually resolve the problem. This might involve physically altering the network or issuing commands into network equipment by changing the configurations. Whatever you do, keep track of what you change.

Step Six: Test the Solution

Re-create the issue and see if the problem is fixed. Since the fix may affect other network traffic, make sure everything else is in working order as well.

Step Seven: Document the Changes

Documentation is one of the most important and most skipped steps. A good network administrator keeps a detailed log of what changes are made to the network infrastructure. Keeping track of what changes are made during troubleshooting is also important because the solution might create unintended problems in other areas of the network. Additionally, you may run into the same issue at another point in time, and there's no sense reinventing the wheel.

Troubleshooting Tools

The Juniper firewall has several troubleshooting tools built in to it. This section covers these tools in detail. Each has a specific purpose and should cover any troubleshooting needs you have.

Tools & Traps…

Secure Troubleshooting

One thing you want to make sure of when troubleshooting your firewall is that you don't compromise your security during the troubleshooting process. If you're using HTTP (Hypertext Transfer Protocol) or Telnet to access your firewall, someone may be able to sniff your packets while you're working to solve the problems.

The WebUI can be encrypted with SSL (Secure Sockets Layer) or tunneled through a VPN. It is recommended that this connection be secured at all times. The certificate can be self-signed by the Juniper firewall, so no certificate has to be purchased.

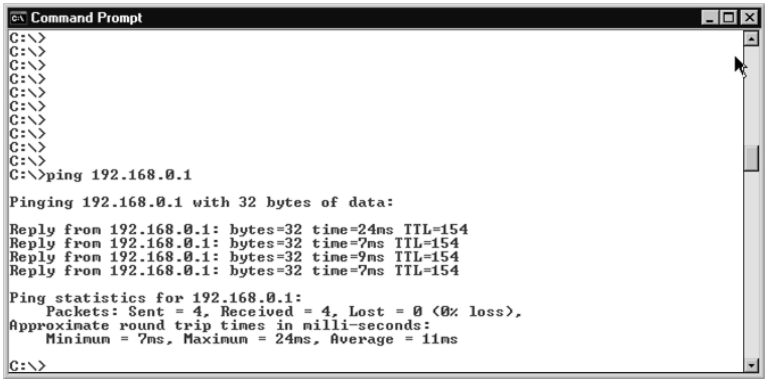
The command-line interface can be encrypted by using SSH (Secure Shell) to log in to your firewall. Telnet should be disabled so it cannot be used by anyone. If Telnet access is required for some reason, be sure to encrypt the packets using a VPN tunnel. Serial console access requires physical access to the firewall. You can disable all CLI access if you wish and require serial access to manage the box, but this measure might be a bit extreme.

Ping

Ping is probably the most well-known network troubleshooting utility in existence. The *ping* command is used to test for network connectivity. Every network operating system has a version of it preinstalled. It was written in December, 1983 by Mike Muuss for BSD Unix. The BSD Unix network stack has been ported to many operating systems, including every version of Microsoft Windows. Although the name was originally derived from a sonar analogy, it is now referred to as an acronym for Packet InterNet Groper.

The functionality is simple: send an ICMP (Internet Control Message Protocol) echo-request and wait for an ICMP echo-reply. The code shown in Figure 13.1 is an example of sending a ping to IP address 192.168.0.1, and getting four replies in return. This is a connectivity check from a Windows machine to a router.

Figure 13.1 The *ping* Command in Windows



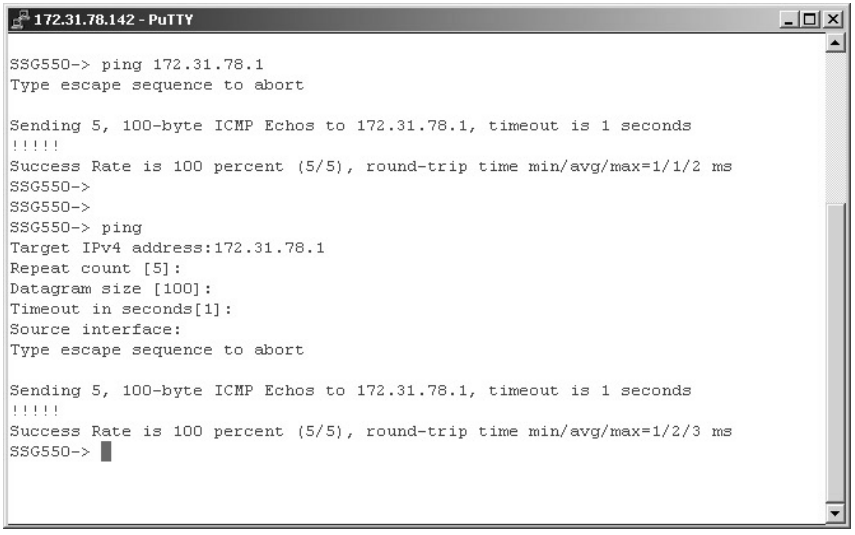
By default, the NetScreen device will send five ICMP echo requests of 100 bytes each with a two-second timeout. Advanced settings can also be included on the command line:

```
ping <address> from <interface name>
ping <address> count <number of pings to send>
```

You may also set all of the options manually by entering only the command ***ping*** and pressing **Enter**. At this point, you will be prompted for each one of the options to build the command you wish to execute, specifying target IP, the number of requests, the datagram size, and so on.

Figure 13.2 shows an example of using the *ping* command in ScreenOS 5.

Figure 13.2 The *ping* Command in ScreenOS-5



Keep in mind that the results of the *ping* command may not always be accurate. Some network traffic does not pass ping traffic and could possibly change the results of the command.

You can also ping from a specific interface with the ping command *ping <ipaddress> from <interfacename>* (see Figure 13.3)

Figure 13.3 Pinging from a Specific Interface

```
ns5gt-wlan-> ping 192.168.222.1 from trust
Type escape sequence to abort

Sending 5, 100-byte ICMP Echos to 192.168.222.1, timeout is 1 seconds from trust
!!!!!
Success Rate is 100 percent (5/5), round-trip time min/avg/max=43/100/299 ms
ns5gt-wlan->
```

traceroute

The *traceroute* command is useful in troubleshooting multihop routing. *traceroute* uses the TTL (Time to Live) field of the IP protocol to get an ICMP TIME_EXCEEDED response from each gateway the packet goes through to reach the destination. Figure 13.4 shows an example of *traceroute* in ScreenOS.

Figure 13.4 *traceroute* in ScreenOS

```
172.31.78.142 - PuTTY

SSG550-> trace-route syngress.com
Type escape sequence to escape

Send ICMP echos to syngress.com [155.212.56.73], timeout is 2 seconds,  maximum
hops are 32
1       3ms     5ms     3ms     172.31.78.1
2       3ms     4ms     4ms     172.31.67.1
3       3ms     4ms     4ms     66.129.242.2
4       5ms     5ms     5ms     65.206.180.109
5       6ms     5ms     6ms     152.63.91.246
6       74ms    62ms    60ms    152.63.27.29
7       60ms    60ms    60ms    152.63.29.193
8       62ms    62ms    62ms    65.200.139.145
9       65ms    65ms    66ms    209.113.217.116
10      72ms    72ms    72ms    209.113.217.229
11      182ms   79ms    77ms    216.41.101.99
12      74ms    74ms    74ms    209.113.217.126
13      76ms    76ms    76ms    155.212.56.65
14      78ms    77ms    77ms    155.212.56.73
Trace complete
SSG550->
```

```
SSG550-> trace-route 192.168.0.1
Type escape sequence to escape

Send ICMP echos to 192.168.0.1, timeout is 2 seconds,  maximum hops are 32
1       67ms     2ms      3ms      192.168.0.1
Trace complete
```

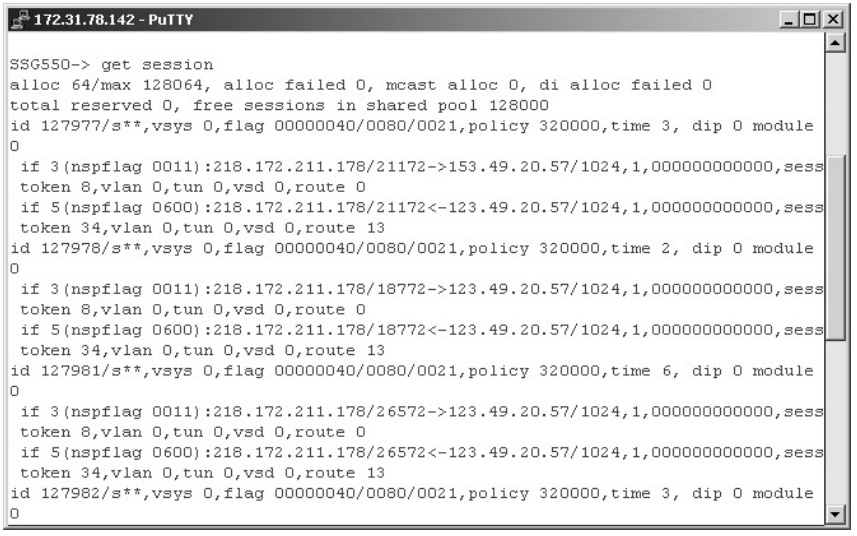
traceroute results should also be taken with a grain of salt. Since *traceroute* uses TTL fields in the packets, any devices that do not respond to that field will not return valid data.

Get Session

The *get session* command will show all current established sessions going through the Juniper firewall. If an entry exists in the session table, the connection has passed through the routing table and the policy successfully.

Each session entry has three lines of information. The first line contains the policy rule number, which can be viewed by the *get policy* command. The *time* entry shows idle time and resets every time traffic goes through the firewall. Figure 13.5 illustrates these points.

Figure 13.5 Get Session in ScreenOS



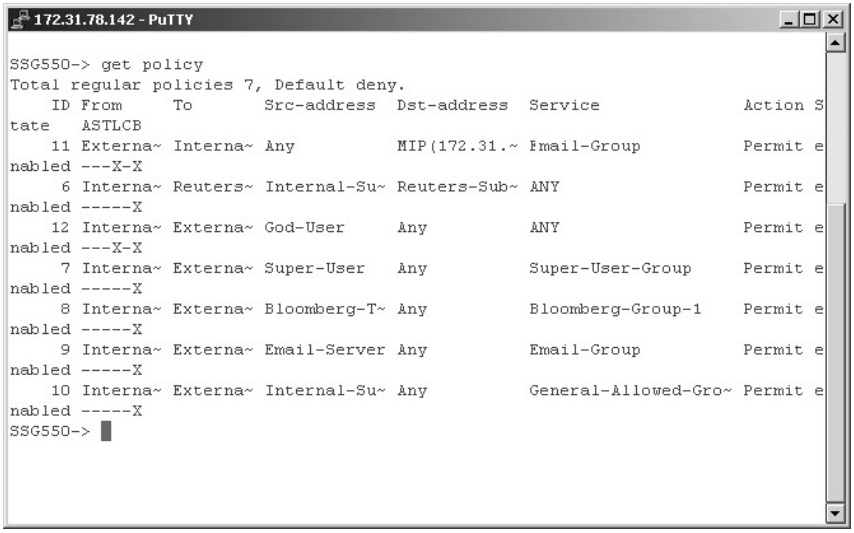
The output from the *get session* command can seem a bit overwhelming at first, but it isn't really that bad once you break it down. First, the command specifies how many sessions are currently allocated (in the preceding case, it is 64 with a maximum number of 128064). This command also specifies how many sessions failed to be allocated (both regular and DI sessions) and how many multicast sessions are allocated. It also provides statistics for the memory and sessions pools. The next part of the command that you really should be con-

cerned with is the information about the source IP address, source port, traffic direction, destination address, and destination port. The first entry in Figure 13.5 is: 218.172.211.178/18772->123.49.20.57/1024. This stands for a source address of 218.172.211.178, with a source port of 18772 going outbound to destination 123.49.20.57 port 1024. It will be using route 0, which you can verify with the *get route* command and compare that against the route ID value in the output. Traffic with the <- symbol designates the inbound (return) traffic. The return traffic may also show the NAT'd value of the packet, and the subsequent route which may be taken to reach the destination. You can also see which policy (in this case 320000) is being matched for this session.

Get Policy

The *get policy* command displays the current NetScreen policy. This command is useful as a reference to see which policy ID is assigned to each rule. Pay attention to the From and To fields. These indicate which zones each policy crosses, as shown in Figure 13.6.

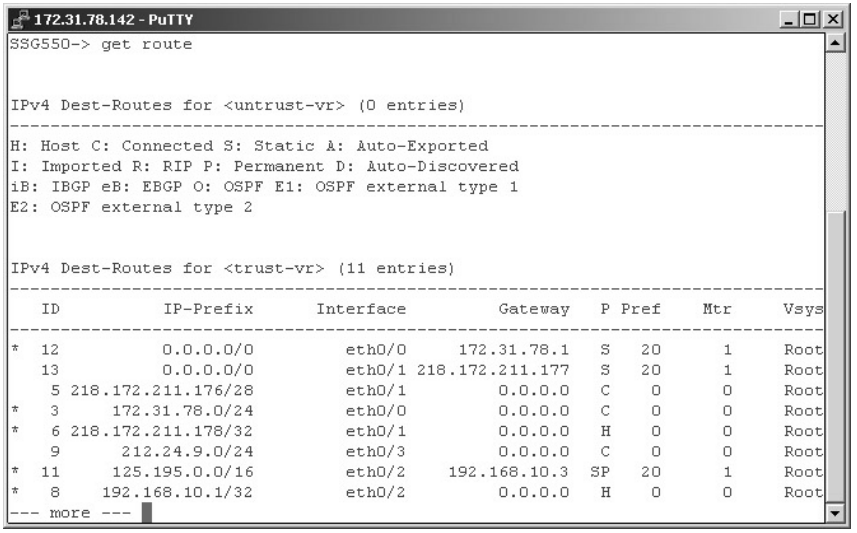
Figure 13.6 *get policy* in ScreenOS



Get Route

The *get route* command shows the current NetScreen routing table. There is a separate routing table for each virtual router. In the example in Figure 13.7, there are no routes for the untrust-vr, which is the default configuration. Make sure you differentiate which routes are static and which are added by a routing protocol.

Figure 13.7 *get route* in ScreenOS



Remember that the * next to a route designates that it is the active route in the routing table, and the ID is the value that is also referenced in other troubleshooting commands such as the *get session* command. This output shows you that route 12 is active over the same route (different next hop) route 13. They are both Static routes with a preference of 20, and a metric of 1. It is not immediately clear in this case why route 12 is valued higher than 13, but the reason could be because ethernet0/1 is physically down.

Get Interface

The *get interface* command shows detailed interface statistics. This command (shown in Figure 13.8) is useful to see which zone an interface is in and which hardware MAC (Media Access Control) address is assigned to each interface. You can also see the IP address, VLAN, and what state the interface is currently in (U for Up, D for Down.)

Get ARP

The ARP (Address Resolution Protocol) table of the Juniper firewall can be viewed by using the *get arp* command. This can be useful when troubleshooting OSI layer 1 and layer 2 issues. Figure 13.9 shows the ARP table of the Juniper firewall.

Figure 13.8 *get interface* in ScreenOS

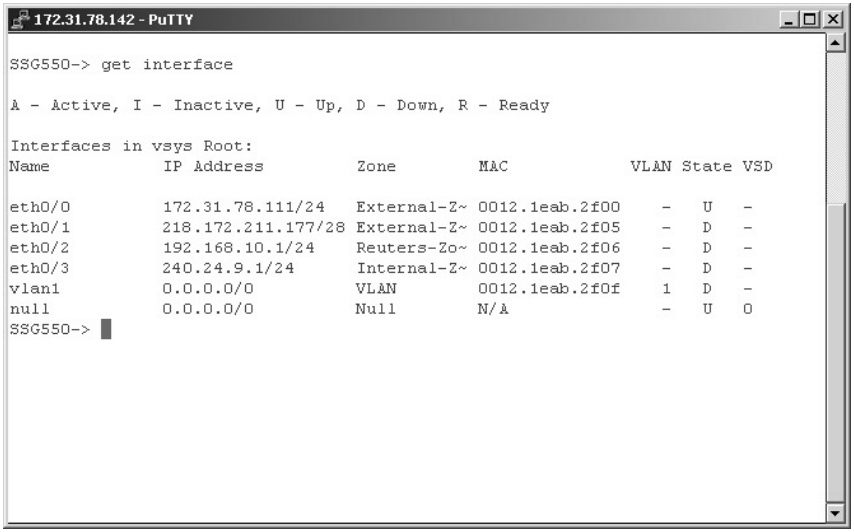
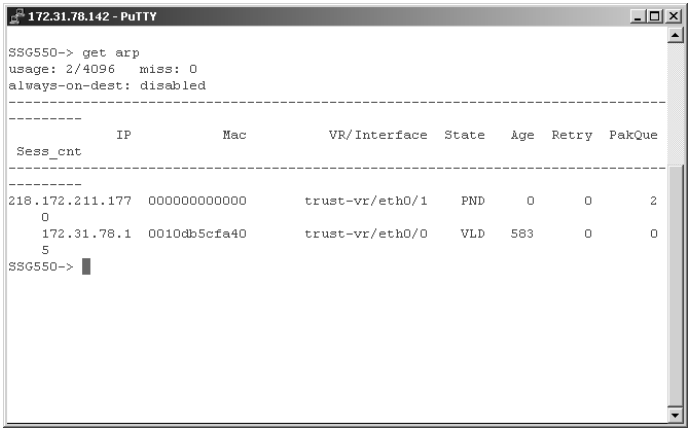
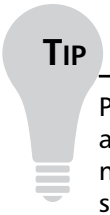


Figure 13.9 *get arp* in ScreenOS



We can see in this example that the MAC address for 218.172.211.177 is invalid (000000000000.) It also specifies what interface this will try to learn the MAC address on, which will be whatever interface has an IP address in the same subnet as the IP address that you are ARPing for. This can be very useful to troubleshoot layer 2 issues, especially when devices are connected directly to your firewall.



TIP

Please remember that if you are replacing one network gateway device with another (such as the SSG), the MAC address will change because there will be a new hardware interface in place of the old one (assuming you are keeping the same IP address). This will mean that other devices may not recognize this new MAC address until either their ARP cache times out (often 10 minutes on most systems), or you can manually clear it, such as issuing the *clear arp* on the Juniper firewall, or *arp –d* on Windows.

Get System

The *get system* command gives you several important pieces of information. Use this command to get an overview of your firewall and the setting for each interface. On an unknown firewall, this should be the first command you use.

- **Serial Number** This can be used to reset the device to the factory defaults. Use the serial number as the username and password when logging in on the serial interface. Be aware that this will also wipe out any configuration changes you have made. The serial number is used to generate the license keys for your device as well.
- **Software Version** The software version of the ScreenOS device in running memory.
- **Date and Time** Returns the date and time on the NetScreen device.
- **Total Device Resets** Tracks the total number of asset recovery resets. This number counts the number of times the system has been reset to the factory defaults.
- **User Name** The username of the current user.

```
ns5gt-wlan-> get system
Product Name: NetScreen-NS5GT-WLAN
Serial Number: 0129102005002244, Control Number: 00000000
Hardware Version: 1010(0)-(00), FPGA checksum: 00000000, VLAN1 IP (0.0.0.0)
Software Version: 5.4.0r1.0, Type: Firewall+VPN
Feature: AV-K
Compiled by build_master at: Tue Jul 18 21:22:51 PDT 2006
Base Mac: 0012.1eb3.4e30
File Name: ns5gt.5.4.0r1.0, Checksum: c2a2c761
```

```
Date 10/31/2006 22:18:01, Daylight Saving Time enabled
The Network Time Protocol is Enabled
Up 1775 hours 52 minutes 51 seconds Since 18Aug2006:23:25:10
Total Device Resets: 0

AP software version: 4.1.3.15.20
AP bootrom  version: 1.1

Regulatory Domain: FCC

Box in trust-untrust mode

System in NAT/route mode.

Use interface IP, Config Port: 8383
Mng Host IP: 192.168.2.0 255.255.255.0
User Name: admin451

Interface trust:
  description trust
  number 2, if_info 176, if_index 0, mode nat
  link up, phy-link up/full-duplex
  vsys Root, zone Trust, vr trust-vr
  dhcp client disabled
  PPPoE disabled
  admin mtu 0, operating mtu 1500, default mtu 1500
  *ip 192.168.2.1/24   mac 0012.1eb3.4e32
  *manage ip 192.168.2.1, mac 0012.1eb3.4e32
  route-deny disable
  bandwidth: physical 100000kbps, configured egress [gbw 0kbps mbw 0kbps]
             configured ingress mbw 0kbps, current bw 0kbps
              total allocated gbw 128kbps
Interface wireless1:
  description wireless1
  number 10, if_info 880, if_index 0, mode nat
  link up, phy-link up
  vsys Root, zone Wzone1, vr trust-vr
  PPPoE disabled
  admin mtu 0, operating mtu 1500, default mtu 1500
  *ip 192.168.225.1/24   mac 0012.1eb3.4e3a
  wireless AP mac 0012.1eb3.4e40
```

```
  *manage ip 192.168.225.1, mac 0012.1eb3.4e3a
  route-deny disable
Interface wireless2:
  description wireless2
  number 11, if_info 968, if_index 0, mode nat
  link down, phy-link up
  vsys Root, zone Trust, vr trust-vr
  PPPoE disabled
  admin mtu 0, operating mtu 1500, default mtu 1500
  ip 192.168.200.1/24   mac 0012.1eb3.4e3b
  wireless AP mac 0012.1eb3.4e41
  manage ip 0.0.0.0, mac 0012.1eb3.4e3b
  route-deny disable
Interface untrust:
  description untrust
  number 1, if_info 88, if_index 0, mode route
  link up, phy-link up/full-duplex
  vsys Root, zone Untrust, vr trust-vr
  dhcp client enabled
  PPPoE disabled
  admin mtu 0, operating mtu 1500, default mtu 1500
  *ip 68.62.10.4/21   mac 0012.1eb3.4e31
  gateway 68.62.8.1
  *manage ip 68.62.10.4, mac 0012.1eb3.4e31
  route-deny disable
  bandwidth: physical 100000kbps, configured egress [gbw 0kbps mbw 0kbps]
             configured ingress mbw 0kbps, current bw 0kbps
              total allocated gbw 256kbps
Interface serial:
  description serial
  number 6, if_info 528, if_index 0
  link down, phy-link down
  vsys Root, zone Null, vr untrust-vr
  admin mtu 0, operating mtu 1500, default mtu 1500
  *ip 0.0.0.0/0   mac 0012.1eb3.4e36
  bandwidth: physical 92kbps, configured egress [gbw 0kbps mbw 0kbps]
             configured ingress mbw 0kbps, current bw 0kbps
              total allocated gbw 0kbps
```

Debug

The debug utility in ScreenOS is a powerful troubleshooting tool that allows you to track sessions going through the Juniper firewall. The firewall has a memory buffer set aside for the debug system, and packets can be captured in this memory for inspection. The following outlines various uses of the debug system:

Step 1. Set any filters necessary for the debug. This is optional, but it might help consolidate the results. Optionally, you might also want to clear the buffer of old debugs so that you get a better snapshot.

Step 2. Issue the *Debug* Command.

Step 3. Issue the *get db str* command to get the output stored in the memory buffer from the debug.

Step 4. Stop the debug with the *undebug all* command which will halt any debugs. Alternatively you can keep issuing the *get db str* command to keep getting output from the debug.

Step 5. Clear the memory buffer with the *clear db* command.

WARNING

You must be mindful that issuing debug commands can increase the load on the firewall. Althought it is not as crippling as debugs on other platforms (historically,) it can cause problems if you are not careful. It is best to use flow filters, and turn the debugs off as soon as possible.

Flow Filters

A filter can also be put into place to limit what traffic gets sent to the debug buffer. The command *set ffilter* allows you to select the type of traffic to collect. The following filters are available:

- **dst-ip** Destination IP address
- **dst-port** Destination port
- **ip-proto** Internet protocol number
- **src-ip** Source IP address
- **src-port** Source port

If multiple filters are specified in the *set ffilter* command, the filter will only collect traffic that matches all of the filters specified. The *set ffilter* command can be executed multiple times, and traffic will be collected if it matches any of the filters. For example, to filter all tcp traffic from 192.168.0.1 to 10.1.1.1, issue the following command:

```
SSG550-> set ffilter src-ip 192.168.0.1 dst-ip 10.1.1.1 ip-proto 6
```

To view current filters, use the *get ffilter* command. Each filter in place has an ID number to identify it. To remove a filter, use the *unset ffilter* command, followed by the ID number of the filter to be deleted.

Snoop

Snoop is a full packet sniffer. The output of snoop goes into the same memory buffer that debug sends to. The biggest difference between debug and snoop is that snoop can dump the actual contents of the packets to the memory buffer. snoop output is more difficult to read than debug output and is typically used when the contents of the packets need to be analyzed. The following are the commands for using snoop:

- **snoop** Starts the snoop capture.
- **snoop info** Displays current snoop status.
- **snoop detail** Enables full packet logging. This logs the full contents of the packets.
- **snoop off** Turns off the snoop capture.
- **snoop filter** Allows you to filter what gets captured. Employs syntax similar to that used for debug filtering.
- **clear db** Clears the debug memory buffer.
- **get dbuf stream** Displays the output for analysis.

Firewall Session Analyzer (FSA)

Juniper has created a new Web-based tool called Firewall Session Analyzer (FSA) to help make sense of the torrent of information that can come from running a *get session* command. As discussed earlier, this command shows all current established sessions going through the NetScreen device, and this can seem a little daunting when viewed in the console.

After uploading a log of the *get session* command output to the FSA (located at http://tools.juniper.net/fsa/), it will generate the following seven reports.

- Rank based on destination IP address
- Rank based on destination port
- Rank based on source IP address

- Rank based on source port
- Rank based on protocol
- Rank based on Virtual System Device (VSD)
- Rank based on source IP with protocol and destination port information

In order to use the tool, you need to log the command output to a file on a TFTP server by using the following command.

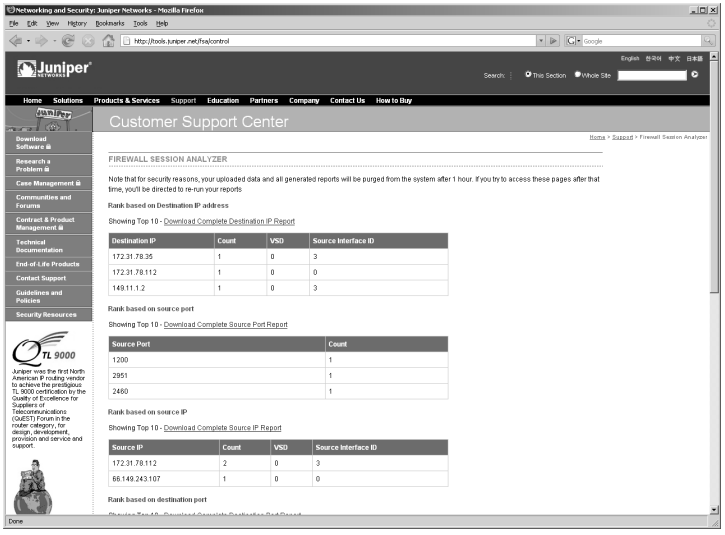
```
SSG550-> get session > tftp <server ip> <filename>
```

You may also choose to capture the screen output to a file and upload it to the analyzer in the same manner as the file stored on the TFTP server. Once you have the log file, generating the FSA reports is simple.

1. Go to http://tools.juniper.net/fsa/ using your Web browser.
2. Browse to your *get session* .log or .txt file, first making sure the file does not exceed 10MB.
3. Choose the version of ScreenOS the file was captured from (ScreenOS v4 or v5).
4. Click **Submit**. After several seconds, your results will be viewable in a new screen.

The top 10 results for each of the seven previous reports will be viewable on one page (see Figure 13.10), at which point you can download each complete report as an individual csv file by selecting the link for the report you desire. This information will be available for you to view for one hour following the execution of the analyzer. After one hour, the information processed by the tool, and the corresponding reports, will be deleted from the Juniper site for security reasons.

Figure 13.10 Firewall Session Analyzer



Putting It All Together

When troubleshooting the Juniper firewall, use any of the previous commands necessary to resolve the issue. When a packet arrives at an interface of the firewall, the following things happen.

1. The packet goes through a "sanity check" to make sure it isn't corrupt.
2. A session lookup is performed. If the packet is part of an existing session, it follows the rest of the packets in the same session.
3. The packet is routed, based on the routing table and zones.
4. The packet is checked against the firewall policy.
5. The ARP cache is referenced.
6. A session is created if one does not exist, and the packet is forwarded.

Notice that the session is not created until the packet passes through the routing table and the firewall policy.

Network Troubleshooting

Before you blame the firewall, you need to determine whether or not the firewall is actually the root of the problem. Several tools are available for network troubleshooting. The first thing you need is a decent packet sniffer. A packet sniffer is a network analyzer that grabs packets on the network and sometimes display them in a readable format. Ethereal (www.ethereal.com) is the best free sniffer available and will do the job nicely.

Make sure the firewall can ping the default gateway. The firewall should also be able to ping something on the Internet as well as internal resources. If the firewall cannot reach a host, it will be difficult for a packet to reach it after going through the firewall, unless there is another firewall blocking the traffic from the firewall itself.

Debugging the Juniper Firewall

When debugging traffic flowing through the Juniper device, we need to keep in mind that we might have more than one virtual router (VR) involved. ScreenOS has two VRs by default: a *trust-vr* and an *untrust-vr*. Each virtual router has its own routing table. By default, all of the zones, with the exception of the *null zone*, are associated with the trust-vr.

Let's start off with a Juniper firewall that has one virtual router in use, the default. Use the ***get route*** command to display the routing table. Make sure your added route belongs to the trust-vr. An asterisk (*****) will appear next to all routes in use. The most common problems are due to incorrect routing. The *debug flow basic* command is useful when troubleshooting issues with routing. Use it if you cannot figure out where the packets are going. Ask yourself the following questions:

- Do you have a default route? (A default route is a route to 0.0.0.0.)
- Is your default route going to the correct address and interface?
- Do you have a route to the network you are trying to get to? Do you need one?
- Is the added route going to the right interface?

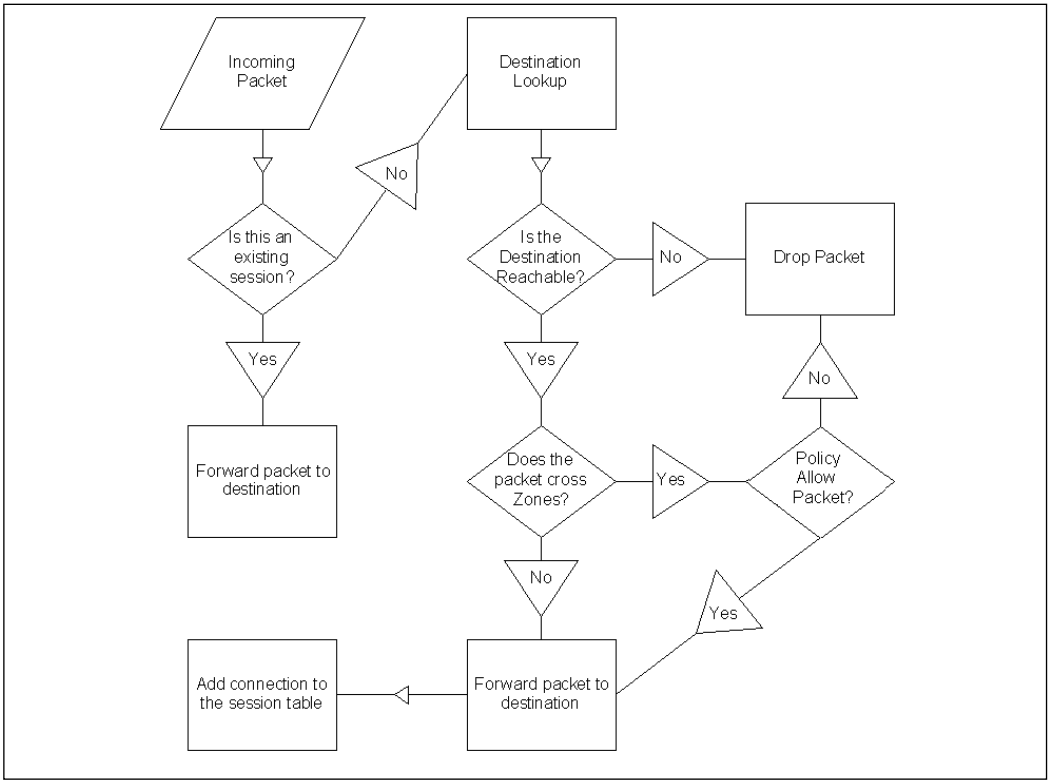
When troubleshooting issues with multiple virtual routers, it's important to remember that each virtual router needs a default gateway. For example, when using the *get route* command, look for a default route to the untrust-vr in the trust-vr. Also pay attention to added routes from routing protocols such as OSPF (Open Shortest Path First) and RIP (Routing Information Protocol). Are you getting OSPF routes added to the wrong virtual router?

Once the routing is taken care of, the next thing to look at is the policies. Use the command *get policy* to make sure your packets are going to the correct zone based on the policy. Take note of the policy ID number so you can reference it when using the *debug* command if needed. Intra-zone blocking may be enabled, which requires a policy for traffic to pass. You can see if this setting is enabled by looking at the zone details by using *get zone trust*, for example.

You can also use the *get policy global* command to see if any global policies are getting in the way. The global zone is referenced if the packets pass through all the rules of the specific zone policy, or there happen to be no rules in the specific zone policy. Adding a global zone rule that logs all dropped traffic can aid the troubleshooting process.

Every packet that goes through the Juniper firewall follows a path based on internal decisions that are, in turn, based on the contents of the packet and where it came from. When a packet arrives at an interface of the Juniper firewall, the firewall first checks to see if the packet is part of an existing session. If it is, the packet is forwarded out the predetermined interface. If the packet is not part of an existing session, it is processed through the routing table. If the packet is not routable, it is dropped. If it *is* routable, a zone lookup is performed. If the packet does not traverse a zone, it is then forwarded to its destination and added to the session table. If the packet does cross a zone, however, a policy lookup is done. Then the packet is either dropped or added to the session table and forwarded. This is a simplified description of what happens when a packet is inspected, but it helps for troubleshooting purposes. Figure 13.11 shows a flowchart of the process that the Juniper firewall follows with each packet that arrives at an interface.

Figure 13.11 The NetScreen Traffic Process Flowchart



Damage & Defense…

Deep Inspection

Deep Inspection (DI) allows the Juniper firewall to inspect layer 7 data in packets that transverse the firewall. Keep your DI signatures up-to-date to make sure you are making the best of this feature. DI signatures can be configured to automatically update.

DI does not replace a full-blown Intrusion Prevention System (IPS). A properly secured network should be multilayered and have security measures at every hop throughout the network. Make sure your IPS system is monitoring every segment of your network.

Tracing a Debug

When all else has failed, a debug will tell you what decisions are made on a step by step basic on the firewall. We will first issue a flow filter to help reduce other output that is unnecessary.

```
ns5gt-wlan-> set ff dst-ip 72.14.203.104
filter added
ns5gt-wlan-> clear db
ns5gt-wlan-> debug flow basic
ns5gt-wlan-> get db str
****** 6393831.0: <Wzone1/wireless1> packet received [60]******
  ipid = 32716(7fcc), @01c29494
  packet passed sanity check.
  wireless1:192.168.225.103/6156->72.14.203.104/1536,1(8/0)<Root>
  no session found
  flow_first_sanity_check: in <wireless1>, out <N/A>
  chose interface wireless1 as incoming nat if.
  flow_first_routing: in <wireless1>, out <N/A>
  search route to (wireless1, 192.168.225.103->72.14.203.104) in vr trust-vr for
vsd-0/flag-0/ifp-null
  [ Dest] 752.route 72.14.203.104->68.62.8.1, to untrust
  routed (x_dst_ip 72.14.203.104) from wireless1 (wireless1 in 0) to untrust
  policy search from zone 16-> zone 1
 policy_flow_search  policy search nat_crt from zone 16-> zone 1
  RPC Mapping Table search returned 0 matched service(s) for (vsys Root, ip
72.14.203.104, port 12112, proto 1)
  No SW RPC rule match, search HW rule
  Permitted by policy 33
  choose interface untrust as outgoing phy if
  no loop on ifp untrust.
  session application type 0, name None, nas_id 0, timeout 60sec
  service lookup identified service 0.
  flow_first_final_check: in <wireless1>, out <untrust>
  existing vector list 201-35a3630.
  Session (id:2000) created for first pak 201
  flow_first_install_session======>
  route to 68.62.8.1
  arp entry found for 68.62.8.1
  nsp2 wing prepared, ready
  cache mac in the session
  make_nsp_ready_no_resolve()
```

```
  search route to (untrust, 72.14.203.104->192.168.225.103) in vr trust-vr for
vsd-0/flag-3000/ifp-wireless1
  [ Dest] 3.route 192.168.225.103->0.0.0.0, to wireless1
  route to 192.168.225.103
  flow got session.
  flow session id 2000
  update policy out counter info.
 flow_send_vector_, vid = 0, is_layer2_if=0
  send packet to traffic shaping queue.
  flow_ip_send: 7fcc:68.62.10.109->72.14.203.104,1 => untrust(60) flag 0x20000,
vlan 0
 pak has mac
  Send to untrust (74)
****** 6393831.0: <Untrust/untrust> packet received [60]******
ns5gt-wlan-> undebug all
```

As we can see in this output, we began by setting a flow filter (in this case) based on the destination IP address to help reduce the output. We then cleared the memory buffer to help reduce unnecessary output. We then issued the *debug flow basic* command, which is used to give step-by-step output of every packet decision in the firewall (it should look familiar to the packet flow diagrams that we presented earlier). First, we see that the packet was received from zone WZone1, interface Wireless1. The firewall performs a sanity check on the packet, which involves examining it with the SCREEN features to help prevent certain types of attacks and address spoofing. Next, we see the 192.168.225.103/6156->72.14.203.104/1536 output, which specifies the source address, source port, -> (outbound) to 72.14.203.104 (the IP address that we set in our flow filter to match as a destination address) with a destination port 1536. We then see that no session currently exists. If a session did exist, the packets would bypass some of the steps to determine what proper action should be taken for this packet. Note that this does not bypass any security checks, just session establishment steps. Next we search for a route (since we are in route mode, transparent mode would search for an ARP entry). This helps us determine what zone the traffic is destined to go to (since we already know what zone it arrived on). We now know the route, and thus the direction, from zone 16 to zone 1. You can issue the *get zone* command to show you what zones have what IDs in the system. In our case Zone 16 is WZone1 and Zone 1 is Untrust. We then match the policy starting at the top of the Wzone1 to Untrust ruleset and go down until we hit the policy that matches this traffic. This matches policy 33 (you can run a *get policy id x* to see what the policy is on your system). We now know what the outgoing interface should be, and we can determine if there is any NAT. We then perform the final route lookup to determine how to route the traffic outbound and then perform an ARP lookup for the next hop, which in this case was 68.62.8.1. Assuming that we have that, the packet is set into the ether.

When you debug on VPN traffic, the output looks slightly different as you can see from the following output:

```
ns5gt-wlan-> get db str
****** 6394753.0: <Wzone1/wireless1> packet received [60]******
  ipid = 34184(8588), @01c1c294
  packet passed sanity check.
  wireless1:192.168.225.103/26894->192.168.222.80/1536,1(8/0)<Root>
  no session found
  flow_first_sanity_check: in <wireless1>, out <N/A>
  chose interface wireless1 as incoming nat if.
  flow_first_routing: in <wireless1>, out <N/A>
  search route to (wireless1, 192.168.225.103->192.168.222.80) in vr trust-vr for
vsd-0/flag-0/ifp-null
  [ Dest] 9.route 192.168.222.80->0.0.0.0, to tunnel.1
  routed (x_dst_ip 192.168.222.80) from wireless1 (wireless1 in 0) to tunnel.1
  policy search from zone 16-> zone 1
 policy_flow_search  policy search nat_crt from zone 16-> zone 1
  RPC Mapping Table search returned 0 matched service(s) for (vsys Root, ip
192.168.222.80, port 56909, proto 1)
  No SW RPC rule match, search HW rule
  Permitted by policy 7
  No src xlate ## 2006-10-31 22:44:23 : NHTB entry search no found: vpn none tif
tunnel.1 nexthop 192.168.222.80
  choose interface untrust as outgoing phy if
  no loop on ifp untrust.
  session application type 0, name None, nas_id 0, timeout 60sec
  service lookup identified service 0.
  flow_first_final_check: in <wireless1>, out <untrust>
  existing vector list 205-35a2900.
  Session (id:1938) created for first pak 205
  flow_first_install_session======>
  cache mac in the session
  make_nsp_ready_no_resolve()
  search route to (tunnel.1, 192.168.222.80->192.168.225.103) in vr trust-vr for
vsd-0/flag-3000/ifp-wireless1
  [ Dest] 3.route 192.168.225.103->0.0.0.0, to wireless1
  route to 192.168.225.103
  flow got session.
  flow session id 1938
  update policy out counter info.
  skipping pre-frag
  going into tunnel 40000014.
  flow_encrypt: pipeline.
chip info: PIO. Tunnel id 00000014
(vn2)  doing ESP encryption and size =64
```

```
ipsec encrypt prepare engine done
ipsec encrypt set engine done
ipsec encrypt engine released
ipsec encrypt done
        put packet(28e5b10) into flush queue.
        remove packet(28e5b10) out from flush queue.
```

Everything looks rather similar until you get toward the end when you see that the traffic is sent to be encrypted. If you are having problems with traffic being sent out unencrypted, you might want to run this command on the traffic to help determine why it is not doing what you expect it to do. It might be a policy or route configuration issue, but this command should definitely help you figure that out. Note, that when traffic is coming back and being decrypted, you will see that happen before other actions are taken on the packet.

Debugging NAT

Troubleshooting Network Address Translation (NAT) requires a deep understanding of the network infrastructure. The Juniper firewall has two types of NAT: *policy-based* and *interface-based*.

- **Interface-based NAT** This is only applicable for Trust zones to Untrust zones and is configured by accessing Network | Interfaces in the WebUI. This method of translation is also referred to as *hide NAT*. Trust zone interfaces are in this mode by default. Interface-based NAT works by translating the source port of the packets to the egress interface IP address and changing the source port of the packets to keep track of the return traffic.
- **Policy-based NAT-src** This is configured in the Policies section of the WebUI and is the most configurable way to set up address translation. There is no limitation on which zones can use policy-based NAT-src. Dynamic IP pools (DIPs) can be employed using this method.

You can tell if the firewall is in interface-based NAT mode by typing **get config** and looking for the text "System in NAT/route mode." The interface in NAT mode will say "mode nat." Use the *get session* and the *debug log* commands to troubleshoot issues with all types of NAT. When using DIP pools, the command *get dip* will show the status of all IP addresses available in the pool.

The following lists common problems with source address translation:

- Invalid DIP address range
- Missing DIP pool

Common problems with destination address translation are shown next:

- VIP (Virtual IP) not mapped to the correct internal host
- Route in the wrong zone
- Missing inbound policy

The *debug flow basic* command can also be very helpful if you are trying to determine what translations are occurring on the packet as it traverses the firewall. This will help you pinpoint the decisions that are taking place and hopefully point you in the right direction.

Debugging VPNs

Troubleshooting virtual private networks can be easy if the right steps are followed. With Juniper firewalls, there are actually two different types of VPNs. Policy-based VPNs are based on rules in the Policies page of the firewall. Route-based VPNs are based on *tunnel* interfaces. They can also have policies on top of the tunnel interfaces blocking certain types of traffic through the tunnel.

When troubleshooting VPNs, the most important thing to remember is that both ends of the VPN have to share the same encryption settings. The following is a list of which VPN settings must be set the same on both ends of the tunnel. These settings are for both route-based and policy-based VPNs.

- Phase 1 key management protocol—for example, IKE
- Phase 1 encryption algorithm to encrypt the key—for example, DES, 3DES, AES, or CAST
- Phase 1 hash/authentication algorithm—for example, SHA1 or MD5
- Phase 1 authentication—for example, PRE-SHARED SECRET or CERTIFICATE
- Phase 1 mode—for example, MAIN or AGGRESSIVE
- Phase 2 encryption algorithm to encrypt the data—for example, DES, 3DES, AES, or CAST
- Phase 2 hash/authentication algorithm—for example, SHA1 or MD5
- Phase 2 Perfect Forward Secrecy—for example, YES-GROUP1, YES-GROUP2, YES-GROUP5, or NO
- Outgoing interface of the VPN tunnel
- Encryption domain

The Event log contains VPN events. When troubleshooting a VPN on a Juniper firewall, keep an eye on the Event log for PKI (public-key infrastructure) events. The following debug commands can be useful during troubleshooting Phase 1 issues:

- **get ike cookie** This will display all completed Phase 1 negotiations.
- **debug flow basic** This will enable debugging.
- **debug ike** This will enable detailed VPN debug logs with an empasis on phase 1 of the communication.
- **debug sa** This will turn on a debug with an emphasis on phase 2 of the VPN setup.
- **clear ike** This will force a VPN tunnel to renegotiate. It will clear Phase 1 and Phase 2 for the specified tunnel.

Troubleshooting commands useful for Phase 2 issues are shown next:

- **get sa active** This will display all completed Phase 2 negotiations.
- **unset ike policy-checking** This will tell the firewall to ignore the policy and allow all routed traffic through the VPN.

Policy-Based VPNs

The following are some common issues regarding policy-based VPNs:

- *Policies are in the wrong order.* Remember the rule base is parsed from top to bottom.
- *Missing a rule in the other direction*. VPN policies require a rule to allow inbound as well as outbound traffic.
- *Wrong VPN tunnel is selected*. Double-check the address book entries and the VPN tunnel selected.
- *Policy is in the wrong zone.* Make sure the traffic going into the VPN is allowed by a policy.

Route-Based VPNs

Troubleshooting route-based VPNs requires special attention to the routing table of the firewall. Since policies may be optional, we should also make sure a policy is not blocking our VPN traffic. These scenarios are often dependent on where your tunnel interface is in respect to the zone sending the traffic. Next are some common issues concerning route-based VPNs:

- The route is not in place to send traffic to the tunnel interface.
- A policy is in place blocking VPN traffic.

- The tunnel interface and originating traffic aren't in the same zone, but a policy isn't in place.
- Intrazone blocking is turned on when the tunnel interface is in the same zone as the source or destination of the traffic. If you want intrazone blocking on in this scenario, you must have a policy to permit the traffic.

Debugging NSRP

The NetScreen Redundancy Protocol provides redundancy and failover functionality for Juniper firewalls. NSRP allows for stateful failover, which means that the connection won't be broken when the failover occurs. On some smaller Juniper firewalls, such at the 5GT and the 25, this failover is not stateful (something known as H/A Lite). A dedicated link is required for the session table to be synchronized.

When troubleshooting NSRP, use the following commands:

- **get nsrp cluster** Displays the cluster information.
- **get nsrp monitor** Displays a list of monitored interfaces.
- **get nsrp vsd id 0** Displays Virtual Security Device number 0.
- **exec nsrp sync global-config check-sum** Tells you if the cluster members are synchronized.
- **exec nsrp sync global save** Synchronizes the configuration between cluster members; a reboot is necessary to complete the update.

Several factors can contribute to a slow failover. Auto-negotiation of the ports the firewall is plugged into should be manually set. The time it takes to negotiate the port speed and duplex might mean downtime. The heartbeat interval can be shortened. The default heartbeat interval is 1000ms, but it can be set as low as 200ms. Also keep in mind that a truly redundant configuration requires multiple switches, routers, and network connections.

Debugging Traffic Shaping

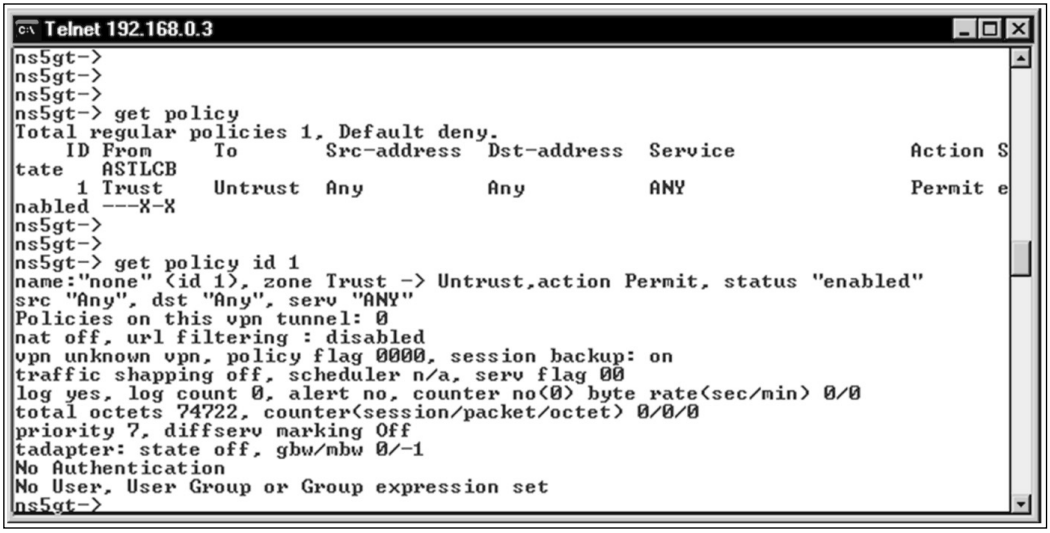
The Juniper firewall has the capability to limit the bandwidth packet use on a per-policy basis. If incorrectly configured, this can decrease the performance of the firewall significantly. When using traffic shaping, packets are placed into queues and released based on the shaping rules. Traffic shaping rules consist of guaranteed bandwidth, maximum bandwidth, and priority settings.

Each interface has a *traffic bandwidth* setting. To successfully use traffic shaping, this bandwidth setting should be configured for the most efficient bandwidth. Keep in mind that these settings are full duplex. If you set a rule to allow 1500 Kbps in one direction, the firewall will allow 1500 Kbps in the reverse direction.

- **Guaranteed Bandwidth** Reserves bandwidth in a policy rule.
- **Maximum Bandwidth** Limits the bandwidth used in a policy rule.
- **Priority** These settings let you give certain traffic higher priority over other traffic.

By default, all traffic is set to the lowest-priority queue. Take note of the guaranteed bandwidth settings on your rules since they cannot exceed the traffic bandwidth of the interface the traffic is flowing out of. You can get the traffic shaping rules from a policy using the *get policy id* command (see Figure 13.12).

Figure 13.12 *get policy id* in ScreenOS



Notes from the Underground…

Advanced Syslog

As you know, the Juniper firewall generates syslog messages. One of the best things you can do if you do not have a NetScreen Security Manager (NSM) server is to set up a syslog server to collect logs from your firewall.

Several excellent (and free) syslog filtering systems are available. If you search for syslog on Freshmeat.net (http://freshmeat.net), you will find over 100 different free software products that can make your firewall administration much easier. I recommend you install an Apache/PHP front end to the syslog server so

Continued

you can easily search through the logs, sort them, and generate reports for those management types.

A dedicated syslog server should be locked down tight. This server can accept syslog from any servers you point to it. Typically, this server should only answer on the syslog port, the HTTPS port for reports, and possibly SSH for administration.

NetScreen Logging

Juniper firewalls have the capability to log network traffic, and studying these logs can help your troubleshooting efforts immensely. Logs can be distributed via the following methods:

- **Console** Some log messages are sent to the console (serial, SSH, or Telnet).
- **Internal** The firewall can store a limited amount of logs for real-time troubleshooting.
- **E-mail** The Juniper firewall can be set up to e-mail syslog-generated log files.
- **SNMP** Simple Network Monitoring Protocol allows the NetScreen device to alert an SNMP management system.
- **Syslog** The Unix standard for log messages.
- **WebTrends** A third-party log analyzer.
- **NSM** NetScreen Security Manager is a management system for Juniper firewalls.

Traffic

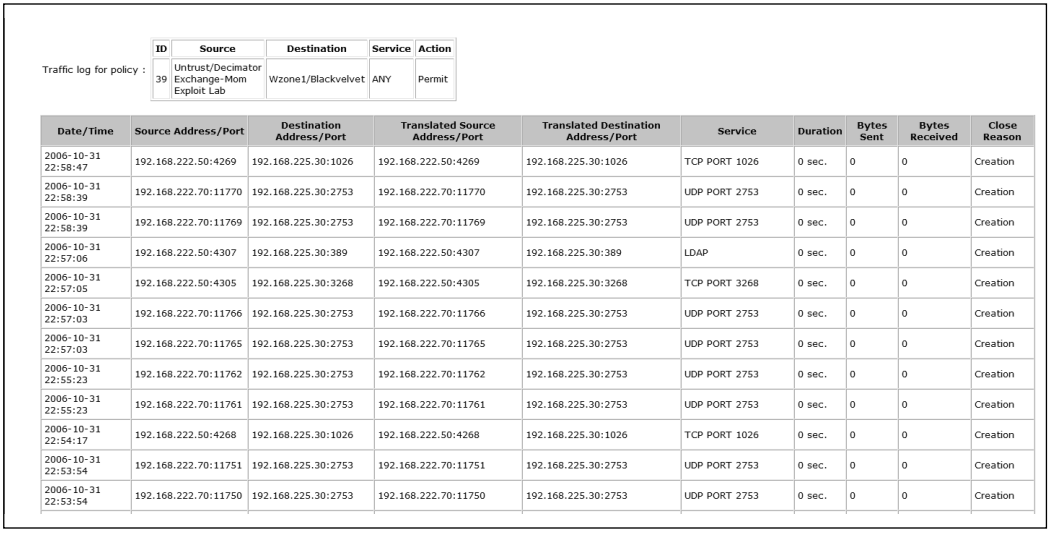
Logging can be enabled on a per-rule basis. You can access specific logs for a rule by clicking the log icon in the policy editor. This is helped by syslog's capability to output all traffic logs to a centralized syslog server.

The traffic log has the following fields (see Figure 13.13):

- Date/Time
- Source Address/Port
- Destination Address/Port
- Translated Source Address/Port
- Translated Destination Address/Port
- Service
- Duration
- Bytes Sent

- Bytes Received
- Reason for Session Close

Figure 13.13 Policy Traffic Log



Traffic log for policy :

ID	Source	Destination	Service	Action
39	Untrust/Decimator Exchange-Mom Exploit Lab	Wzone1/Blackvelvet	ANY	Permit

Date/Time	Source Address/Port	Destination Address/Port	Translated Source Address/Port	Translated Destination Address/Port	Service	Duration	Bytes Sent	Bytes Received	Close Reason
2006-10-31 22:58:47	192.168.222.50:4269	192.168.225.30:1026	192.168.222.50:4269	192.168.225.30:1026	TCP PORT 1026	0 sec.	0	0	Creation
2006-10-31 22:58:39	192.168.222.70:11770	192.168.225.30:2753	192.168.222.70:11770	192.168.225.30:2753	UDP PORT 2753	0 sec.	0	0	Creation
2006-10-31 22:58:39	192.168.222.70:11769	192.168.225.30:2753	192.168.222.70:11769	192.168.225.30:2753	UDP PORT 2753	0 sec.	0	0	Creation
2006-10-31 22:57:06	192.168.222.50:4307	192.168.225.30:389	192.168.222.50:4307	192.168.225.30:389	LDAP	0 sec.	0	0	Creation
2006-10-31 22:57:05	192.168.222.50:4305	192.168.225.30:3268	192.168.222.50:4305	192.168.225.30:3268	TCP PORT 3268	0 sec.	0	0	Creation
2006-10-31 22:57:03	192.168.222.70:11766	192.168.225.30:2753	192.168.222.70:11766	192.168.225.30:2753	UDP PORT 2753	0 sec.	0	0	Creation
2006-10-31 22:57:03	192.168.222.70:11765	192.168.225.30:2753	192.168.222.70:11765	192.168.225.30:2753	UDP PORT 2753	0 sec.	0	0	Creation
2006-10-31 22:55:23	192.168.222.70:11762	192.168.225.30:2753	192.168.222.70:11762	192.168.225.30:2753	UDP PORT 2753	0 sec.	0	0	Creation
2006-10-31 22:55:23	192.168.222.70:11761	192.168.225.30:2753	192.168.222.70:11761	192.168.225.30:2753	UDP PORT 2753	0 sec.	0	0	Creation
2006-10-31 22:54:17	192.168.222.50:4268	192.168.225.30:1026	192.168.222.50:4268	192.168.225.30:1026	TCP PORT 1026	0 sec.	0	0	Creation
2006-10-31 22:53:54	192.168.222.70:11751	192.168.225.30:2753	192.168.222.70:11751	192.168.225.30:2753	UDP PORT 2753	0 sec.	0	0	Creation
2006-10-31 22:53:54	192.168.222.70:11750	192.168.225.30:2753	192.168.222.70:11750	192.168.225.30:2753	UDP PORT 2753	0 sec.	0	0	Creation

Self

Logs sent to the Juniper firewall itself are referred to as *self logs*. These logs are very similar to the traffic logs.

The self log has the following fields:

- Date/Time
- Source Address/Port
- Destination Address/Port
- Duration
- Service

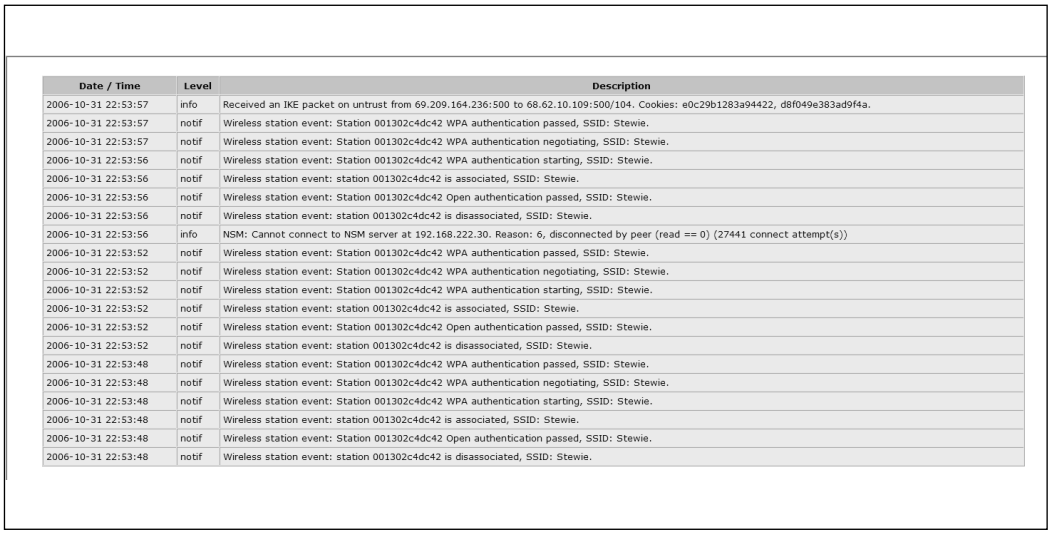
Event

Event logs are system logs generated when the Juniper firewall performs, or does not perform, a function. This log is useful to see when users log in to the WebUI or wish to determine the status of PKI certificates.

The event log has the following fields (see Figure 13.14):

- Date/Time
- Level (the severity level of the event, including Emergency, Alert, Critical, Error, Warning, Notification, Information, and Debugging)
- Description

Figure 13.14 Event Log Example in WebUI



Date / Time	Level	Description
2006-10-31 22:53:57	info	Received an IKE packet on untrust from 69.209.164.236:500 to 68.62.10.109:500/104. Cookies: e0c29b1283a94422, d8f049e383ad9f4a.
2006-10-31 22:53:57	notif	Wireless station event: Station 001302c4dc42 WPA authentication passed, SSID: Stewie.
2006-10-31 22:53:57	notif	Wireless station event: Station 001302c4dc42 WPA authentication negotiating, SSID: Stewie.
2006-10-31 22:53:56	notif	Wireless station event: Station 001302c4dc42 WPA authentication starting, SSID: Stewie.
2006-10-31 22:53:56	notif	Wireless station event: station 001302c4dc42 is associated, SSID: Stewie.
2006-10-31 22:53:56	notif	Wireless station event: Station 001302c4dc42 Open authentication passed, SSID: Stewie.
2006-10-31 22:53:56	notif	Wireless station event: station 001302c4dc42 is disassociated, SSID: Stewie.
2006-10-31 22:53:56	info	NSM: Cannot connect to NSM server at 192.168.222.30. Reason: 6, disconnected by peer (read == 0) (27441 connect attempt(s))
2006-10-31 22:53:52	notif	Wireless station event: Station 001302c4dc42 WPA authentication passed, SSID: Stewie.
2006-10-31 22:53:52	notif	Wireless station event: Station 001302c4dc42 WPA authentication negotiating, SSID: Stewie.
2006-10-31 22:53:52	notif	Wireless station event: Station 001302c4dc42 WPA authentication starting, SSID: Stewie.
2006-10-31 22:53:52	notif	Wireless station event: station 001302c4dc42 is associated, SSID: Stewie.
2006-10-31 22:53:52	notif	Wireless station event: Station 001302c4dc42 Open authentication passed, SSID: Stewie.
2006-10-31 22:53:52	notif	Wireless station event: station 001302c4dc42 is disassociated, SSID: Stewie.
2006-10-31 22:53:48	notif	Wireless station event: Station 001302c4dc42 WPA authentication passed, SSID: Stewie.
2006-10-31 22:53:48	notif	Wireless station event: Station 001302c4dc42 WPA authentication negotiating, SSID: Stewie.
2006-10-31 22:53:48	notif	Wireless station event: Station 001302c4dc42 WPA authentication starting, SSID: Stewie.
2006-10-31 22:53:48	notif	Wireless station event: station 001302c4dc42 is associated, SSID: Stewie.
2006-10-31 22:53:48	notif	Wireless station event: Station 001302c4dc42 Open authentication passed, SSID: Stewie.
2006-10-31 22:53:48	notif	Wireless station event: station 001302c4dc42 is disassociated, SSID: Stewie.

Summary

In this chapter, we covered various ways to troubleshoot network traffic passing through the Juniper firewall. We discussed the path a packet makes as it goes through the firewall, the various tools at our disposal, and tips for troubleshooting different functions available through ScreenOS.

Several troubleshooting tools are built in to ScreenOS. The *ping* command allows us to test connectivity. *traceroute* lets us find the path a packet takes through a network. Various *get* commands on the CLI show us internal tables in memory. ScreenOS also has a complete debugging system that allows us to view what happens to a packet as it goes through the firewall step by step. Snoop allows us to view the entire content of the packets that transverse the firewall.

The Juniper firewall is unique in that it can have virtual routers. Troubleshooting these routers can be easy as long as we note which settings apply to which virtual routers. Each interface on the firewall belongs to one zone, and each zone on the firewall belongs to a virtual router. Policies determine what happens when a packet needs to cross a zone. Intra-zone blocking, for instance, forces packets going to and from the same zone to pass through a policy lookup.

Troubleshooting VPNs requires configuration settings to agree on both ends of the VPN. Most VPN issues are due to a misconfiguration of the VPN settings on one end of the tunnel. The outgoing interface of the VPN tunnel must be set in order for the VPN to work properly.

NSRP is the NetScreen method of high availability. The heartbeat interval of the cluster can be tweaked to improve failover performance. Juniper firewalls support traffic prioritization. When troubleshooting traffic shaping, make sure the guaranteed bandwidth of the policy does not exceed the maximum bandwidth of the outgoing interface.

The firewall has a complete and detailed logging system. Traffic logs contain detailed logs of network traffic going through the firewall, while self logs contain detailed logs of traffic destined to the firewall itself. Event logs, on the other hand, contain system logs and alerts.

Solutions Fast Track

Troubleshooting Tools

- ☑ *ping* is a connectivity tool.
- ☑ *traceroute* verifies the path the packets are taking.
- ☑ *get session* allows you to view the session table in real time.
- ☑ *get policy* allows you to view the rule base of the firewall.

- ☑ *get route* lets you view the routing table.
- ☑ *get interface* permits you to view the interfaces of the firewall.
- ☑ *get arp* allows you to view the ARP table of the firewall.
- ☑ *get system* lets you view the status of the firewall and various settings.
- ☑ *debug* can be used to follow traffic through the firewall.
- ☑ *snoop* performs a detail packet capture.
- ☑ Firewall Session Analyzer makes *get session* information more manageable/readable.

Network Troubleshooting

- ☑ Use packet sniffers in your network to assist with troubleshooting.
- ☑ Make sure to determine if the packets are even reaching the firewall interface.

Debugging the Juniper Firewall

- ☑ Pay attention to the routing table.
- ☑ Survey both zone settings and policy settings.

Debugging VPNs

- ☑ Verify that the Phase 1 and Phase 2 settings are the same on both ends of the VPN tunnel.
- ☑ Make sure the outgoing interface of the VPN is set correctly.
- ☑ Double-check that VPN traffic is routed to the tunnel interface for route-based VPNs.
- ☑ Be certain VPN policies are in place with correct address book entries for policy-based VPNs.

Debugging NSRP

- ☑ The *exec nsrp sync global-config check-sum* command tells you if the cluster members are synchronized.
- ☑ Shorten the heartbeat interval if the firewalls are not failing over as quickly as you want them to.
- ☑ Verify that the speed and the duplex of the ports the firewall is plugged into are manually set.

Debugging Traffic Shaping

- ☑ Verify that each interface has its traffic bandwidth set.
- ☑ Make sure the guaranteed bandwidth settings do not exceed the traffic bandwidth for the interface.

NetScreen Logging

- ☑ Traffic logs contain logs of network traffic for rules with logging enabled.
- ☑ Self logs contain logs of network traffic terminated at the firewall.
- ☑ Event logs contain system events of the firewall.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: What command is used to view the full contents of packets?

A: The *snoop* command will dump the entire contents of the packets.

Q: How are the default routes usually configured when using two virtual routers?

A: One zone has a default gateway of the other zone. The other zone has a default gateway of the Internet router.

Q: What commands would you typically use to generate a debug log?

A: First, *debug flow basic* would enable the debug log. *undebug all* will stop logging. *get dbuf stream* will display the results. Of course, you may need to issue the *clear db* command as well, which will clear the buffer from previous results. You should do this before starting your debug.

Q: How would you limit what gets logged with *debug* or *snoop*?

A: The *set ffilter* command lets you set filters on what gets placed into the debug log.

Q: What commands would you use to verify that Phase 1 and Phase 2 of a VPN tunnel were completed?

A: The command *get ike cookie* lists Phase 1 completions, while *get sa active* lists Phase 2 completions.

Q: What is intra-zone blocking?

A: Intra-zone blocking blocks traffic within the zone unless there is a policy that allows the traffic to pass. By default, intra-zone blocking is disabled.

Chapter 14

Virtual Systems

Solutions in this chapter:

- **What Is a Virtual System?**
- **How Virtual Systems Work**
- **Configuring Virtual Systems**

- ☑ **Summary**
- ☑ **Solutions Fast Track**
- ☑ **Frequently Asked Questions**

Introduction

The Juniper firewall is a truly scalable device. On the high-end firewalls, you can divide the firewall into multiple virtual firewalls or virtual systems. A virtual system (vsys) is a logical firewall that is contained in a single physical firewall. Firewalls that support virtual systems enable you to create as many virtual systems as you are licensed for. Each virtual system can share components with other virtual systems or the root system.

Internet service providers (ISPs) or large organizations are the typical users of virtual systems. Both of these groups use virtual systems because of the need for many firewalls in a single location. For these users it would be impractical for them to have large numbers of firewalls. ISPs use the VSYS technology as a way to give customers access to their very own firewall while maintaining hundreds of virtual systems without the need for dedicated firewalls for each customer. Large organizations that require the use of many separate firewalls would benefit from the technology as well. The cost to use virtual systems is not an inexpensive proposition, but compared to maintaining many physical firewalls it can provide some cost benefits.

In this chapter, we will explore the virtual system technology and how to implement it. Together, we first look at the virtual system technology and what it provides. Next, we explore how virtual systems work. Looking deeply into how one physical device can differentiate traffic to dozens, if not hundreds, of different virtual systems. This is by far the most complex portion of virtual systems. There are two different methods to specify which traffic should be sent to which virtual system. We will look into each type of traffic classification and when to apply each one.

The last section of this chapter will be dedicated to creating virtual systems. This is the easiest part of the process of using a virtual system. Planning is always the biggest part of any battle. We will look at the creation, deletion, and administration of virtual systems.

What Is a Virtual System?

A virtual system is a unique security domain inside of a Juniper firewall. Each virtual system contains its own address book, user lists, custom service definitions, VPNs, and policies. Virtual systems also have their own virtual system administrators. These administrators are limited to accessing a specific virtual system. This limits the VSYS administrator to their own virtual system, thus keeping them out of other virtual systems and the root system.

Virtual System Components

Each virtual system has three components that can be used either in a shared state or exclusively. Each of these components should already be familiar to you. When shared, the component is made available to other systems, virtual or root. If the component is exclusive to a virtual system, it can be used only by that specific system. The following are the three main components of a virtual system:

- **Virtual Routers** When a virtual system is created, it automatically gains access to all shared virtual routers (VRs). A new virtual router is created simultaneously and named <vsys name>-vr. This VR is unable to be shared. Designing a routing architecture inside of your virtualized firewall is critical. You should do it before you even enter any commands into your firewall device.
- **Zones** Just as with virtual routers, when a VSYS zone is created, it has access to all shared zones. When a virtual system is created, three new zones are automatically created: Trust-<vsys name>, Untrust-Tun-<vsys name>, and Global-<vsys name>.
- **Network Interfaces (Shared)** Untrust Zone Interface Types: You can use several types of interfaces for the Untrust zone on a virtual system: a dedicated physical interface, subinterface (with virtual local area network [VLAN] tagging), and shared interface (physical, subinterface, redundant interface, aggregate interface) with the root system only.
- **Network Interfaces (Non-Shared)** Trust Zone Interface Types: dedicated physical interface, subinterface (with VLAN tagging), and shared physical interface with root system (using IP-based traffic classification).

Tools & Traps…

Sharing Nicely with Others

We have begun to discuss the idea of "shared" objects. A shared object is an object that can be used by multiple systems residing on the same physical firewall. On a virtual system, this consists of zones, interfaces, and virtual routers. Sharing the same objects across several virtual systems allows for the efficient distribution of resources.

All firewalls, no matter how large they are, have built-in limitations. I would strongly suggest that you plan the usage of the resources carefully. You never want to rebuild your network because you didn't plan ahead. The firewall products are very flexible and offer you many options. Just ensure yourself room for growth for your design.

How Virtual Systems Work

When using virtual systems, you essentially have one physical firewall device and many virtual firewalls inside that single firewall. Amazingly enough, the NS-5400 firewall can support up to 500 virtual systems. To support this, Juniper has derived some amazing ways to support this type of architecture. In this section of the chapter, we will look into just how Juniper enables this to happen.

Classifying Traffic

There are two types of mechanisms that Juniper uses to determine where traffic entering the physical firewall should go. It decides this based upon the type of traffic entering the device. The first type of traffic is traffic that is destined for the virtual system itself. Because there is no other place for the traffic to go except for the configured component of the virtual system, the traffic goes to that particular virtual system. This type of traffic includes traffic destined for a virtual private network (VPN), mapped IP (MIP), or virtual IP (VIP).

However, there is a second type of traffic that creates a great deal of difficulty when attempting to categorize it: through traffic. This is traffic destined for hosts beyond the firewall itself; thus, it passes through your firewall. There are two methods of handling traffic of this type. The first method is to use VLAN tagging. This method determines which traffic is destined for particular virtual systems. The second method available is called IP traffic classification. This requires you to manually configure which subnets or IP ranges are destined for a particular virtual system.

VLAN-Based (Interface) Classification

VLAN-based traffic classification employs the use of VLAN tagging to determine the destination of traffic. Each virtual system using a subinterface would require that interface to have its own specific VLAN dedicated to that interface. This requires not only configuring your firewall but configuring your network infrastructure as well. This requires that you divide your network into a configuration that encompasses VLANs. Juniper firewalls support the IEEE 802.1Q standard tags.

VLANs are bound to virtual systems by a subinterface. A VSYS must be associated with a VLAN when it shares its Untrust interface with the root system and has an interface bound to its own "Trust-<vsys name>" zone. Also, if the VSYS has a subinterface bound to the Untrust zone, that particular VSYS must be associated with another VLAN in the Untrust zone.

The same idea applies if you use a physical interface instead of a VLAN interface. If an interface is not imported into the VSYS, then the VSYS administrator doesn't have any control over it.

IP-Based Classification

There will be times when it may not be possible to reconfigure your network, so it encompasses a design with VLANs. If such is the case, another option can be used called IP-based classification. IP-based classification enables you to manually specify which traffic should be sent to which VSYS.

Typically, IP-based classification is used when you are using the VSYS as a transit to a remote network and you have no actual dedicated interfaces for the VSYS. In my experience, this is the lesser used of the two designs.

Virtual System Administration

Administering a virtual system is the same as administering a regular appliance. The only difference is that you can have only one read-write administrator and one read-only administrator per VSYS. A read-write administrator for a VSYS has the same privileges as a read-write administrator for an appliance. As the name goes, the read-only administrator can only view the configuration of the virtual system.

The root administrator over the entire firewall device is allowed to create and delete virtual systems. The root administrator is required to give resources to virtual systems. If you wanted to give a virtual system access to interfaces or virtual routers, the root administrator is required to do this. These tasks cannot be done by a VSYS admin. After the VSYS admin has the interfaces or VRs in the VSYS, it can do whatever it wants with them, but only after the root administrator gives the VSYS access to the resource.

Configuring Virtual Systems

As complicated as virtual systems sound, their configuration is actually quite easy. In this section, we'll discuss the creation of virtual systems. Creation of a virtual system is easy—the hardest part is in their planning. Due to the complexity of using multiple firewalls in the same unit, you should always have a complete plan when using virtual systems. The last thing you want is to have important questions arise at the zero hour. In fact, I firmly believe that success is 99% planning and 1% execution. Thus, everything should be so well mapped out beforehand that your project's execution feels seamless.

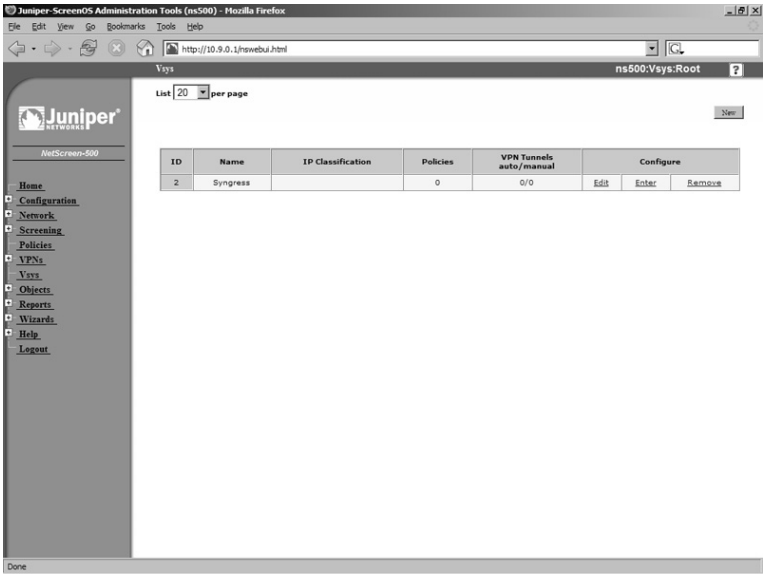
Creating a Virtual System

When initially creating a virtual system, you only have one decision to make. You must determine the name for your new virtual system. When you create a virtual system from the CLI, you will immediately enter the virtual system so you can continue configuring it. In the WebUI, you must click the button labeled Enter to access the VSYS.

When generating a VSYS, you must create an administrator for the newly created system. At login, this administrator will have access to that system only. In the WebUI, you create the virtual system and the administrator in one fell swoop. You can see the virtual

system main page in Figure 14.1. When creating virtual system administrators from the CLI, it requires a few additional commands, just as any CLI configuration takes.

Figure 14.1 The Virtual System Main Page (WebUI)



From the WebUI:

VSYS

1. In the upper right-hand corner, click the button labeled **New**. The VSYS | Edit page will appear.
2. To configure a VSYS, you must configure at least a name. Enter the name in the text box labeled **Vsys Name**.
3. To create a read-write admin for this VSYS, enter the VSYS admin's name in the text box labeled **Vsys Admin Name**, and the new password in the boxes labeled **Vsys Admin New Password** and **Confirm New Password**.
4. You have several more options on the VSYS creation page. For instance, you can also create a read-only VSYS admin.
5. To create a read-only admin for this VSYS, enter the VSYS admin's name in the text box labeled **Vsys Read-Only Admin Name** and the new password in the boxes labeled **Vsys Read-Only Admin New Password** and **Confirm New Password**.
6. The last option enables you to determine what type of virtual router to use. You have three options. The first is to create a new default virtual router. To do this,

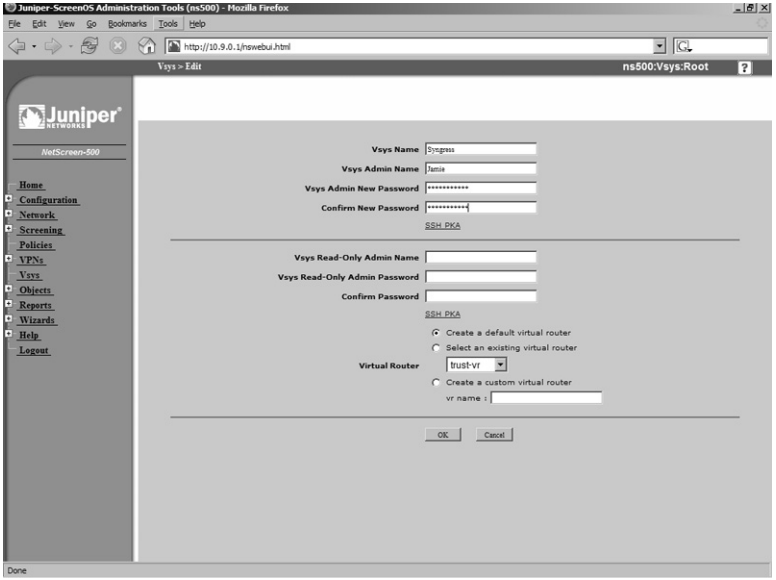
select the radio button labeled **Create a Default Virtual Router**. To select an existing virtual router, mark the radio button labeled S**elect an Existing Virtual Router** and choose a virtual router from the drop-down box under that selection. The last option is to create a custom virtual router. To do this, select the radio button labeled **Create a Custom Virtual Router** and enter the new VR name in the text box labeled **vr name**.

From the CLI:

```
Ns500-> set vsys Syngress
Ns500(Syngress)-> set admin name Jamie
Ns500(Syngress)-> set admin password MasterCheif
Ns500(Syngress)-> save
Ns500(Syngress)-> exit
Ns500->
```

Figure 14.2 displays the virtual system creation page.

Figure 14.2 The Virtual System Creation Page (WebUI)



Network Interfaces

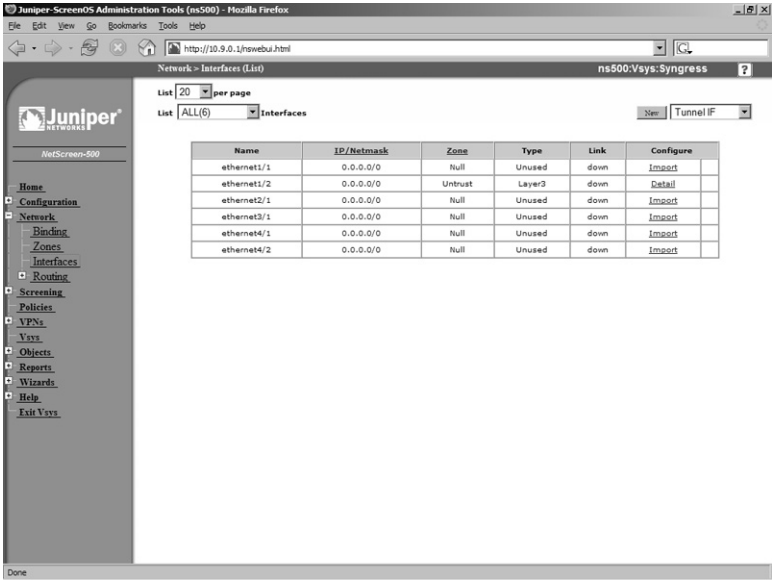
Because a virtual system is a firewall, its key components are its interfaces. Interfaces are required to pass traffic from one zone of your network to another. As we briefly discussed earlier, there are several types of interfaces that a virtual system can use:

- **Physical Interface** As the name states, a physical interface is an actual physical port that would connect directly to the network.
- **Subinterface** Subinterfaces are tied directly to a physical interface and require the use of VLAN tagging to differentiate which traffic is destined to which subinterface.
- **Shared Interface** Interfaces can only be shared with the root system. There are several types of shared interfaces you can configure (physical, subinterface, redundant interface, aggregate interface). It would be normal to use a shared interface, for example, when you have an Internet connection you want to configure to share with your virtual systems. When using shared interfaces, you must also configure traffic classification to determine which virtual system should process the traffic. Multiple virtual systems can share the same interface with root. But don't think of this as two virtual systems sharing an interface with each other. Just think of it as the root system sharing the interface with each of the virtual systems.

Physical Interfaces

When you decide to dedicate a physical interface to a virtual system, that virtual system gets exclusive use of that interface. To do this, you must import the physical interface to the virtual system. When you want to import an interface, it must be bound to the Null zone at the root level before it can be imported. When you've finished using a physical interface, you must export it to give the interface back to the root system. In Figure 14.3, you can see the screen used during the importation process.

Figure 14.3 Interface Importation (WebUI)



From the WebUI:
To import the physical interface to the virtual system, do the following:

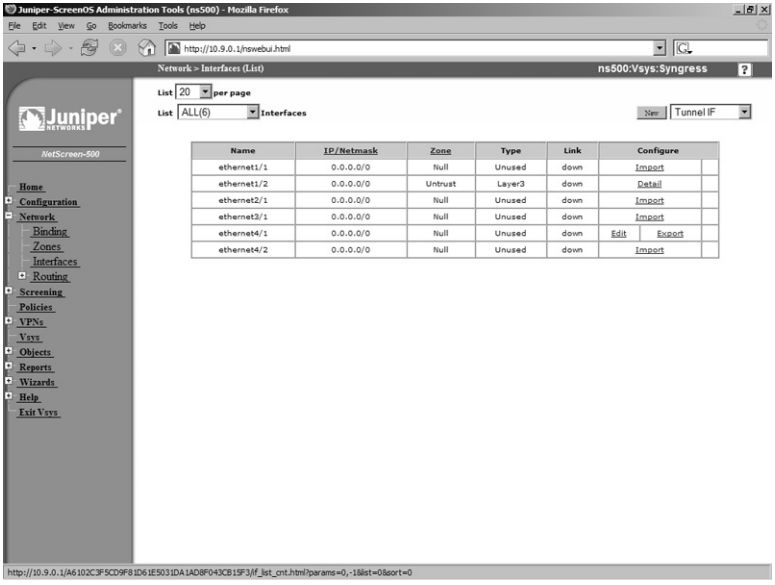
1. Log in as the root administrator or read-write administrator for the root system.
2. Identify the Vsys you want to import a physical interface to.
3. In the Vsys row, click the link labeled **Enter**. The Vsys you chose appears.
4. Select **Network | Interfaces**. You are presented with the list of interfaces for the device.
5. Identify the interface you want to use and import it into the virtual system.
6. Click the link labeled **Import**.
7. You will be prompted with a dialog that reads "You are about to import an interface. Are you sure you want to continue?" Click **OK**.
8. After the interface is imported, you can add a zone to the interface and then an IP address, just as you normally would.
9. To export an interface, enter the virtual system, as mentioned earlier, and go to the **Network | Interfaces** location in the WebUI.
10. The interface must have the IP address removed and the zone removed before it can be exported. Identify the interface you want to export, and then click the link labeled **Export**.
11. You will be prompted with a dialog that reads, "You are about to export an interface. Are you sure you want to continue?" Click **OK**.

From the CLI:

```
Ns500-> set vsys Syngress
Ns500(Syngress)-> set interface ethernet4/2 import
Ns500(Syngress)-> set interface ethernet4/2 zone Trust-Syngress
Ns500(Syngress)-> set interface ethernet4/2 ip 10.10.10.1/24
Ns500(Syngress)-> save
Ns500(Syngress)-> exit
Ns500-> set vsys Syngress
Ns500(Syngress)-> unset interface ethernet4/2 ip 10.10.10.1/24
Ns500(Syngress)-> unset interface ethernet4/2 zone Trust-Syngress
Ns500(Syngress)-> unset interface ethernet4/2 import
Ns500(Syngress)-> save
Ns500(Syngress)-> exit
Ns500->
```

Figure 14.4 displays the screen used during the exportation process.

Figure 14.4 Exporting an Interface (WebUI)



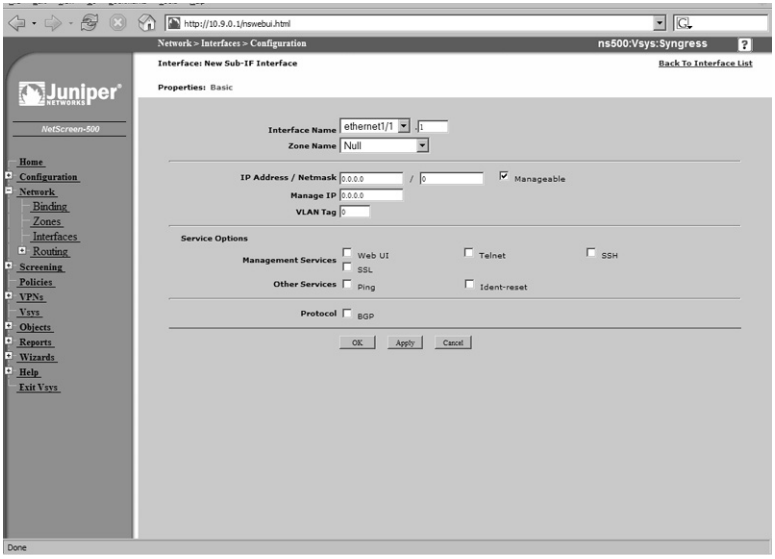
Subinterfaces

Subinterfaces are virtual interfaces that are bound to a specific physical interface. When using subinterfaces, you are also required to use VLANs. This means you may have to reconfigure your network to support VLANs. Figure 14.5 shows the WebUI Subinterface Configuration screen.

From the WebUI:

1. Log in as the root administrator or read-write administrator for the root system.
2. Identify the Vsys you want to create a subinterface for.
3. In the Vsys row, click the link labeled **Enter**.
4. Go to **Network | Interfaces**.
5. In the upper right-hand corner of the page, choose **Sub-If** from the drop-down menu.
6. Click the button labeled **New** in the upper right-hand corner.
7. Configuring a subinterface is similar to configuring a physical interface (as shown in Chapter 3). Enter your IP addressing, subnetting, and management configuration here.

Figure 14.5 Subinterface Configuration (WebUI)



8. Choose which physical interface you want to be bound to at the top of the page, and then set your VLAN tag in the text box labeled **VLAN Tag**.
9. After you have completed your configuration, click **OK**.

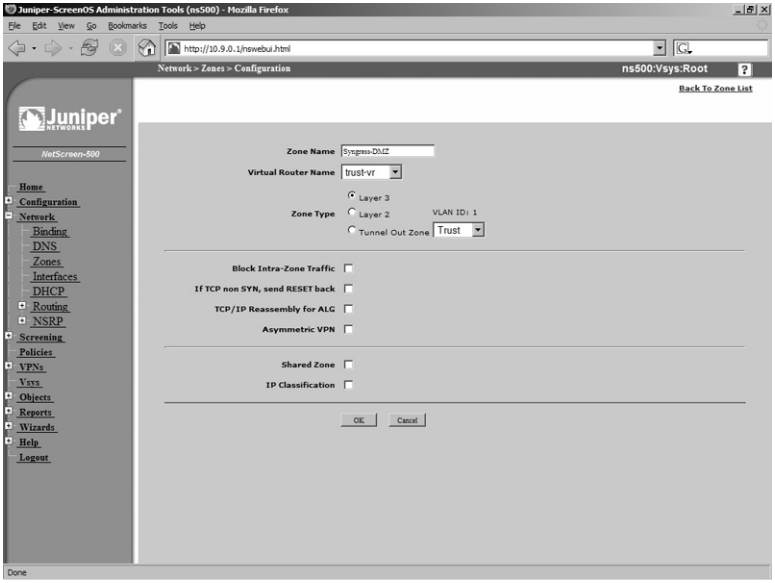
From the CLI:

```
Ns500-> set vsys Syngress
Ns500(Syngress)-> set interface ethernet4/2.1 zone Trust-Syngress
Ns500(Syngress)-> set interface ethernet4/2.1 ip 10.10.10.1/24 tag 4
Ns500(Syngress)-> save
Ns500(Syngress)-> exit
Ns500-> set vsys Syngress
Ns500(Syngress)-> unset interface ethernet4/2.1 ip 10.10.10.1/24
Ns500(Syngress)-> unset interface ethernet4/2.1 zone Trust-Syngress
Ns500(Syngress)-> save
Ns500(Syngress)-> exit
Ns500->
```

Shared Interface

Configuring a shared interface relies on the configuration of a shared zone. Because of the zone hierarchy, you must create a zone that's shared. After you apply that shared zone to an interface, it is then automatically shared to all virtual systems. In Figure 14.6, you can see the zone configuration screen with the option to share the zone.

Figure 14.6 Shared Zone Configuration (WebUI)



From the WebUI:
To configure a shared interface, do the following:

1. Log in as the root administrator or read-write administrator for the root system.
2. Go to **Network | Zones**.
3. Click the button labeled **New** in the upper-right hand corner.
4. In the text box labeled **Zone Name**, enter the zone name.
5. At the bottom of the page, check the box labeled **Shared Zone**.
6. Go to **Network | Interfaces**.
7. Identify the interface you want to share and click the link titled **Edit**.
8. At the top of the page in the drop-down box labeled **Zone**, choose the shared zone you created.
9. Click **OK** at the bottom of the page.

From the CLI:

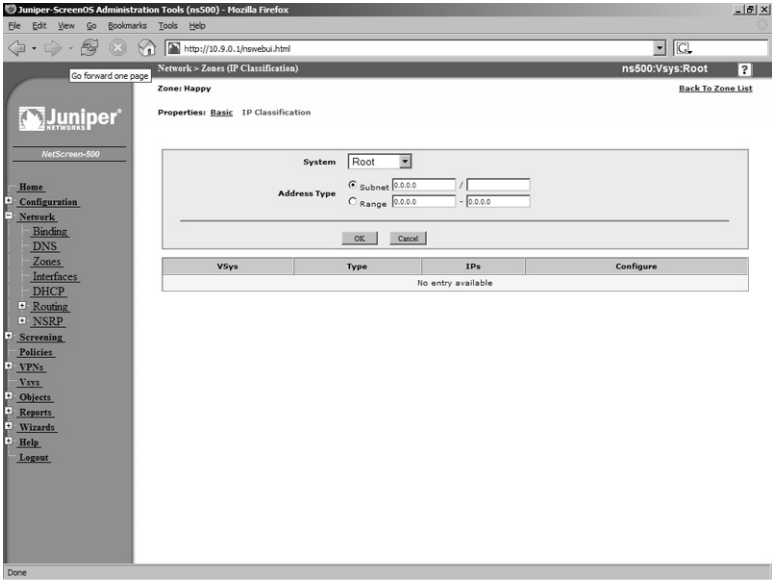
```
Ns500-> set zone name Syngress-DMZ
Ns500-> set zone Syngress-DMZ shared
Ns500-> set interface ethernet4/2 zone Syngress-DMZ
Ns500-> set interface ethernet4/2 ip 10.10.10.1/24
Ns500-> save
```

```
Ns500->
Ns500-> unset interface ethernet4/2 ip 10.10.10.1/24
Ns500-> unset interface ethernet4/2 zone Syngress-DMZ
Ns500-> save
Ns500-> exit
Ns500->
```

Traffic Classification

The second part to using a shared zone is traffic classification. By configuring traffic classification, you are explicitly defining which zone traffic is destined for. Traffic classification is configured on a per-zone basis. You would configure traffic classification on the shared zone. Figure 14.7 shows the WebUI configuration screen for traffic classification.

Figure 14.7 Traffic Classification (WebUI)



From the WebUI:
To configure traffic classification, do the following:

1. Log in as the root administrator or read-write administrator for the root system.
2. Go to **Network | Zones**.
3. Identify the zone you want to configure traffic classification for and click the link labeled **Edit**.
4. In the center of the top part of the page, click the link titled **IP Classification**.

5. In the drop-down box labeled **System**, choose which system you want to map the defined traffic to. You can choose either an IP subnet or an IP range.
6. If you want to define an IP subnet, select the **Subnet** radio button and enter the **IP subnet** in the first text box to the right of the text **Subnet** and then the netmask in the text box labeled "/" in the same row as the text **Subnet**.
7. If you want to define an IP address range, select the **Range** radio button and enter the first IP address in the first text box to the right of the text **Range**. In the text box next to the text "-", enter the last IP address in the specified range.
8. After you have defined your range or subnet, click **OK**.
9. To remove the range or subnet from the Network | Zones (IP Classification), identify the subnet/IP range you want to remove and click the link titled **Remove**.
10. You will be prompted with a dialog that reads, "You are about to remove an IP classification list. Are you sure you want to continue?" Click **OK**.

From the CLI:

```
Ns500-> set zone Syngress-DMZ ip-classification range 1.2.3.4-1.2.3.20 vsys
Syngress
Ns500-> set zone Syngress-DMZ ip-classification
Ns500-> save
Ns500->
Ns500-> unset zone Syngress-DMZ ip-classification range 1.2.3.4-1.2.3.20 vsys
Syngress
Ns500-> unset zone Syngress-DMZ ip-classification
Ns500->
```

Tools & Traps…

So Many Configurations, So Little Time

As you can see, configuring an actual VSYS is an easy process. It contains many of the same elements used when configuring a regular Juniper appliance. When deploying a Juniper system that is capable of using virtual systems, you will be working with a product that costs tens of thousands of dollars. In some cases, the virtualization license that gives a system the capability to use virtual systems can cost tens of thousands of dollars as well.

Continued

This type of deployment, whether for one or for five hundred virtual systems, requires intense planning and great scrutiny to ensure a successful implementation. It's not because the configuration is complex; it usually has to do with the scale of the deployment. Make sure you have documented what you want to configure on each virtual system. Diagrams are always a great benefit to any deployment and are of great help in long-term documentation.

Virtual System Profiles

In ScreenOS 5.4 and later, you can create Virtual System Profiles. These profiles enable you to limit the resources that are available to a specific VSYS. Each firewall, no matter how large it is, will always have resource limitations to it. Because of this, in some environments you may need to limit the resources available to the VSYS on that device. If you control the root firewall and all of the associated VSYS, then it is easy for you to control the resources available to each VSYS. You can create a total of eighteen VSYS profiles.

However, when there is delegated administration and each VSYS is controlled by a different administrator, this task can be more difficult. This is where the VSYS profiles come into play. The following list includes the various items that you can limit access to:

- Dynamic IP addresses (DIPs)
- MIP addresses
- User-defined services and groups
- Policies and multicast policies
- Sessions
- Zone address book entries and groups, which are per-zone, per-VSYS limits
- User-defined security zones
- CPU weights

Most of the items that you can limit, such as MIPs and VIPs, are a very straightforward concept. For these items you can specify a maximum and a reserve value. The maximum value is the most of a specific item you can create. The reserve value is the total amount of a specific item you are guaranteed to get access to. This ensures that you will have access to a specific resource for the use inside of a VSYS. The maximum value does not guarantee access to the specified number.

Configuring session limitations enables you to limit the number of sessions that a specific VSYS can use. This feature prevents a specific VSYS from overwhelming an entire device. When configuring session limitations, you can configure three different values. The first value is the maximum number of sessions. This value is the total number of sessions you can have for a specific VSYS. Secondly, you can also configure the total number of reserved

sessions. This number represents the total number of sessions a VSYS is guaranteed. It can not exceed the configured maximum number of sessions configured for a VSYS. The last configuration option is the alarm threshold. The threshold is configured in a percentage. When the threshold is met it triggers an alarm.

In the past you could run into a situation where a single virtual system could consume all the CPU resources of your firewall. This really defeats some of the value that you get out of using virtual systems for consolidation. In ScreenOS 5.4, Juniper added the feature to restrict the number of CPU cycles used per VSYS. This is done by configuring CPU weights. This method uses the following formula to determine the percentage of CPU utilization: (VSYS_Weight)/(Total_VSYSWeight) = CPU Percent. When configuring CPU utilization protection, you can use any values you want to determine this and available CPU percentage is determined by the formula above.

From the WebUI:

To configure traffic classification, do the following:

1. Log in as the root administrator or read-write administrator for the root system.
2. Go to VSYS | Profile.
3. Identify the profile you want to modify and click **Edit**
4. Enter the values for DIPs, MIPs, Policies, Sessions, and CPU weight here.
5. Click **OK** when completed.

From the CLI:

```
Ns500-> set vsys-profile name Syngress-Profile cpu-weight 30
Ns500-> set vsys-profile Syngress-Profile dips max 25 reserve 5
Ns500-> set vsys-profile Syngress-Profile mips max 25
Ns500-> set vsys-profile Syngress-Profile mpolicies max 5
Ns500-> set vsys-profile Syngress-Profile policies max 50
Ns500-> set vsys-profile Syngress-Profile sessions max 1200
Ns500-> save
```

Tools & Traps…

Troubleshooting Virtual Systems

When working with any product, you are bound to run into trouble now and again. When you troubleshoot a VSYS problem, this can well be challenging to do. Any sort of debugging can be done by only the root user in the root VSYS. This encompasses any debug commands or "get dbuf" commands. The root user can, of course, also enter any VSYS to do connectivity testing with ping, traceroute, or mtrace.

The limitation comes when a VSYS administrator needs to troubleshoot an issue. They are limited to only ping, traceroute, or mtrace inside of their specific VSYS. Although connectivity testing is helpful, it does not give the VSYS administrator the capability to debug and get to the root of an issue.

Summary

In this chapter, we looked at virtual systems. As you have learned, virtual systems are a powerful tool you can use to divide up your Juniper firewall system into several firewalls or virtual systems. This enables you to maximize the return on investment (ROI) of a single large firewall, enabling it to be divided into multiple independent firewalls. This provides several benefits.

It allows for separate management domains. You can divide your firewall into several smaller logical devices and thereby separate management resources from one another. You can use it the same way you would if you had two or more separate firewalls. This is often done to logically separate two distinct parts of the network. In the case where you would use two separate physical firewalls, you could use just one Juniper system that's capable of running virtual systems.

Virtual systems are just the next logical step in the evolution of firewall design and show off Juniper's excellent product design by demonstrating that the Juniper firewall is such a scalable device.

Solutions Fast Track

What Is a Virtual System?

- A virtual system is a unique security domain inside a Juniper firewall.
- Virtual systems can use components shared by the root system.
- You can define a virtual system so it will use its own virtual router.

How Virtual Systems Work

- Juniper firewalls have two ways of classify traffic, thereby deciding which virtual system to send it to.
- When using a subinterface, you must configure VLAN tagging to differentiate traffic.
- You can only have one read-write administrator and one read-only administrator per virtual system.

Configuring Virtual Systems

- Creating a virtual system is an easy one-step process.
- Physical interfaces that are dedicated to a virtual system must be imported into the virtual system.
- If you are going to use shared interfaces, you must configure IP classification to decide which virtual system will receive which traffic.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: Virtual systems seem like a great idea, but are they practical for my environment?

A: Organizations very rarely use virtual systems. They are only practical to use when you require many separate firewalls. Only large organizations and ISPs have the type of environment that requires virtual systems. Even though the application of virtual systems may be beneficial to you, the cost may be prohibitive.

Q: Why would you want to share a resource instead of using a dedicated resource?

A: There are many valid reasons why you would want to share resources instead of using dedicated resources. The first good reason would be to conserve resources. You may require many resources and dedicating them may not be feasible. A great second reason would be practicality. It may be easier to have one physical interface connected to the Internet and share it amongst five virtual systems than to dedicate five interfaces to the same Internet connection. The great part about this device's design is that you could do either depending on your requirements.

Q: Configuring and managing many virtual systems seems complex. Is there a better way to manage all of this?

A: Juniper provides a platform called the Juniper Security Manager for all your central management needs. The NSM is discussed in Chapter 16 and is a great investment in a heavy Juniper infrastructure.

Q: How do I get my network to support VLANs?

A: A network switch that uses VLANs is required to support VLAN architecture. Many switches today support the use of VLANs. Typically, a managed switch can support VLANs. Look to the documentation of your switch manufacturer to see what your switches can do.

Q: Can you give VSYS administrators the same name on different VSYSs?

A: Administrator names are unique. After you specify an administrative user with a particular name, that name cannot be used a second time.

Index

802.1x
  RADIUS server support of, 292
  switch infrastructure, 56
  user type, 252, 257
802.1x authentication
  components of 802.1x, 278–279
  configuration of, 279–284
  description of, 277

A

AAA (authentication, authorization, auditing), 18
ABR (Area Border Router), 336–337, 338
access control, 160–162
access lists
  configuring, 307–308
  extended, 383–386
  overview of, 306
  properties, 306–307
access policy, Deep Inspection, 511–512
accessibility, security and, 19
account type, 255
Acct-session-ID length, 256
ACK flag, 500
ACK packet, 14
action groups, 387–389
actions, policies and, 159
Active/Active cluster
  setting up, 657–664
  VSD default routes and, 688
Active/Active HA, 589
Active/Passive HA, 589
address book entries
  address groups, 171–172
  binding to zones, 189
  creating, 168–170
  IP address ranges and, 188–189
  modifying/deleting, 171
  policy creation and, 168, 189
address groups, 171–172
Address Resolution Protocol (ARP)
  ARP/Traceroute mode, 461–462
  IP to MAC address resolution, 16
  lookup, policy checking and, 164
  packets, 671
  transparent mode and, 461, 478
address translation. *See* Network Address Translation (NAT)
Admin user type, 253
administrative users, 93–95
administrator, names, 743
Administrator accounts
  external Admin authentication, 237–239
  external authentication server and, 255
  LDAP server's support of, 263
  local Admin authentication, 235–237
  RADIUS server's support of, 257
  SecureID server's support of, 261
  types of, 235
ADSL (asymmetric digital subscriber line), 593–597
advertisements
  BGP routes, 365–366
  link state, 340
  RIP, 330–332
Agentless Infranet Authentication, 276–277
aggression, route
  BGP and, 368–369
  configuring, 369–370
Agobot, 490
AH (authentication header), 553
alarm in threshold, 496
ALG (Application Level Gateway), 403–404
ALGs (Application Layer Gateways), 540–542
American Heritage Dictionary, 18–19
anti-spam, 160
Antivirus
  global parameters, 534–535
  network antivirus concepts, 533
  planning, 533–534
  profile settings, 537–538
  rules, 538–539
  Scan Manager settings, 536–537
  verification of protection, 540
  virus traits, 532
Antivirus profile, 537–538, 539
antivirus scanning, 160
antivirus software, 24, 532
"any", 174
appliances, midrange, 641–642
application access, 54
application layer
  Deep Inspection at, 501–502
  IP connectivity, 7
  of OSI model, 4
Application Layer Gateways (ALGs), 540–542
Application Level Gateway (ALG), 403–404
application protocols
  Antivirus profile, 537–538
  Deep Inspection engines for, 503–504
application proxy firewall, 28–29
Application setting, 521, 522
application threats, 25–26
application-level defense, 480
application-level inspection
  of Juniper firewall, 59–60
  need for, steps of, 60–61
applications, traffic-shaping and, 192–194
Application-Specific Integrated Circuit (ASIC), 52, 62, 88, 548
Area Border Router (ABR), 336–337, 338
areas
  OSPF, properties of, 343–345
  OSPF, types of, 337–338
  overview of, 336–337
  routers within, 338–339
  virtual links, 337
ARP. *See* Address Resolution Protocol
AS. *See* autonomous system
as paths, 364–365, 366
ASBR Summary LSA (4), 340
ASIC. *See* Application-Specific Integrated Circuit
asymmetric digital subscriber line (ADSL), 593–597
attack, anatomy of
  Black Hat hackers, 484–487
  hack, phases of, 484
  malware, 486–490
  script kiddies, 484–485
  social engineering, 486
attack detection and defense
  Application Layer Gateways, 540–542
  best practices, 542–543
  content filtering, 524–540
  SCREEN settings for security, 490–491
  SCREEN settings, TCP/IP behavior anomaly detection, 491–497
  SCREEN settings, TCP/IP protocol anomaly detection, 498–501
attack detection and defense, Deep Inspection, 501–523
  attack objects, 510
  concepts of, 503–505
  contexts, regular expressions, 514–519
  database for, 507–510
  description of, 501–503
  planning, 505, 507
  policy with CLI, 512–514
  policy with WebUI, 511–512
  search algorithm, 519
  signature writing with IDP, 523
  signatures, creation of, 519–523
  support of, 506
attack groups
  in Deep Inspection, 510
  Deep Inspection, policy with CLI, 512–514
  Deep Inspection, policy with WebUI, 511–512
  signatures in Deep Inspection, 522–523
attack objects
  in Deep Inspection, 510
  signatures, creation of, 519–523
attack threshold, 496, 497
attacks
  anatomy of, 484–490
  Brain virus, 1986, 481
  bug databases, 483
  CVE dictionary, 483
  IDP multi-method detection of, 55–56
  Juniper security research team, 483–484
  Morris worm, 1988, 481–482
  Panix SYN flood, 1996, 482
  types of, 480–481
  unified threat management, 482
  vulnerability databases, 482–483
attributes, BGP, 355–356
audit capability, 20
Auth groups
  configuration of, 240–241
  definition of, 239
Auth user type
  configuration of users, groups, 240–241
  LDAP server's support of, 263
  Local authentication server's support of, 253
  properties of, 239–240
  RADIUS server's support of, 257
  SecureID server's support of, 261
Auth users
  configuration of, 240–241
  definition of, 239
authentication
  dangers of, 264–265
  definition of, 20
  security options for, 25
  . *See also* user authentication
authentication, authorization, auditing (AAA), 18
authentication header (AH), 553
authentication servers
  802.1x, configuration of, 280–281
  external, 254–269
  local, 252–253
  optional parameters, 292
authentication users, 239–252

802.1x user type, 252
Auth user type, 239–241
IKE user type, 241–244
L2TP user type, 249–252
XAuth user type, 244–249
authorization/access control, 20
automated attacks, 484
autonomous system (AS)
BGP and, 355
boundary router, 338
description of, 336
availability, 20, 75
. *See also* high availability

B

Backbone Area, OSPF, 337
backbone router, OSPF, 338
backdoor detection, 55
back-to-back VPNs, 579
backup designated router, OSPF, 339
backup firewalls, 651
backup servers, 255
bandwidth
interface, configuring, 209–210
interface, properties of, 199
interface-based traffic shaping and, 197–199
virtual interface, 199–200
bandwidth-based traffic shaping, 195–197
banner messages, firewall, 284–287
basic-single firewall, 36
bastion hosts
in DMZ configuration, 33–34
in DMZ design, 39
securing/hardening, 41
traffic flow, 36–37
behavior anomaly detection, TCP/IP, 491–497
DoS flood protection, 493
ICMP network scan, 494
ICMP rate limiting, 494
IP session limiting, 493
need for, 491–492
reconnaissance detection, 492–493
TCP SYN host scan, 494–495
TCP SYN rate limiting, 495–497
UDP data rate limiting, 497
best practices, 542–543
BGP. *See* Border Gateway Protocol
Black Hat hackers, 484–487
Border Gateway Protocol (BGP)
attributes, 355–356
autonomous systems, 355
community, 367–368
confederation, 371
configuration, 358, 372–373
configuring route to advertise via, 365–366
flapping information, 373–374
iBGP, 357
informational commands, 372
instance, configuring in VR, 359–361
messages, 356
neighbor properties, 361–362
neighbors, configuring, 362–364
neighbors, viewing, 373
overview of, 354, 395
AS paths, 364–367
peers, 355
route aggression, 368–369
route aggression, configuring, 369–370
route flapping, 358
route reflectors, 370–371
routing table, 374
state, summarizing, 372
VR properties, 358–359
Bot. *See* zombie
Brain virus, 1986, 481
bridge, 16
broadcast
definition of, 16
methods, 461–462
networks, 339
bug databases, 483
bursty traffic, 195

C

cabling
connecting HA links via switches, 618–619
crossover Ethernet cable, 16
directly connected HA links, 617–618
for full-mesh configuration, 616–617
for NSRP clusters, 614
cache cleaner, 54
Cain & Abel tool, 264
Calling Station ID, 256
central processing unit (CPU), 548
CERT Vulnerability Notes Database, 482–483
certificate revocation lists (CRLs), 561
certificates
digital, 560–561
IKE user authentication with, 242
character class, 517
checking, policy, 164–165
CheckPoint FireWall, 541, 584
CIA (confidentiality, integrity, and availability), 18, 20
CIDR (classless interdomain routing), 354
Cisco
Juniper firewall products and, 51, 584
routers, 27–28
classless interdomain routing (CIDR), 354
CLI. *See* command-line interface
client-side security, 53
client-to-server (CTS) flows, 505
closed systems, security of, 19
clusters
Active/Active, 657–664
for HA through NSM, adding members to, 677–679
for HA through NSM, configuring, 676–677
ID, setting, 619–620
ID/name, setting, 620
NSRP, adding NetScreen to, 619
NSRP, enabling RTO mirroring in, 655
Code Red worm, 22, 490
Cohen, Fred, 481
collision domain, 16
command-line interface (CLI)
for cluster name/heartbeat settings, 646
Deep Inspection policy with, 512–514
for Juniper firewall, 99–103
Juniper firewall management with, 53
policy administration from, 58–59
policy creation via, 183–186
policy options, 186
router shortcuts, 397
commands
RIP informational, 332–335
AT via *exec modem* command, 600
common name dictionary, 483
common name identifier, 262
Common Vulnerabilities and Exposures (CVE) naming standard, 483
communications
data link layer, 15–17
TCP, 13–14
UDP, 14
communities, BGP, 367–368
compound signatures, 56
compression, 194
computer security, 18
confederations
configuring, 371
iBGP, 357
confidentiality, 20
confidentiality, integrity, and availability (CIA), 18, 20
configuration
Juniper firewall file, 95–99
Juniper firewall, first-time, 121–122
consolidation, 484, 487
content filtering
antivirus, 532–540
Web filtering, 524–532
contexts
Deep Inspection, 514–515
signatures, creation of, 521–523
control messages, 612
Control Mode, 279
costs, deployment, 738–739
counting
configuring, 218–220
description of, 215
performance and, 232
policies and, 160
traffic-shaping and, 216–218
CPU (central processing unit), 548
CRLs (certificate revocation lists), 561
crossover Ethernet cable, 16
CTS (client-to-server) flows, 505
CVE (Common Vulnerabilities and Exposures) naming standard, 483

D

data files, virus and, 532
data link layer
communications, 15–17
in IP connectivity, 7
of OSI model, 5–6
data packet. *See* packet
data transmission
data link layer for, 5–6
DMZ design for, 42
database
bug, 483
Deep Inspection, 507–510
RIP, 334–335
vulnerability, 482–483
Data-Link layer, 277
DDNS (dynamic DNS), 562
debug
for firewall troubleshooting, 703–704
Juniper firewall, 706–712, 721
tracing, 709–710
debug flow basic command, 706, 723
Deep Inspection (DI), 501–523
attack objects, 510
concepts of, 503–505
contexts, regular expressions, 514–519
coverage of, 548
database for, 507–510
description of, 501–503, 708
of NetScreen firewalls, 52
planning, 505, 507
policy with CLI, 512–514
policy with WebUI, 511–512
search algorithm, 519
signature writing with IDP, 523
signatures, creation of, 519–523
as supplement for security, 88
support of, 506
deep packet inspection, 60, 61
default policy, 164
defense-in-depth, 542
demilitarized zone (DMZ)
configurations, 31–32, 33–35
design fundamentals, 41–42
designs, 38–40
need for, 47
predesign path, 32–33
traffic flow, 35–38
traffic flow, protocols, and, 43–44
Denial of Service (DoS)
floods, protection against, 491–492, 493
IDP attack detection, 56
Panix SYN flood, 1996, 482
denies, implicit, 543
deployment options, 466, 477
designated router, OSPF, 338–339
destination address, 159
destination NAT, 428–445
firewalls, hackers bypassing, 429
function of, 428
methods of, 405
policy-based destination NAT, 433–443
with source NAT, 444–445
VIP, 429–433
destination PAT
description of, 401–402
scenario, 441–443
destination threshold, 496
destination translation
many-to-many mapping, 440
many-to-one mapping, 437

one-to-one mapping, 435
policy-based destination NAT property, 434
source/destination NAT, 444
destination-based forwarding, 304
destination-based routing tables, 297
destination-based static routes
configuring on firewall, 315–317
overview of, 314–315
Deterministic Finite Automaton (DFA) syntax, 515, 519
device
NSRP device level monitoring, 626
transparent mode configuration, 462–466, 477
device architecture, Juniper firewall, 61–62
DFA (Deterministic Finite Automaton) syntax, 515–518, 519
DI. *See* Deep Inspection
dial-up
advanced backup configuration, 599–600
for redundancy, 597
simple backup configuration, 597–599
dial-up VPNs
NetScreen Remote, 570–575
overview of, 569–570
Differentiated Services (DiffServ), 200–202
Diffie, Whitfield, 558
Diffie-Hellman, 558
DIP
configuration on policy, 422–423
DIP shift, 426–428
function of, 449
overview of, 420
properties of, 421
Sticky DIP, 423–425
DIP pool
configuration on policy, 422–423
definition of, 420
properties for configuration of, 421
DIP shift, 426–428
direction, 158
directly connected routes, 296
distinguished name (dn), 262–263
DMZ. *See* demilitarized zone
DNS ALG, Symantec, 541–542
DNS query, 14
domain name, 255
domain of interpretation (DOI), 553
DoS. *See* Denial of Service
dot-star, 516
Drop Unknown MAC option, 497
DSCP class mapping, 214
DSCP marking, 214
dual ADSL modems, 593–595
dual firewall with DMZ
design of, 38, 39
pros/cons of, 40
dual HA links, 612–613
dual providers, 646–651
dual untrust interfaces
ADSL modem/ADSL router, 595–597
dual ADSL modems, 593–595
for redundancy, 592–593
dual-firewall DMZ configuration, 34, 35
dynamic DNS (DDNS), 562
dynamic port allocation, 403–404
dynamic routes, 296
dynamic routing protocol, 396

E

EAP (Extensible Authentication Protocol), 278–279
echo reply packet, 10
echo request packet, 10
ECMP (Equal Cost Multiple Path), 299–300, 397
egress filtering, 543, 549
egress guaranteed bandwidth, 200
egress interface
interface-based source NAT, 407–409
policy-based source NAT, 417–418
source/destination NAT, 444
egress maximum bandwidth, 199
egress policing, 197
egress traffic, 230–231
EICAR (European Institute for Computer Antivirus Research), 540
EIGRP, 396–397
e-mail
Deep Inspection and, 502–503
firewall development and, 26
virus, 480
encapsulating security payload (ESP), 553
encryption
algorithms, 584
of network access, 25
enhanced pluggable interface module (EPIM) slots, 78
enterprise class, Juniper Networks, 76–81
enterprise management, 83–84
EPIM (enhanced pluggable interface module) slots, 78
Equal Cost Multiple Path (ECMP), 299–300, 397
errors, VPN configuration, 585–586
ESP (encapsulating security payload), 553
Ethereal, 425–426
Ethernet
data link layer communication, 15–17
interface, 281–282, 591
European Institute for Computer Antivirus Research (EICAR), 540
evasion technique, 537–538
event logs, 718
exchanges
Phase 1, 557–558
Phase 2, 559
exec modem command, 600
exec nsrp sync global-config command
for out-of-sync configurations, 687
RTO mirroring for state synchronization, 655
for synchronization, 621–622
executable file, virus attached to, 532
exploit, 484
export
interface, 732–734
of routes, 311–313
extended access lists, 383–386
Extensible Authentication Protocol (EAP), 278–279
external authentication, for Admin accounts, 237–239
external authentication servers, 254–269
Admin users and, 291–292
advantages of, 254
Infranet Authentication, 265–269
LDAP server, 262–264
properties for, 254–255
RADIUS server, 256–259
SecurID server, 260–262
External Groups, 291
External LSA (5), 340
external threats, 21

F

fail over
ARP packets, adjusting number after, 671
for external authentication server, 255
forcing, 687–688
IP tracking to determine, 601–602
NSRP, determining when to, 685
overview of, 670–671, 686
virtual systems, 671–673
file transfer application protocols, 533
File Transfer Protocol (FTP)
Application Layer Gateways and, 540–541
banner message, 284–287
not allowed on Serial interface, 601
policy creation and, 173
file-based resources, 54
filtering
antivirus filtering, 532–540
egress filtering, 543, 549
packet filters, 27–28
Web filtering, 524–532
filters, 165, 703–704
FIN flag, 500
Firewall Banner messages, 284–287
firewall limits, 450–455
firewall policies
access control theory, 160–162
overview of, 158–160, 187–188
policy-based user authentication, 269–277
preparation for making, 166–167
types of, 162–165
firewall policy components
address book entries, 168–172
overview of, 188
services, 172–177
zones, 167–168
firewall policy creation
overview of, 176–177, 188
via CLI, 183–186
via WebUI, 177–182
firewall rule, 431
firewall sandwich, 38, 39
firewall security technologies, 24
Firewall Session Analyzer (FSA), 704–705
firewalls
backup, forcing links down on, 651
bastion hosts, 41
as core security product, 17
DMZ concepts, 31–35
DMZ design, 41–42
DMZs, networks with/without, 38–40
external authentication integration to, 254
function of, 26
hackers bypassing, 429
ideologies, 31
networking and, 3
routers, connecting directly to, 613–615
routers, connecting to via switches, 615–616
spoofing source address of packets, 429
stateful, ECMP and, 300
traffic flow concepts, 35–38
traffic flow, protocol, 43–44
types of, 26–31
virtualizing, 608–610
. *See also* Juniper firewall; Juniper NetScreen firewall; specific firewalls
5-tuple, 165
flag, TCP, 500–501
flapping, route, 358
flapping information, 373–374
flash memory, of Juniper firewall, 63
flood mode, 461
floods
DoS flood protection, 493
ICMP, 494
IP fragment floods, 499
Panix SYN flood, 1996, 482
TCP SYN rate limiting, 495–497
UDP flood, 497
flow filters, 703–704
forced timeout, 255
fragmentation, IP, 498–499
frame, 7
Friedl, Jeffrey E. F., 519
FSA (Firewall Session Analyzer), 704–705
FTP. *See* File Transfer Protocol
full-mesh Active/Active, 589, 664–670
full-mesh configuration, 616–617
function zone, 57

G

gateway mode, 55
gateway redundancy, 578–579
gateways
for policy-based VPNs, 566
tracking default, 602–603
get arp command, 698–700
get ike cookie command, 723
get interface command, 698
get nsrp command, 638–641, 685
get policy command, 697
get policy global command, 707
get route command, 697–698
get session command, 696–697
get sys-cfg command, 450–455
get system command, 700–702
global policy, 58, 163
Global Pro product, 84
global zone, 409, 431
granularity, 216, 294
Grey Hat, 486, 548

Group Expressions
configuration of, 287–288
External Groups in, 291
function of, 284
properties of, 287
groups
address, 171–172
attack groups, 510
service, 175–176
guaranteed bandwidth, 200, 202–203

H

HA. *See* high availability
HA Lite, 590, 591
hack, phases of, 484
hackers
Black Hat, 485–487
Grey Hat, 548
script kiddies, 484–485
social engineering by, 486
threats from, 22
handshake, 495, 496
hardware, in NetScreen firewall architecture, 62
hashing algorithms, 584
heartbeat frequency, 646
heartbeats, 578, 624–625
Hellman, Martin, 558
high availability (HA)
failing over, 670–673
full NSRP, taking advantage of, 654–670
get nsrp, reading output from, 638–641
links, crossing, 687
need for, 588–589, 683
NetScreen SOHO appliances for, 591–608
with NetScreen-200 series, 75
with NetScreen-500, 77
no-brain problem, avoiding, 674–676
NSM, configuring through, 676–682
NSRP cluster, building, 613–624
NSRP failovers, 624–637
NSRP overview, 608–613
NSRP-lite on midrange appliances, 641–651
options, 589–591
overview of, 588
redundant interfaces, creating, 652–654
split-brain problem, avoiding, 673–674
high end line, Juniper Networks, 72–76
home Internet users, 488
Honeypot Networks, 489
hops, 322
host checker, 53
host IP address
in MIP configuration, 412, 413, 415
MIP property, 410
host routes, 296
host virtual router name, 410
hub, 16, 579–580
Hypertext Transfer Protocol (HTTP)
as application layer protocol, 4
banner message, 284–287
headers, 520
requests, Web filtering and, 524–525
signatures in Deep Inspection, 519–521
traffic-shaping and, 203, 206

I

IANA (Internet Assigned Numbers Authority), 12
iBGP, 357
IBM PC, Brain virus, 481
IC (Infranet Controller), 265–266
ICMP. *See* Internet Control Messaging Protocol
ICSA certification, 52, 87
IDP. *See* Intrusion Detection and Prevention
IDs, cluster, 619–620
IE. *See* Infranet Enforcer
ifconfig command, 11, 12
IKE. *See* Internet key exchange
IKE user type
configuration of users, groups, 241–244
Local authentication server's support of, 253
properties of, 241
with XAuth for single user, 248–249
import, of routes, 311–313
information security
areas of concern, 19–20
concepts of, 18–19
. *See also* security
informational commands
BGP, 372–374
OSPF, 350–354
RIP, 332–335
Infranet Auth
choice of, 270
configuration of, 276–277
function of, 275–276
settings, 276
Infranet Authentication
description of, 265
IE configuration, 267–269
IE properties, 266–267
policy-based, 275–277
UAC product overview, 265–266
Infranet Controller (IC), 265–266
Infranet Enforcer (IE)
authentication process, 265–266
configuration of, 267–269
for network access control, 56
properties of, 266–267
ingress maximum bandwidth, 199
ingress policing, 197
ingress traffic, 230–231
initial hold-down value, 625
initial state, VSD, 611
insertion, router, 346–347
integrated security application, 61
Integrated Security Gateway (ISG), 60, 78–81
integrity, 20
interactive applications, 192–194
interface modes
of Juniper firewall, 58
NAT mode, 458–459
overview of, 477
route mode, 459
. *See also* transparent mode
interface types
Function zone, 125–126
loopback, 126
security zone, 123–125
tunnel, 126
interface-based NAT
description of, 712
policy-based NAT *vs.*, 449
interface-based source NAT
description of, 407–408
function of, 404
properties of, 408–409
interface-based traffic shaping, 197–199
interfaces
bandwidth, configuring, 209–210
bandwidth properties of, 199
binding policy to, 392
dual untrust for redundancy, 592–597
Ethernet/serial for redundancy, 591
failing over between, 592
local in NSRP-lite, 646–651
monitoring, 626, 629–630, 634–637
NetScreen-500 configuration, 78
OSPF, properties of, 345–349
position within hierarchy, 295
redundant, creating, 652–654, 685–686
RIP, configuring on, 327–329
RIP settings per, 325–326
serial, 601
traffic shaping and, 197–198
transparent mode, converting to, 464–465
VIP definition and, 429
virtual, bandwidth properties of, 199
VLAN1, transparent mode on, 462–464
VSYS network, 731–732
internal authentication server, 252–253
internal router, OSPF, 338
internal threats, 21, 42
Internet
access, providing HA with NSRP-Lite, 642–646
firewall development and, 26–27
information security and, 19
SSG firewalls and, 354
TCP/IP for, 3, 6
threats to security from, 21–23
VPNs and, 552
Internet Assigned Numbers Authority (IANA), 12
Internet Control Messaging Protocol (ICMP)
description of, 10–11
flood protection, 494
fragment, 499
length validation, attack signatures, 499–500
network scan detection, 494
Internet key exchange (IKE)
AutoKey configuration, 566–567
description of, 553
heartbeats, gateway redundancy and, 578
IKE user type, 241–244, 253, 284–289
IPSec tunneling and, 557
overview of, 555–556
route-based VPNs and, 569
in site-to-site VPNs, 561
Internet Protocol (IP)
connectivity example, 7–8
fragmentation, attack signatures, 498–499
function of, 6
IP packet communication and, 9–11
Internet Service Provider (ISP), 12
interzone policy, 58, 163, 579
intra-zone blocking
description of, 723
on Juniper firewall, 707
route-based VPNs, 715
intrazone policy, 58, 163
intrusion detection, 25
Intrusion Detection and Prevention (IDP)
DI signature with, 523
firewall, function/features, 51, 54–56
function of, 25, 50
integration with ISG firewalls, 78
of Juniper firewall, 59–60
intrusion prevention, 59–60
IP. *See* Internet Protocol
IP address
address book entries and, 168
Address Resolution Protocol and, 16
allocation, 12
appearance of, 11–12
DIP, 422–428
of external authentication servers, 255
ICMP network scan detection, 494
interface-based source NAT, 407–409
L2TP user configuration with, 250–252
MIP, 409–417
NAT, advantages of, 402–403
NAT and, 400
NAT for private IP address, 13
policy creation and, 166
policy-based destination NAT, 433–443
policy-based source NAT, 417–428
port address translation, 401–402
ranges as objects, 188–189
source NAT and, 406–407
source/destination NAT, 444–445
VIP, 429–433
in XAuth user, group configuration, 245–248
IP option validation, 498
IP packet header, 8–9
IP packets
communication process, 8–11
NAT for private IP address, 13
traffic flow through firewall, 43–44
IP Pool
caution when defining, 248
L2TP user configuration with, 250
XAuth user and, 245, 247
IP session, 493
IP tracking
description of, 626
to determine failover, 601–604
to determine VPN availability, 632–634
managed interface and, 688

monitoring, 634–637
NSRP, overview of, 630–632
ipconfig command, 11, 12
IPSec
IPsec VPNS with NetScreen firewalls, 52
key management, 555–556
modes, 553
NetScreen-Remote VPN Client connection to, 65
overview of, 552–553, 581
protocols, 553–555
security associations, 556
traffic shaping and, 214
tunnel negotiations, 556–559, 581–582
VPN clients for security, 25
VPN connection for security, 23–24
VPN tunnels, 265–266
ISG (Integrated Security Gateway), 60, 78–81
ISP (Internet Service Provider), 12

J

Juniper Engineering Security Team, 503
Juniper firewall
administrative users, 93–95
command line interface, 99–103
configuration, first-time, 121–122
interface modes, 478
interfaces, types of, 123–126
local file system/configuration file, 95–99
management interface, securing, 104–118
management of, 90–93
NAT features, 404–405
network configuration, 131–142
overview of, 90
packet flow, 405–406
policy capacity for, 419–420
ScreenOS, updating, 118–119
security zones configuration, 126–131
source NAT and, 406
on static multicast routes, 320–321
system recovery, 119–121
system services configuration, 142–153
user authentication options of, 234
virtual routers on, 123, 295–298
VRs on, 295–298
WebUI, 103
zones, types of, 122–123
Juniper firewall core technologies
application-level inspection, 60–61
device architecture, 61–62
interface modes, 58
intrusion prevention, 59–60
policies, 58–59
virtual routers, 57–58
VPN, 59
zones, 57
Juniper firewall, troubleshooting
debug utility, 703–704
debugging, 706–712, 721
FSA, 704–705
get arp, 698–700
get interface, 698
get policy, 697
get route, 697–698
get session, 696–697
get system, 700–702
methodology, 690–692
NAT, debugging, 712–713
NetScreen logging, 717–719
network, 706
NSRP, debugging, 715
overview of, 690
ping, 693–695
snoop, 704
tools for, 692–693, 721
trace-route, 695–696
traffic shaping, debugging, 715–717
VPNs, debugging, 713–715
Juniper NAT, 404–405
Juniper NetScreen firewall
Antivirus support, 533–534
best practices, 542–543
design of, 2
device architecture, 61–62
intrusion prevention, 59–60
logging, 717–719
NAT for private IP address, 13
NSRP cluster, adding to, 619–620
policies in, 58–59
reasons to use, 46
security features of, 480
SOHO appliances, 591–592
specific layers for, 3
TCP/IP behavior anomaly detection, 491–497
tiers of, 50
traffic flow through, 43–44
virtual router, 57–58
VPN features, 59
Juniper NetScreen firewall products
choice of right tool, 64–65
enterprise class, 76–81
enterprise management, 83–84
firewalls, 52–53
high end, 72–76
IDP product, 54–56
midrange line, 70–72
NetScreen-Remote Client, 65
overview of products, 51
product line, 63–64
service provider class, 81–83
SOHO line, 66–70
SSL VPN, 53–54
UAC product, 56
Juniper Networks
IDP, 54–56
Juniper firewalls, 52–53
Secure Access SSL VPN, 53–54
security products overview, 50–51
Unified Access Control product, 56
Juniper Networks firewalls
choice of right tool, 64–65
enterprise class, 76–81
enterprise management, 83–84
high end, 72–76
midrange line, 70–72
NetScreen-Remote Client, 65
overview of, 52–53
product line, 63–64
service provider class, 81–83
SOHO line, 66–70
Juniper policies. *See* firewall policies
Juniper routers, 27–28
Juniper Security Center, 532
Juniper Security Manager, 743
Juniper security research team, 483–484
Juniper SSG firewall products
device architecture, 62
enterprise class, 76–81
enterprise management, 83–84
features of, 52, 70
high end, 72–76
midrange line, 70–72
NetScreen-Remote Client, 65
product line, 63–64
service provider class, 81–83
Small Office/Home Office (SOHO) line, 66–70
SSG 5/SSG 20, 66–68
SSG 140, 70–71, 72
SSG 520, 72–74, 75–76
SSG 550, 76–77, 78
WAN interface support of, 65

K

Kapersky Lab, 52, 69
keep alive parameter, 535
key
Deep Inspection license key, 507
Internet key exchange, 555–556
shared key, 257
. *See also* licensing; public key cryptography

L

L2TP user type
configuration of users, groups, 250–252
LDAP server's support of, 263
Local authentication server's support of, 253
properties of, 249–250
RADIUS server's support of, 257
SecureID server's support of, 261
XAuth and, 291
L4/L7 firewalls, 2
L7 protocol attacks, 501
LAND attack protection, 501
latency sensitive traffic, 194
layer 2 switches, 615–616
Layer 2 Tunnel Protocol (L2TP) VPNs, 575–576
Layer 2 Tunneling Protocol. *See* L2TP user type
layer 2 zone
creating custom, 465–466
description of, 460
for transparent mode, 478
layer 3 devices, 462
Layer-2 detection, 55
layers
data link layer communication, 15–17
OSI model layers *vs.* TCP/IP layers, 6–7
LDAP. *See* Lightweight Directory Access Protocol
learning
RIP, controlling, 330–332
when to use, 397–398
least privilege, 18
licensing
Deep Inspection license key, 507
for Juniper features, 549
for NetScreen 5-GT, 69, 70
NetScreen-500, 78
lifetime values, VPN, 586
Lightweight Directory Access Protocol (LDAP)
for Auth user type authentication, 240
for L2TP user authentication, 250
server, 262–263
for XAuth user type authentication, 245
Link State Advertisements (LSAs), 340
links
forcing down on backup firewall, 651
HA, connecting via switches, 618–619
HA, crossing, 687
HA, directly connected, 617–618
HA dual, value of, 612–613
local authentication
for Admin accounts, 235–237
of Auth user type, 240
of L2TP user, 249
for XAuth user type, 244
Local authentication server, 252–253
local file system, 95–99
local interfaces, 646–651
logging
NetScreen, 717–719, 722
NSM and, 83
policies and, 159
login
Administrator login process, 238
XAuth user type, 244
LSAs (Link State Advertisements), 340

M

MAC address
ARP query for, 461–462
data link layer communication, 15–16
gateway replacement and, 700
mailing lists, NetScreen products, 584
malware, 486–490
management interface, Juniper firewall, 104–119
management zone (MGT), 57
manual attacks, 484–485
manual key VPNs, 556–557
many-to-many mapping, 439–441, 442
many-to-one mapping, 436–439
map to IP, 430
map to port, 434
map to service, 430
Mapped IP (MIP)
configuration, 411, 413, 415
for destination NAT, 405
function of, 404
limitations of, 410–411

overview of, 409
properties of, 410
property, 410
scalability of, 449–450
scenarios, 411–417
as source/destination NAT, 428
VIP *vs.*, 449
mapping, 433–441, 442
Mastering Regular Expressions, 2nd Edition (Friedl), 515
match groups, 386–387
match strings, 515–518
maximum bandwidth, 202–203
maximum users, 279
memory allocation, 61
messages, BGP, 356
metrics, route, 299
Microsoft, MS-RPC, 541
midrange line, Juniper Networks, 70–72
MIP. *See* Mapped IP
MMD (multi-method detection), 55–56
modes, IPSec, 553
monitoring
data, storage of, 543
interface/zone/IP tracking, combining, 634–637
NSRP interface, 627–628
NSRP optional, 626–627
NSRP zone, 629–630
VPNs, 577–578
Morris, Richard, 481
Morris worm, 1988, 481–482
MTG (management zone), 57
multicast routing, 320–321
multicast routing tables, 298
multi-method detection (MMD), 55–56
multitunnel VPNs, 580
multivector malware, 489–490
MyDoom, 490

N

name
cluster, 620
of external authentication servers, 255
naming convention errors, 170
NAT. *See* Network Address Translation
National Security Agency (NSA), 542
negate character class, 517
negation, 181–182
neighbor relationships, OSPF, 339–340
neighbors, BGP, 361–364, 372
netmask
in MIP configuration, 412, 413, 415
MIP property, 410
NetScreen 5-GT, 66–68, 69–70
NetScreen 5-XT, 66–68, 69
NetScreen appliances, 582–583
NetScreen Redundancy Protocol (NSRP)
Active/Active cluster, setting up, 657–664
debugging, 715, 722
firewall virtualization, 608–610
full, taking advantage of, 686
full-mesh Active/Active setup, 664–670
HA links, dual, 612–613
NSRP states, 610–612
NSRP-lite *vs.*, 654
overview of, 608, 684
RTO mirroring, 655–656
NetScreen Redundancy Protocol (NSRP) clusters
building, 613, 684
configuration synchronization, 621–624
firewall-to-router, connecting via switches, 615–616
firewall-to-router direct connection, 613–615
full-mesh, cabling for, 616–617
HA links, connecting via switches, 618–619
HA links, directly connected, 617–618
NetScreen, adding to, 619–620
NetScreen Redundancy Protocol (NSRP) failovers
IP tracking, 630–637
monitoring, interface, 627–628
monitoring, optional, 626–627
monitoring, zone, 629–630
NSRP heartbeats, using, 624–625
when to, 624
. *See also* NSRP-Lite
NetScreen Remote, 570–575
NetScreen Security Manager (NSM)
clusters, adding members to, 677–679
clusters, creating, 676–677
enterprise management with, 83–84
function of, 53
HA, configuring through, 676
NetScreen-Hardware Security Client management, 69
NSRP parameters, configuring, 680–681
policy administration from, 58–59
storage for firewall devices, 63
VSD, configuring, 682
NetScreen SOHO appliances
dial-up, falling back to, 597–600
dual untrust interfaces for redundancy, 592–597
failing over between interfaces, 592
improving availability with, 684
IP tracking to determine failover, 601–604
product line, 66–70
restricting policies to subset with serial interface, 601
VPN monitoring to determine failover, 604–608
NetScreen-25, 70–72
NetScreen-50, 70–71, 72
NetScreen-204, 72–75
NetScreen-208, 64, 72–75
NetScreen-500, 76–78
NetScreen-5200, 81–83
NetScreen-5400, 81–83
NetScreen-Hardware Security Client, 66–68, 69
NetScreen-ISG 1000, 76–77, 78–80
NetScreen-ISG 2000, 76–77, 78–81
NetScreen-Remote Client, 52, 65
NetScreen-Remote Security Client, 65
NetScreen-Remote VPN Client, 65
NetScreen-Security Client, 52
network
broadcast networks, 339
Juniper firewall configuration, 131–132
non-broadcast-multiple-access networks, 339
point-to-point networks, 339
segmentation for transparent mode, 466–470
threats, 25, 26
troubleshooting, 706, 721
with/without DMZs, 38–40
Network Address Translation (NAT)
advantages of, 449
debugging, 712–713, 721
destination NAT, 428–445
interface mode, 458–459
Juniper NAT overview, 404–405
Juniper packet flow, 405–406
overview of, 400–404
policies and, 159
for private IP address, 13
security policy need and, 450
source NAT, 406–428
network honeypot, 55
network interfaces, VSYSs, 731–732
network layer, 5, 6
Network LSA (2), 340
network object, 465–466
network protocols
attacks, 490–491
OSI model for, 4
protocol anomaly detection, 498–501
networking
coverage overview, 2–3
data link layer communication, 15–17
Internet Protocol, 6–8
IP address, 11–12
IP address allocation, 12
IP packets, 8–11
knowledge of, 47
NAT for private IP address, 13
OSI model, 3–6
ports, TCP/UDP, 14–15
TCP communications, 13–14
UDP communications, 14
Nimda worm, 22
NMap, 492, 494–495
no-brain problem, 674–676, 687
non-broadcast-multiple-access networks, 339
nonrepudiation, 20
Not So Stubby Area (NSSA), 338
novelty traffic, 195
NSA (National Security Agency), 542
NSM. *See* NetScreen Security Manager
NSRP. *See* NetScreen Redundancy Protocol (NSRP) clusters; NetScreen Redundancy Protocol (NSRP) failovers
NSRP-Lite
basic usage, 642–646
local interfaces, working with in, 646–651
on midrange appliances, 641–642, 685
NSSA (Not So Stubby Area), 338
NSSA LSA (5), 340

O

one-to-many mapping, 429
one-to-one mapping
destination NAT scenario, 435–436
destination PAT scenario, 441–443
with DIP shift, 426
with MIP, 409
policy-based source NAT for, 419
Open Shortest Path First (OSPF)
area properties, 343–345
areas and, 336–339
autonomous systems and, 336
configuration, 341, 350–351
informational commands, 350
interface properties, 345–349
interface status, showing, 352
link state advertisements, 340
link state protocol properties, 397
neighbor relationships, 339–340
neighbors/LSA database, 352–353
overview of, 335, 394
routing table, 353–354
in VRs, 341–343
Open System Interconnection (OSI) model
application layer, 4
data link layer, 5–6
data link layer communication, 15–17
layers of, 3–4
network layer, 5
physical layer, 6
presentation layer, 5
session layer, 5
TCP/IP layers *vs.*, 6–7
transport layer, 5
use of, 46
operating system (OS)
firewall on/firewall integrated with, 30–31
in NetScreen firewall architecture, 61
updates, 543
ordering
least-to-most restrictive, 188
policy via CLI, 183
policy via WebUI, 180–182
OS. *See* operating system
OSI model. *See* Open System Interconnection (OSI) model
OSPF. *See* Open Shortest Path First
out-of-the-box policy, Juniper firewall, 164

P

packet
ARP, adjusting number sent after failing over, 671
Deep Inspection and, 502
dual HA links and, 612
filters, 27–28
flow, 405–406, 415
IP connectivity, 7–8
IP packets, 8–11, 43–44
logic, 164–165
NetScreen exchanges, types used in, 612
network layer and, 5

in TCP communications, 13–14
viewing contents with snoop, 723
packet capture program, 425–426
Panix SYN flood, 1996, 482
password
cracking, 259–260
of RADIUS server, 256
SecurID server and, 260–262
PAT. *See* Port Address Translation
patches
operating system updates, 543
patch management for security, 25
updating, 22
virtual patch with Deep Inspection, 503
PBR. *See* policy-based routing
peers, BGP, 355
people hacking, 22
performance
application proxy and, 28–29
debug command and, 703
permits, explicit, 543
Phase 1
IPSec tunneling, 557–558
NetScreen Remote and, 573–574
Phase 2
IPSec tunneling, 558–559
NetScreen Remote and, 573–574
policy-based VPNs and, 564
phishing, 480, 487
physical interfaces, in VSYSs, 732–734
physical layer, 6, 7
physical security, 18
physical threats, 25, 26
PIMS (pluggable interface modules), 72, 78
ping
for firewall troubleshooting, 693–695
ICMP for, 10
ICMP fragmentation of, 499
Ping of Death attack, 499–500
PKI (Public Key Infrastructure), 560
planning
Antivirus, 533–534
Deep Inspection, 505, 507
for successful implementation, 738–739
Web filtering, 525, 527
pluggable interface modules (PIMs), 72, 78
point-to-point networks, OSPF, 339
poison reverse, 323
policies
advanced options, 229
Deep Inspection, with CLI, 512–514
Deep Inspection, with WebUI, 511–512
DIP configuration on, 422–423
five-tuple policy, 44
interzone, back-to-back VPNs and, 579
of Juniper firewall, 58–59
NetScreen firewall limits on, 88
PBR, 389–390
SCREEN settings, 491
with serial interface, 601
traffic-shaping and, 200–201, 210–215
transparent mode and, 478
for Web filtering rules, 530–531
. *See also* firewall policies
policy binding
to interface, 392
overview of, 390
to VR, 391
to zone, 391–392
policy configuration, advanced
advanced options, 215–216
counting, 216–222
overview of, 192
scheduling, 222–227
traffic-shaping, deploying, 197
traffic-shaping enforcement methods, 197–202
traffic-shaping examples, 205–215
traffic-shaping fundamentals, 192–197
traffic-shaping mechanics, 202–205
Policy Editor, IDP, 56
policy-based destination NAT, 433–443
destination PAT scenario, 441–443
function of, 405
options, 433
properties, 434–435
scenarios, 435–441
source/destination NAT combination, 444–445
when to use/not use, 434
policy-based NAT
description of, 712
interface-based NAT *vs.*, 449
policy-based routing (PBR)
action groups, 387–389
components of, 383
extended access lists, 383–386
match groups, 386–387
overview of, 383, 395–396
policies, 389–390
policy binding, 390–392
static routing *vs.*, 397
policy-based source NAT, 417–428
configuration of, 418–419
description of, 417–418
DIP, 420
DIP, configuration on policy, 422–423
DIP properties, 421
DIP shift, 426–428
function of, 404–405
policy capacity, 419–420
Sticky DIP, 423–425
policy-based user authentication, 269–277
description of, 269–270
Infranet Authentication, 275–277
User Auth, policy configuration with, 270–272
User Auth properties, 270
Web Auth, authentication with, 272–275
policy-based VPN
debugging, 714
description of, 59
in NetScreen appliances, 563–564
site-to-site, 564–569
Port Address Translation (PAT)
destination PAT scenario, 441–443
DIP and, 420
DIP pool configuration and, 421, 423
source/destination PAT, 401–402
port density, 80, 82
port modes, 69
port scanning, 492–495
ports
FTP ALG and, 540–541
of Juniper Networks' high-end line, 75
LDAP server port, 262
of NetScreen-25, 72
port control, 279
RADIUS server port, 256
of SSG 550, 78
TCP/UDP, 14–15
transport layer and, 5
VIP properties and, 430
position, 167
preference, route, 299
presentation layer, 5
pre-shared key, 572
primary interface, 654
priority queuing, 202, 203
priority-based traffic shaping, 196, 197
private IP address, 13
profile
Antivirus profile settings, 537–538, 539
virtual systems, 739–740
protocol anomalies, 510
protocol anomaly detection
ICMP length validation, attack signatures, 499–500
IDP attack detection, 55
IP fragmentation, validation, attack signatures, 498–499
IP option validation, 498
L7 protocol attacks, 501
TCP attack signatures, 501
TCP flag validation, 500–501
protocol shaping, 203
protocols
attack objects and, 510
Deep Inspection and, 502, 503–504
Diffie-Hellman, 558
IPSec, 553–555
OSI model and, 4, 5–6
protocol anomaly detection, 498–501
routing, attacking, 329
tunneling, traffic shaping on, 214
. *See also* specific protocols
proxy IDs, 565, 570
public key cryptography
certificates, 560–561
CRLs, 561
overview of, 559–560, 582
PKI, 560
Public Key Infrastructure (PKI), 560

Q

quality of service (QoS). *See* traffic-shaping
queue size, 497

R

RADIUS server
802.1x support, 292
for Auth user type authentication, 240
authentication capabilities, 257
configuration of, 257–259
external Admin authentication, 238–239
for L2TP user authentication, 250
properties of, 256–257
for XAuth user type authentication, 244
rate limiting
TCP SYN, 495–497
UDP, 497
Read Only access, 238
Read/Write access, 238
real-time applications, 192–194
real-time operating system (RTOS), 61
reauthentication period, 279
reconnaissance
in Black Hat attack, 486–487
detection of, 492–493
as hack step, 484
in script kiddie attack, 485
TCP SYN host scan, limiting, 494–495
recovery attempts, 578
redirect, 276
redistribution, route
into BGP, 380–382
in Juniper firewall, 375–376
into OSPF, 378–380
overview of, 311, 375, 395
between routing protocols, 376–378
redundancy
with dual untrust interfaces, 592–597
gateway, 578–579
interfaces, 652–654
. *See also* NetScreen Redundancy Protocol
redundant interfaces
creating, 652–654, 685–686
physical interfaces, grouping into, 652–653
primary interface, changing, 654
simple setup, 653–654
reflectors, route, 370–371
regular expressions, 515–518
relationships, OSPF, 339–340
remote access
NetScreen Remote Client for, 65
with Secure Access SSL VPN, 53–54
Remote Procedure Call, 541
Request for Comment (RFC) 1631, 402
resource control, 739–740
resources
for DH protocol, 558
Juniper discussion forum, 584
NetScreen Mailing List Archive, 584
shared, 743
for syslog filtering systems, 716–717
. *See also* Web site links
retransmissions, 280
retry timeout, 256
retry times, 256
RIP. *See* Routing Information Protocol
rip config, 350–354
risks, 23
root, policy creation, 177–178
Root Admin account, 235
Root-Level Read-Only Admin account, 235

Root-Level Read/Write Admin account, 235
route aggression, 368–370
route flapping, 358
route maps
  example of, 310–311
  overview of, 306
  properties, 308–309
route mode, 459
route redistribution
  into BGP, 380–382
  in Juniper firewall, 375–376
  into OSPF, 378–380
  overview of, 311, 375, 395
  between routing protocols, 376–378
route reflection, 357
route reflectors, 370–371
route-based VPN
  debugging, 714–715
  description of, 59
  in NetScreen appliances, 569
  VPN monitoring and, 577–578
Router LSA (1), 340
routers
  firewalls, connecting directly to, 613–615
  firewalls, connecting to via switches, 615–616
  ICMP and, 11
  insertion, malicious OSPF, 346–347
  packet filter in, 27–28
  virtual routers of Juniper firewall, 57–58
  . *See also* virtual routers
routes
  dual VR default, 723
  import/export of, 311–313
  preference *vs.* metrics, 299
  summary, OSPF, 344–345
  types of, 296
routing
  action groups, 387–389
  BGP, configuring, 358–371
  BGP, overview of, 354–358
  BGP informational commands, 372–375
  IP packet, 9
  match groups, 386–387
  OSPF, configuring, 341–349
  OSPF, informational commands, 350–354
  OSPF, link state advertisements, 340
  OSPF, neighbor relationships, 335–340
  OSPF, overview of, 335–339
  overview of, 294
  policies, 389–390
  policy binding, 390–392
  policy-based routing, 383–392
  protocols, attacking, 329
  RIP, informational commands, 332–335
  RIP, overview of, 321–331
  route redistribution, 375–382
  selection process, 298–299
  static, 313–321, 393–394
Routing Information Protocol (RIP)
  concepts, 322–323
  configuring on interface, 327–329
  enabling within VR, 327
  information, summarizing, 332–333
  informational commands, 332
  interface state, 333–334
  learned/advertised routes, 330–332
  neighbors, 334
  overview of, 321–322, 394
  properties in VRs, 323–325
  rip config, 333
  routes/database, 334–335
  settings via interface, 325–326
  v.1 *vs.* v.2, 322
  when to use, 396
routing selection process, 298–299
routing tables
  BGP, displaying, 374
  destination-based, 297
  multicast, 298
  source interface-based, 297–298
  source-based, 297
  virtual router and, 57, 296
RTO mirroring
  description of, 654
  enabling in NSRP cluster, 655
  preventing backup of sessions, 655–656
  synchronizing state with, 655
RTOS (real-time operating system), 61
rules
  Antivirus, 538–539
  Web filtering, 530–531

S

SAD (security association database), 556
SAs. *See* security associations
SBR (source-based routing), 304–305, 396
scalability, 449–450
Scan Manager, 536–537
scanning, 492–495, 500–501
scheduling
  configuring, 222–225
  description of, 215–216
  policies and, 160
  properties, 222
  traffic denial and, 232
  traffic-shaping and, 222, 225–227
SCREEN features, 480
SCREEN settings
  for security, 490–491
  TCP/IP behavior anomaly detection, 491–497
  TCP/IP protocol anomaly detection, 498–501
ScreenOS
  Antivirus support, 533–534
  Juniper firewall, updating, 118–119
  in NetScreen firewall architecture, 61
  Web filtering support, 525–526
ScreenOS 5.1, 507, 514–515
script kiddies, 21, 484–485
search, 515–516
search algorithm, 519
Secure Access SSL VPN
  features of, 53–54
  function of, 50, 51
secure application manager, 54
Secure Internet Protocol, 403
Secure Meeting product, 50, 51, 54
Secure Shell (SSH), 236, 237
Secure Sockets Layer (SSL)
  Juniper SSL VPN product line, 50, 53–54
  WebAuth connections over, 273
SecurID
  for Auth user type authentication, 240
  for L2TP user authentication, 250
  server, 260–262
  for XAuth user type authentication, 245
security
  best practices, 542–543
  concepts, 18–19
  definition of, 18
  DMZ design and, 42
  firewalls for, 2
  information security, 19–20
  Internet and threats, 21–23
  need for, 17, 46
  physical, network, application threats, 25–26
  standards, 17–18
  technologies for, 24–25
  threats, identification of, 23
  transparent mode and, 466–467
  troubleshooting and, 693
  VPN connection for, 23–24
security association database (SAD), 556
security associations (SAs)
  clearing, 585
  description of, 553
  overview of, 556
security modules, 80
security research team, Juniper, 483–484
Security Services Gateway (SSG) firewall, 50
  . *See also* Juniper SSG firewall products
security zone
  configuration of, 126–131
  definition of, 57
self logging, 718
serial interfaces
  policy restriction to subset, 601
  port modes for, 597
  for redundancy, 591
server name, 279
server-to-client (STC) flows, 505
service auto detection, 430
service provider class, Juniper Networks, 81–83
services
  custom, creating, 173–174
  groups, 175–176
  modifying/deleting, 174–175
  objects *vs,* groups, 189
  policies and, 159
  policy creation and, 166, 172–173
session layer, 5
session table, 164
sessions
  802.1x, checking, 283–284
  IP session limiting, 493
  session layer to control, 5
set arp always-on-dest command, 637
set filter command, 723
set nsrp monitor track-ip command, 634
shared interfaces, 732, 735–738
shared key, 257
shared resources, 743
shared secret, 256
SIBR (source interface-based routing), 297–298, 305
signatures
  antivirus, 533
  creation of, 519–523
  of Deep Inspection, 504–505, 507
  Deep Inspection, automatic updates, 508–509
  Deep Inspection contexts, regular expressions, 514–518
  Deep Inspection, manual updates, 509–510
  definition of, 510
  writing with IDP, 523
silent period, 280
Simple Network Management Protocol (SNMP), 577
single firewall, 40
single firewall with bastion host, 40
single firewall with screened subnet/bastion host, 40
site-to-site policy-based VPNS, 564–569
site-to-site VPNs, 561–563
slow scan, 495
Small Office/Home Office (SOHO). *See* NetScreen SOHO appliances
SMTP-From context, 503
SNMP (Simple Network Management Protocol), 577
snoop, 704, 723
social engineering, 22, 486
SOHO (Small Office/Home Office). *See* NetScreen SOHO appliances
source address, 159
source interface, 255
source interface-based routing (SIBR), 297–298, 305
source interface-based static routes, 319–320
source NAT, 406–428
  description of, 400–401
  with destination NAT, 444–445
  function of, methods of, 404–405
  interface-based, 407–409
  Juniper firewall and, 406
  MIP, 409–417
  policy-based, 417–428
source PAT, 401–402, 404
source threshold, 495–496
source translation, 444
source-based routing (SBR), 304–305, 396
source-based routing tables, 297
source-based static routes, 317–318
spammers, 488–489
Spanning Tree Protocol (STP), 634–635
split-brain problem, 673–674, 686
spoofing
  IDP attack detection, 56
  protection with NetScreen firewall, 495, 496
  source address of packets, 429

spyware, 480
Squid, 543
SSG (Security Services Gateway) firewall, 50
. *See also* Juniper SSG firewall products
SSG 5, 66–68, 70
SSG 20, 66–68, 70
SSG 140, 70–71, 72
SSG 520, 72–74, 75–76
SSG 550, 76–77, 78
SSH (Secure Shell), 236, 237
SSL. *See* Secure Sockets Layer
standards, security, 17–18
stateful firewalls, 300
stateful inspection, 29–30
stateful signatures, 55
states, 610–612
static address
L2TP user configuration with, 251–252
XAuth user with, 245–247
static NAT, 409
static routes, 296
static routing
on Juniper firewall, 314–317
multicast routing, 320–321
overview of, 313–314, 393–394
PBR *vs.*, 397
source interface-based, 318–320
source-based, 317–318
statistics, 802.1x, 283–284
STC (server-to-client) flows, 505
Sticky DIP, 421, 423–425
STP (Spanning Tree Protocol), 634–635
string matching, 515–518
stripping separator, 255
structured attacks, 22
Stub Area, 338
subinterfaces
traffic-shaping and, 199
in VSYSs, 732, 734–735
subnet, 409, 410, 417
subset, 601
sub-shells, 185
Summary LSA (3), 340
summary routes, 344–345
SurfControl Integrated Mode
availability of, 548–549
Web filtering configuration with, 529–530
SurfControl Redirect Mode, 528–529
sweep, 494
switches
connecting firewalls to routers via, 615–616
in network communications, 16–17
Symantec, DNS ALG, 541–542
SYN flag, 500
SYN packet, 14
synchronization, 621–624
Syslog, 716–717
system recovery, Juniper firewall, 119–121
system services configuration, 142–153

T

tables, session, 164
TCP. *See* Transmission Control Protocol
TCP (Transport Control Protocol), 194
TCP SYN
host scan, limiting, 494–495
rate limiting, 495–497
TCP flag validation, 500–501
TCP SYN cookie, 496–497
TCP SYN flood, 495–497
TCP SYN host scan, 494–495
TCP/IP. *See* Transmission Control Protocol/Internet Protocol
Teardrop attack, 499
Telnet banner message, 284–287
testing
Antivirus protection, 540
Web filtering protection, 531–532
TFTP server, 509
threats
attack types, 480–481
identification of, 23
physical, network, application threats, 25–26
to security from Internet, 21–23
unified threat management, 482
vulnerability databases, 482–483
three-way handshake, 13
threshold, failover, 626–627, 631
throughput, 80, 82
timeout, 255, 496–497
timers, 323, 325
tools, troubleshooting. *See* Juniper firewall, troubleshooting
Totally Stubby Area, 338
trace-route, 695–696
traffic
authentication dangers, 264–265
deep inspection of, 52
egress filtering, 543
egress *vs.* ingress, 230
firewall function, 2
handling, default, 204–205
logging, 717–718
policies for, 58
policy creation and, 166–167
tunneling, 568
traffic alarms
configuring, 220–222
counting and, 216, 217–218
policies and, 160
traffic anomaly detection, 55
traffic bandwidth setting, 715–716
traffic classification
IP-based, 729
in shared zone, 737–738
VLAN-based, 728
in VSYSs, 728
traffic flow
concepts of, 35–38
DMZ design and, 39, 43–44
traffic-shaping
bandwidth-based, 195–196
debugging, 715–717, 722
deploying on firewalls, 197
enforcement methods, 197–202
examples, 205–215
mechanics, 202–205
overview of, 192–194, 228–229
planning for, 231–232
policies and, 160
priority-based, 196
scheduling and, 225–227
selecting type of, 196–197
traffic types, 194–195
translate to IP
many-to-one mapping, 437
one-to-one mapping, 435
policy-based destination NAT property, 434
source/destination NAT, 444
translate to IP range
many-to-many mapping, 440
policy-based destination NAT property, 434
source/destination NAT, 444
Transmission Control Protocol (TCP)
attack signatures, 501
communications, 13–14
flag validation, 500–501
packets, 642
ports, 5, 14–15
signature, 521–522
Transmission Control Protocol/Internet Protocol (TCP/IP)
behavior anomaly detection, 491–497
data link layer communication, 15–17
Internet Protocol, 6–8
Internet security threats and, 21
IP address allocation, 12
IP address format, 11–12
IP packets, 8–11
NAT for private IP address, 13
network protocol attacks, 490–491
OSI model, 3–6
protocol anomaly detection, 498–501
TCP communications, 13–14
TCP/UDP ports, 14–15
UDP communications, 14
transparent mode
broadcast methods, 461–462
custom layer 2 zone/network object, 465–466
deployment options, 466
device configuration for, 462
Drop Unknown MAC option in, 497
interfaces, converting to, 464–465
Juniper firewall in, 58
layer 2 zones, 460
network segmentation, 466–470
overview of, 458–460, 477
VLAN zone, 461
VLAN1 interface configuration, 462–464
VPNs with, 470–476
Transport Control Protocol (TCP), 194
transport layer, 5, 7
transport mode, IPSec, 553
Trend Micro, 52, 534
trickling, 535
Trojan horse
definition of, 480
protection against, 489
zombies as, 488
troubleshooting
IPSec SAs, 556
virtual systems, 741
VPNs, 584–585
Trust zone
assigned to LAN, 57
interface-based source NAT and, 407, 409
policy-based source NAT and, 418
trusted users, 18
tunnel interfaces
configuring OSPF to work with, 348–349
traffic-shaping and, 199
tunnel mode, 553
tunnel zone, 57
tunnels
IPSec, negotiations, 556–559
placement of, 568
policy-based VPNs and, 563–564
VPN, monitoring, 604–608
two-way exchanges, IPSec, 557

U

UAC. *See* Unified Access Control (UAC) suite
UDP. *See* User Datagram Protocol
unicode decoder, 517
Unified Access Control (UAC) suite
features of, 56
function of, 50, 51
IE configuration, 267–269
IE properties, 266–267
product overview, 265–266
unified threat management, 482
Uniform Resource Locator (URL)
filtering, 24, 160
signature in Deep Inspection, 520
Web filtering and, 524–525
Web filtering configuration and, 528, 529–530
Web filtering rules and, 530
Web filtering testing, 531–532
University of Michigan, 262
UNIX, 21
unstructured threats, 21
Untrust zone
assigned to Internet, 57
interface-based source NAT and, 407–409
MIP address and, 417
monitoring, 629
policy-based source NAT and, 418
updates
Antivirus Scan Manager settings, 536–537
Deep Inspection, 508–510
operating system, 543
of patches, 22
URL. *See* Uniform Resource Locator
U.S. National Institute of Standards and Technology, 22
U.S. National Vulnerability Database, 483
user account types, 234–239
Admin account types, 235
Admin accounts, external authentication for, 237–239

Admin accounts, local authentication for, 235–237
authentication users, 239–252
external authentication servers, 254–269
list of, 234–235
local authentication server, 252–253
RADIUS server's support of, 257
user member of multiple, 291
User Auth
authentication with, 270
choice of, 270
policy configuration with, 270–272
WebAuth *vs.*, 273
user authentication
802.1x authentication, 277–284
authentication users, 239–252
enhancing authentication, 284–288
external authentication servers, 254–269
internal authentication server, 252–253
local authentication server, configuration of, 253
options of Juniper firewalls, 234
policies and, 160
policy-based, 269–277
user account types, 234–239
User Datagram Protocol (UDP)
communications, 14
data rate limiting, 497
flood, 497
ports, 14–15
user group assignment, 253, 257
user management, 254

V

virtual interfaces, 198, 199–200, 609–610
Virtual IP (VIP)
destination NAT, 429–433
function of, 405, 429
MIP *vs.*, 449
properties, 429–433
scalability of, 449–450
virtual links, 337
Virtual Port, 430
Virtual Private Network (VPN)
back-to-back, 579
configuration of, 583
connection, security and, 23–24
debugging, 713–715, 721–722
dial-up, 569–575
gateway redundancy, 578–579
hub/spoke, 579–580
IKE user type for, 241
IP tracking for availability, 632–634
IPSec, 552–556
IPSec tunnel negotiations, 556–559
IPsec VPNs with NetScreen firewalls, 52
Juniper firewall VPN features, 59
Juniper SSL VPN product line, 50, 53–54
L2TP, 575–576
monitoring, 577–578
monitoring to determine failover, 604–608
multitunnel, 580
in NetScreen appliances, 582
NetScreen-Remote VPN Client, 65
NetScreen-Security Manager and, 83–84
for network access security, 25
overview of, 552
policy, traffic-shaping on, 212–213
policy-based, 563–569
public key cryptography, 559–561
route-based, 569
route-based, traffic-shaping on, 213–214
route-based VPN, 59, 569, 577–578, 714–715
site-to-site, 561–563
static routing on, 314
with transparent mode, 470–476
tunnel, Juniper packet flow, 406
Virtual Router Redundancy Protocol (VRRP), 610, 631
virtual routers (VRs)
BGP and, 358–361
binding policy to, 391
configuring, 302–303
debugging Juniper firewall and, 706–707
default route preferences, 304
destination-based forwarding, 304
ECMP routing, 299–300
of Juniper firewall, 57–58, 123, 295–298
OSPF properties within, 341–343
overview of, 294, 393–394
properties of, 300–301
RIP, enabling within, 327
RIP properties in, 323–325
route maps/access lists, 306–311
route redistribution, 311
routes, import/export of, 311–313
routing selection process, 298–299
source interface-based routing example, 305
source-based routing example, 304–305
Virtual Security Device (VSD)
binding VSYS to, 671–673
configuration of, 682
failing over and, 670–671
groups, 609–610
NSRP and, 608–609
NSRP states and, 610–612
object monitoring and, 626
Virtual Security Interfaces (VSI), 609–610
Virtual System Profiles, 739–740
virtual systems (VSYSs)
administration, 729
components of, 726–727
configuring, 729, 742
creating, 729–731
description of, 742
failing over, 671–673
NetScreen-500's support of, 78
network interfaces, 731–732
overview of, 726
physical interfaces, 732–734
practical uses of, 743
profiles, 739–740
shared interfaces, 735–738
subinterfaces, 734–735
traffic classification, 728–729
troubleshooting, 741
workings of, 728, 742
virus
antivirus software, 24
Brain virus, 1986, 481
definition of, 480
traits of, 532
. *See also* Antivirus
VLAN zone, 461
VLAN1, 478
VLAN1 interface, 462–464
VLANs, 743
Voice over Internet Protocol (VoIP), 195
VPN. *See* Virtual Private Network
VRRP (Virtual Router Redundancy Protocol), 610, 631
VRs. *See* virtual routers
VSD. *See* Virtual Security Device
VSI (Virtual Security Interfaces), 609–610
VSYS Read-Only Admin account, 235
VSYS Read/Write Admin account, 235
VSYSs. *See* virtual systems
vulnerability databases, 482–483

W

WAN (Wide Area Network), 52
Web filtering
concepts of, 524–525
configuration of, 527–530
definition of, 524
planning, 525, 527
rules, 530–531
support of Juniper firewalls, 526
testing, 531–532
Web filtering, configuration of
SurfControl Integrated Mode, 529–530
SurfControl Redirect Mode, 528–529
WebSense Redirect Mode, 527–528
Web server, 527
Web site links
CERT Vulnerability Notes Database, 483
CVE naming standard, 483
defense-in-depth paper, 542
European Institute for Computer Antivirus Research, 540
Juniper Security Center, 484
NAT, 446
NetScreen updates, 508
U.S. National Institute of Standards and Technology, 22
virus advisories, 532
. *See also* resources
Web User Interface (WebUI)
Deep Inspection policy with, 511–512
Juniper firewall, 103
Juniper firewall management with, 53
policy administration from, 58–59
policy creation via, 177–180
policy options, 182
policy reorder in, 180–182
WebAuth
banner message, 284
choice of, 270
configuration of, 274–275
properties of, 272–273
Web-based resources, 54
WebSense Redirect Mode, 527–528
White Hat hackers, 486
Wide Area Network (WAN), 52
Windows XP, 576
WinNuke attack protection, 501
wireless 802.1x authentication, 282–283
wireless access, 488
Wireshark, 425–426
worms
definition of, 480
early, 487
history of, 489–490
Morris worm, 1988, 481–482
protection against, 489
zombies, 488

X

XAuth group, 244, 245–247
XAuth user, 244, 245–247
XAuth user type
configuration of, 245–247
function of, 244
with IKE for single user, 248–249
L2TP user type and, 291
LDAP server's support of, 263
local authentication server's support of, 253
properties of, 244–245
RADIUS server's support of, 257
SecureID server's support of, 261

Z

zero-day exploits, 22
zombie, 488–489
zone isolation, 542
zone verification, 257, 280–281
zones
binding addresses to, 189
binding policy to, 391–392
function of, 50
of Juniper firewall, 57, 122–123
monitoring, 626, 634–637
network simplification with, 87
policy creation and, 166, 167–168
position within hierarchy, 295