

THE EXPERT'S VOICE® IN CISCO



Cisco Routers for the Small Business

A Practical Guide for IT Professionals

Securely configure Cisco's 800 and SOHO series routers using IOS—Cisco's powerful, Internetwork Operating System.

Jason C. Neumann

Apress®

Cisco Routers for the Small Business

A Practical Guide for
IT Professionals



Jason C. Neumann

Cisco Routers for the Small Business: A Practical Guide for IT Professionals

Copyright © 2009 by Jason C. Neumann

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-4302-1851-7

ISBN-13 (electronic): 978-1-4302-1852-4

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editor: Jonathan Gennick

Technical Reviewers: Dean Olsen, Sebastien Michelet

Editorial Board: Clay Andres, Steve Anglin, Mark Beckner, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Michelle Lowman, Matthew Moodie, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Project Manager: Sofia Marchant

Copy Editor: Octal Publishing, Inc.

Associate Production Director: Kari Brooks-Copony

Production Editor: Kari Brooks-Copony

Compositor: Pat Christenson

Proofreader: Katie Stence

Indexer: Broccoli Information Management

Artist: April Milne

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit <http://www.apress.com>.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at <http://www.apress.com/info/bulksales>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

Contents at a Glance

About the Author	xvii
About the Technical Reviewers	xix
Acknowledgments	xxi
Introduction	xxiii
■ CHAPTER 1 Getting to Know Your Router	1
■ CHAPTER 2 Configuring Your Router	17
■ CHAPTER 3 Configuring DSL Using PPPoE	57
■ CHAPTER 4 Configuring a VPN Using IPSec	81
■ CHAPTER 5 Beyond the Basics	105
■ CHAPTER 6 Understanding Binary and Subnetting	143
■ CHAPTER 7 Routing—What Routers Do Best	157
■ CHAPTER 8 Understanding Variable Length Subnet Mask Networking	173
■ APPENDIX A Sample Configuration for a Cable Modem	183
■ APPENDIX B Sample Configuration for DSL and PPPoE	189
■ APPENDIX C Sample Configuration IPSec VPN Over DSL	197
■ APPENDIX D CCNA CLI Command Reference	207
■ APPENDIX E ACL and Firewall Names Used in This Book	231
■ INDEX	233

Contents

About the Author	xvii
About the Technical Reviewers	xix
Acknowledgments	xxi
Introduction	xxiii
CHAPTER 1 Getting to Know Your Router	1
Understanding Your Ports	1
The Console Port	1
LAN Ethernet Ports (E0 or VLAN1)	2
WAN Ethernet Port (E1 or FA4)	2
Connecting to Your Router	2
Attach the Console Cable	3
Configure Hyper Terminal	3
Power Up the Router	5
Welcome to the Command Line	6
Your First CLI Commands	7
Turn On Privileged EXEC Mode	9
Set the Date and Time	9
Get Help	10
Using Global Configuration Mode	10
Set Your Router's Hostname	11
Set the Privileged EXEC Mode Password	11
Display and Save Your Configuration	12
Summary	13
Ports	14
User EXEC Mode Commands	15
Privileged EXEC Mode Commands	15
Global Configuration Mode Commands	16
Display and Save Your Configuration	16

CHAPTER 2	Configuring Your Router	17
	Erasing the Startup Configuration	17
	Learning Some CLI Tips and Tricks	19
	Use Keyboard Shortcuts	19
	Suppress Console Messages	19
	Undo the Effects of a Command	20
	Configuring Your LAN Interface	20
	Step 1: Assign a Hostname to Your Router	21
	Step 2: Start Interface Configuration Mode	21
	Step 3: Add a Description to Your Interface	21
	Step 4: Assign an IP Address to Your Interface	21
	Step 5: Bring Up the Interface	21
	Step 6: Exit from Interface Configuration Mode	22
	Step 7: Check Your Work	22
	Configuring a DHCP Server	23
	Step 1: Define the DHCP Pool Name	23
	Step 2: Define the Network Address for DHCP	23
	Step 3: Define Your Domain Name	24
	Step 4: Define the Default Gateway	24
	Step 5: Define Your DNS Servers	24
	Step 6: Define a WINS Server (Optional)	24
	Step 7: Define a DHCP Lease Time	24
	Step 8: Define a DHCP-Excluded Address Range	25
	Step 9: Test DHCP Using a Workstation	25
	Step 10: Check Your DHCP Status with the IOS	26
	Configuring Telnet on Your Router	26
	Step 1: Set Your Privileged EXEC Mode Password	27
	Step 2: Set Your VTY Login Password	27
	Securing VTY	28
	Step 1: Create and Name Your ACL	29
	Step 2: Apply Your ACL to VTY	29
	Configuring Your WAN Interface—Dynamic IP	30
	Step 1: Start Interface Configuration Mode	30
	Step 2: Add a Description to Your Interface	30
	Step 3: Configure Your WAN Interface to Use DHCP	31
	Step 4: Set the Duplex and Speed on Your Interface	31
	Step 5: Bring Up the Interface	31
	Step 6: Enable Domain Lookup	31

Configuring Your WAN Interface—Static IP	32
Step 1: Start Interface Configuration Mode	33
Step 2: Add a Description to Your Interface	33
Step 3: Assign an IP Address to Your Interface	33
Step 4: Set the Duplex and Speed on Your Interface	33
Step 5: Bring Up the Interface.	33
Step 6: Assign the Default Gateway.	33
Step 7: Enable Domain Lookup.	34
Configuring NAT on Your Router	34
Step 1: Create and Name an Extended ACL for NAT.	35
Step 2: Create an ACL Rule.	35
Step 3: Configure Inside Address Translation	35
Step 4: Apply NAT to Your Interfaces.	36
Securing Your Interfaces	36
Step 1: Disable IP Unreachable Messages	36
Step 2: Disable IP Redirects	37
Step 3: Disable Proxy ARP.	37
Creating a Basic Firewall.	37
Creating an Advanced Firewall.	38
Step 1: Create Application Rules	38
Step 2: Apply the Rules Outbound	38
Creating an ACL for Your WAN Interface.	39
Step 1: Allow Ping and Traceroute.	40
Step 2: Apply the ACL Inbound	40
Configuring a Basic DMZ.	40
Step 1: Remove the Existing IPFW-ACL.	41
Step 2: Create a New IPFW-ACL.	41
Step 3: Configure NAT to Forward Traffic to a LAN Host	41
Step 4: Apply the Inside Source Rule.	42
Saving Your Configuration.	43
Restoring the Default Configuration	44
Verifying Your Setup.	44
Check Your Interfaces	44
Check NAT	46
Check Your ACLs.	47
Check Your Firewall	48

Summary	48
Erase the Startup Configuration	49
Configure an IP Address on Your LAN Interface	49
Configure a DHCP Server	49
Configure Telnet on Your Router	50
Secure VTY with an ACL	50
Configure Your WAN Interface—Dynamic IP	51
Configure Your WAN Interface—Static IP	51
Secure Your Interfaces	52
Configure NAT on Your Router	52
Create an Advanced Firewall	53
Set Up a Basic DMZ	54
Save Your Configuration	54
Restore the Default Configuration	54
Verify Your Setup	55
 CHAPTER 3 Configuring DSL Using PPPoE	57
Introducing PPPoE	57
Overview of the Steps	58
Collecting Information from Your ISP	59
Enabling Virtual Private Dialup Networking	59
Preparing the Physical WAN Interface	60
Configuring the Virtual WAN Interface	61
Configuring NAT on the Virtual WAN Interface	64
Setting the Default Gateway	65
Adjusting the MSS on the LAN Interface	65
General Troubleshooting	66
Check That the DSL Circuit Has Been Activated	66
Check Your Username and Password and MTU	66
Verify That the Circuit Is Functional	67
Print a Copy of Your Router's Configuration	67
Use the IOS to Troubleshoot PPPoE	67
Using the Cisco Debugger	68
Enable Buffered Logging	68
Check for PPPoE Response	69
Debug the PPP	71
Stop Debugging and Logging	74
A Word About ISPs	75

Summary	76
What You Need from Your ISP	76
Enable VPDN and Create a Dial Group (If Necessary)	76
Prepare the Physical WAN Interface	77
Configure the Virtual WAN Interface	77
Configure NAT on the Virtual WAN Interface (Dialer 1)	77
Assign the Default Gateway to Use the Virtual WAN Interface	78
Adjust the MSS on the LAN Interface	78
Troubleshooting	78

CHAPTER 4 **Configuring a VPN Using IPSec** 81

Preparing Your Sites	81
Setting Up the VPN	82
Step 1: Create a VPN-Friendly ACL for NAT	83
Step 2: Define a VPN Routing Policy for Your WAN Interface	83
Step 3: Apply Your VPN Routing Policy to NAT	84
Step 4: Define a VPN Routing Policy for Your LAN Interface	84
Configuring IKE Phase 1	85
Step 1: Create a Key Exchange Policy	85
Step 2: Define the Encryption Type	86
Step 3: Define a Cryptographic Hash Function	86
Step 4: Define Your IKE Key Type	86
Step 5: Define Your IKE Key Size	86
Step 6: Create a Preshared Key	87
Configuring IPSec Phase 2	88
Step 1: Create a VPN-ACL	89
Step 2: Create a Transform Set	89
Step 3: Create a Crypto Map	89
Step 4: Set the VPN Peer	90
Step 5: Set the Transform Set	90
Step 6: Set the PFS Group	90
Step 7: Apply Your VPN ACL	90
Step 8: Apply the Crypto Map	90
Modifying Your IPFW-ACL	90
Verifying Your VPN Connection	92
Troubleshooting	94
General Network Settings	94
IKE Phase 1 Settings	94
IPSec Phase 2 Settings	95
When in Doubt, Print It Out	95

Summary	95
Set Up the VPN.	95
Branch Office VPN Configuration	98
Corporate Office VPN Configuration	100
Troubleshoot Your VPN.	102
 CHAPTER 5 Beyond the Basics	 105
Creating a Local User on the Router	105
Step 1: Create a User and Password	106
Step 2: Set the Login to Local	107
Configuring Secure Shell (SSH)	107
Step 1: Generate the RSA Keys	107
Step 2: Set the VTY Transport Input Type	108
Step 3: Use SSH to Log in to the Router	108
Recovering a Lost Password	109
Overview of the Process	109
Step 1: Bypass the IOS	110
Step 2: Modify the Configuration Register	110
Step 3: Copy the Configuration and Reset Passwords	111
Step 4: Reset the Configuration Register	112
Upgrading the IOS	113
Step 1: Display the Contents of Flash Memory	113
Step 2: Back Up the Existing IOS Image File	115
Step 3: Delete the Old IOS Image	117
Step 4: Install the New IOS Image	118
Step 5: Boot the New Image	120
Backing Up Your Configuration	121
Method 1: Back Up to Flash Memory	121
Method 2: Back Up to a TFTP Server	122
Method 3: Back Up to an FTP Server	123
Tuning Your ACLs for Performance	124
Step 1: Display ACL Rule Matches	124
Step 2: Reorder the ACL Rules	125
Step 3: Apply the Established Rule	126
Protecting Your Passwords	126
Disabling Show and Tell	127

Safeguarding Your E-mail Server	127
Step 1: Name the EIE Firewall	128
Step 2: Define the Protocols	128
Step 3: Apply the Firewall	129
Configuring a Logging Host for Intrusion Detection	129
Step 1: Perform Basic KIWI Set Up	130
Step 2: Configure E-mail Alarms	130
Step 3: Set the Message Threshold	131
Configuring Logging on Your Router	133
Step 1: View Your Trap Levels	133
Step 2: Change the Log Level	134
Step 3: Timestamp Your Logs	134
Step 4: Define Your Logging Host	135
Defining a Login Banner	135
Summary	136
Create a Local User on the Router	136
Configure Secure Shell (SSH)	136
Recover a Lost Password	137
Back Up the IOS	137
Upgrade the IOS	138
Back Up Your Configuration	138
Tune Your ACLs for Performance	139
Protect Your Passwords	139
Disable Show and Tell	140
Safeguard Your E-mail Server	140
Set Up an Intrusion Detection System	140
Define a Login Banner	141

CHAPTER 6 Understanding Binary and Subnetting	143
Decimal—Base 10	144
Binary—Base 2	145
Subnet Masks	146
Dividing Your Network	147
Method 1: Keeping the Same Subnet Mask	147
Method 2: Subnetting a Network	147
Determining How the Bits Are Used	148
Determining the Number of Subnets Available	148
Determining the Network Numbers and Number of Hosts	149
More Examples	151

Summary	154
Decimal—Base 10	155
Binary—Base 2	155
Subnet Mask	155
Dividing Your Network	155
Determining the Number of Subnets	155
Determining the Network Number and Number of Hosts	156
Quiz Answers	156
Binary Quiz Answers	156
Subnetting Quiz Answers	156

CHAPTER 7	Routing—What Routers Do Best	157
	Routing Defined	157
	Routing vs. Routed Protocols	158
	Routing Information Protocol (RIP)	158
	RIP Basics	158
	Configuring RIP on a Router	158
	Step 1: Enable RIP	160
	Step 2: Advertise Your Networks	160
	Configuring RIP on a Neighbor Router	160
	Step 1: Enable RIP	161
	Step 2: Advertise Your Networks	161
	Step 3: Configure a Passive Interface	161
	Verifying RIP Routing	161
	Use Show IP Protocols	161
	Use Show IP Route	162
	Setting Up a True DMZ	163
	The Bastion Host	164
	Configuring Your Gateway Router	165
	Configuring Your Interior Router	166
	A Note on VPNs and DMZs	167
	Summary	168
	Configure RIP	169
	Configure RIP on a Neighbor Router	169
	Verify RIP Routing	170
	Set Up a True DMZ	171
	VPN Configuration	171

CHAPTER 8	Understanding Variable Length Subnet Mask Networking	173
	Getting Started	173
	Planning a VLSM Network	175
	Route Summarization (Supernetting)	178
	Summary	180
	Planning a VLSM Network	180
	Route Summarization	181
APPENDIX A	Sample Configuration for a Cable Modem	183
	Standard Setup	184
	LAN Interface	184
	WAN Interface	184
	Router Passwords	185
	NAT Setup	185
	CBAC Firewall	186
	DHCP Server	186
	IPFW Access List	187
	VTY Access List	187
	Configure SSH (Version 2)	188
	Encrypt All Router Passwords	188
	Save the Configuration	188
APPENDIX B	Sample Configuration for DSL and PPPoE	189
	Standard Setup	190
	LAN Interface	190
	ENABLE PPPoE	191
	WAN Interface (Physical)	191
	WAN Interface (Virtual Dialer)	191
	Router Passwords	192
	NAT Setup	192
	CBAC Firewall	193
	DHCP Server	193
	IPFW Access List	194
	VTY Access List	194
	Configure SSH (Version 2)	195
	Encrypt All Router Passwords	195
	Save the Configuration	195

■ APPENDIX C	Sample Configuration IPsec VPN Over DSL	197
	Standard Setup	198
	LAN Interface	198
	Enable PPPoE	199
	WAN Interface (Physical)	199
	WAN Interface (Virtual Dialer)	199
	Router Passwords	200
	NAT Setup	200
	CBAC Firewall	201
	VPN Cryptographic Settings	202
	DHCP Server	203
	IPFW Access List	203
	VTY Access List	204
	Configure SSH (Version 2)	204
	Encrypt All Router Passwords	205
	Save the Configuration	205
 ■ APPENDIX D	 CCNA CLI Command Reference	 207
	Cisco Router Commands	207
	Access Control List (ACL)	208
	Backup and Restore the IOS	210
	Cisco Discovery Protocol (CDP)	210
	Command History	211
	Configuration Register Commands	211
	Password Recovery	212
	Console Messages	213
	Date and Time	213
	DHCP Configuration	213
	DNS Lookup	213
	Frame-Relay	214
	Hostname and Message of the Day (MOTD)	214
	Interface—Configuration	214
	Interface—Verifying TCP/IP Configurations	215
	Network Address Translation (NAT)	216
	Password—Encryption	219
	Password—Setting	219
	PPP Configuration	219
	Routing—Default Routes	220
	Routing—EIGRP	221

Routing—IGRP	221
Routing—OSPF	221
Routing—RIP	222
Routing—Static Routes	223
Secure Shell (SSH)	223
Startup-Config and Running-Config Files	223
Telnet	224
VTY ACL for Telnet and SSH	224
Cisco Catalyst Switch Commands	225
Hostnames	225
Interface Configuration	225
Passwords	226
Port Security	226
Saving and Deleting Configurations	226
VLAN—Configuration	227
VLAN—Inter-VLAN Routing Example	228
VLAN—VTP Domain Configuration	230
APPENDIX E ACL and Firewall Names Used in This Book	231
ACL Names	231
CBAC Firewall Names	232
DHCP Pool Name	232
Routing Policy Names	232
INDEX	233

About the Author

Having been professionally involved in computer networking for over 20 years, **JASON NEUMANN** has worked with Cisco routers for more than 10 of those years. Jason is the owner of LAN Technologies LLC, a small networking company located in Anchorage, Alaska, that provides local and wide-area network solutions and support to small businesses using high-end operating systems including the Cisco IOS, Microsoft, Linux, and BSD UNIX. He holds many credentials from industry leaders including Cisco, Microsoft, and Novell.

About the Technical Reviewers

A telecommunications engineer and consultant, **DEAN OLSEN** has over 20 years of experience in IP networking and services. He specializes in IP-based carrier technologies such as MPLS, SONET, Carrier Ethernet, and GSM wireless data networks. Throughout his career Dean has been responsible for designing, implementing, and troubleshooting a variety of networks from simple point-to-point transport to complex multipoint converged service delivery architectures. Currently Dean is working with a regional carrier on the design and implementation of a large-scale multivendor GSM-based converged network supporting SS7 Sigtran, VoIP, and MMS technologies.

SEBASTIEN MICHELET (CCIE #16877) is a senior network engineer in the R&D department at ADP (Automatic Data Processing). He designs and installs Cisco IP telephony solutions for the car dealership market. Before diving into the VoIP world, he was a networking engineer responsible for maintaining, securing, and monitoring large networks of firewalls and routers. His career in Cisco networking spans 12 years. He has an MS in mechanical engineering from the University of Poitiers, France.

Acknowledgments

I wish to extend my sincere gratitude to Lois Weber for her proofreading skills and keen eye for detail; to my daughter, Terra Vleeshouwer-Neumann, for her impeccable knowledge of grammar; to my son, Gabe, for allowing me to cut into our “guy time”; and to my wife, Sharon, for helping me with pretty much every aspect of this book!

Introduction

“The creation of this book, like many things in life, was a complete accident.”

This book is intended for the average network administrators or IT professionals who manage small networks and are currently using, or want to use Cisco IOS-based routers in their networks. After all, why should Cisco routers be reserved for elite Cisco gurus when all you need to know are a few simple concepts and commands? This book is about a Cisco CLI for the regular guy or gal. After reading this book, you'll no longer have to use cheap consumer-grade routers on your small business network. You, too, can have all the reliability and advanced functionality that the Cisco IOS offers.

In my experience, the best way to learn this material is through hands-on experience. The more the better! Therefore, you may want to have a spare Cisco router to work with. You can use the book without one, but it really helps to have an actual router on hand to work through the material. The Cisco 831 and 851 routers will be used throughout my examples. If you don't have a router, you can easily find an older 800 series router on eBay or some other used computer site. In secondary markets like eBay, a Cisco 831 or SOHO91 series router is inexpensive, easy to come by, and will work well with the material. Keep in mind that you will need a router with at least *64 MB of memory* to configure DSL using PPPoE. Also, I assume that the router has an IOS version of 12.4 or greater.

Each chapter of this book has specific configuration examples, in the form of command listings, showing how to configure the features of your Cisco router. Chapters 1 through 4 provide tutorial-based examples of how to configure your router for different broadband technologies, including cable modems, DSL, and setting up VPNs using IPSec. Chapter 5 explains some of the more advanced—but not too advanced—features of the IOS.

Chapter 6 provides IP networking fundamentals that can be very useful to network administrators, IT professionals, or anyone who is preparing to become Cisco CCNA certified. Chapter 7 provides information about setting up an advanced IP network using multiple Cisco routers and the Routing Information Protocol (RIP) to configure a true DMZ on a separate private network. Chapter 8 is about VLSM networking, which is a necessary concept to understand for CCNA certification.

At the end of each chapter is a summary that can be used for quick reference once you're familiar with Cisco concepts and commands, or it can be used right away to help configure your router if you already have some Cisco networking experience.

Finally, there are appendixes at the end of the book that provide keystroke-for-keystroke commands used to configure a router for various scenarios. Although you can use this information exclusively to configure your router, I recommend you first read Chapters 1 through 6 to get a feel for the Cisco IOS and some networking concepts. Appendix D has an extensive IOS command reference guide geared toward CCNA exam preparation.



Getting to Know Your Router

In this book, I assume that you have one of Cisco's 800 Series or SOHO Series broadband routers on hand. You can use other Cisco models, but an 800 series running Cisco Internetworking Operating System (IOS) version 12.4 will be referenced in my examples. The 800 series routers are great, low-cost devices that have all the functionality small businesses need, with the added benefits that Cisco's IOS has to offer.

The Cisco IOS has a Command Line Interface (CLI) that allows you to type in special commands to configure your router. The CLI is a powerful tool that gives you full control of your router's features and is the key to understanding any Cisco IOS router or switch. As you work your way through this book, I will be introducing all of the CLI commands that will allow you to set up and configure your router for a small business. You'll be a Cisco pro in no time at all!

Understanding Your Ports

Cisco routers are essentially small computers. As such, they have all the basic software and hardware components of a PC. In addition to the IOS, they have hardware components such as a processor and memory. They also have several ports (interfaces) that allow you to connect other hardware components, such as workstations and switches as well as cable modems and DSL modems. Before we get started, it's important that you know a little bit about your equipment, so take it out of the box and have a look.

The Console Port

On the back of your router, you will find a port labeled "Console." This is sometimes referred to as the *management port*. It has an RJ45 connector that looks like a regular Ethernet port, but it's not. The console port is used in conjunction with a *console cable*, and allows you to configure your router from a PC. The console cable, which is usually blue, should have arrived with your router. It has an RJ45 connector on one end—which looks like a fat phone connector—and a DB9 female serial port connector on the other. The serial port end plugs into your PC's serial port, and the RJ45 plugs into the console

port on the back of your router. You'll learn more about the console port in the section, "Connecting to Your Router," later in this chapter.

LAN Ethernet Ports (E0 or VLAN1)

There are four Ethernet switch ports on the back of your router. On Cisco SOHO91 and 831 series routers, these ports are collectively labeled "Ethernet 10/100 BaseT Computers (E0)." On the newer 850 and 870 series routers they're labeled "LAN FE0, FE1, FE2, FE3." These ports are used by your workstations, or if you have more than four PC's, as an uplink to another Ethernet switch. What's important to note here is the IOS refers to all four ports as *e0*, or *vlan1*, depending on the model of your Cisco router. Later, when you're configuring your router with the CLI, you will be referring to those ports as either *interface e0* or *interface vlan1*.

WAN Ethernet Port (E1 or FA4)

There's one more port on the back of your router, which on Cisco SOHO91 or 831 series routers is labeled "Ethernet 10 BaseT Internet (E1)." On the newer 850 or 870 series routers, it's labeled "WAN FE4." This is the port where you will plug in your broadband device, which is usually a DSL modem or cable modem. It's important to remember that it is referred to by the Cisco IOS as either *interface e1* (on older models), or *interface fa4* (on the newer 850 and 870 series).

Note It's odd, but Cisco chose to use different interface label names (on the back of the routers) than the names used to configure the interfaces themselves. For example, the label for the WAN port is FE4, but when you configure it using the IOS, it's referenced as FA4.

There are a lot of Cisco router models to choose from, and many of them use different names for their LAN and WAN interfaces. To simplify this introduction, I will be using early 800 series interface naming. The names are: *e0* for the LAN interfaces and *e1* for the WAN interface. In later chapters, I will introduce the 851 series router and use *vlan1* for the LAN interfaces and *fa4* for the WAN interface.

Connecting to Your Router

Cisco provides several methods to connect to and manage their routers. One method is the Security Device Manager (SDM). The SDM is a web-based interface that is accessed from a Java-enabled web browser. When you configure a router from the SDM, you fill in

web-based forms and access options from drop-down menus to configure the features of the router. When you're finished providing the configuration information, you save the data, which is converted to Cisco IOS commands that are then delivered to the IOS.

This may sound convenient, and when it works it is, but the reality is that the SDM interface is cumbersome and very unreliable. Often it will hang during the delivery of the IOS commands, delivering only some of the configuration to the router or none at all. When that happens, you need to start over. Also, the SDM only allows you to configure basic features of the Cisco IOS, even though the IOS version of the router supports a much more advanced feature set. Because of the problematic nature and limitations of the SDM, I won't be discussing it in this book, but it is something you should be aware of.

You can also access the router from a PC using a telnet or SSH application via one of the router's Ethernet ports. Because telnet and SSH are TCP/IP applications, the router must have an IP address and other configuration options set before using this connection method (I will be discussing this in Chapter 2). When you use telnet or SSH to access the router, you need to configure it using the CLI.

Another method used to connect to and configure your router is the console port on the back of the router. The console port allows you to configure your router when it has no IP address or other configuration information. Using a console cable that plugs into a PC, you can gain access to the Cisco CLI and issue IOS commands to configure the router.

Attach the Console Cable

The first time you connect to your router, you'll want to use the console port. The console port allows you to log in to your router via a PC *before* you have set up the router and assigned it an IP address. After you complete the router's basic IP setup, you can use telnet or the SSH application to connect to it from any PC on your network.

Locate the console cable provided with your router, plug the DB9 serial port into your PC, and plug the other end into the console port on the back of your router. If you have a PC or laptop that does not have a serial port, then you will need to purchase a USB-to-Serial adapter. These are inexpensive and can be found at any computer supply store.

Configure Hyper Terminal

Before you can log in, you need a terminal emulation program to allow you to interact with the router. All computers running Windows come with Hyper Terminal, which works nicely for our purposes. If you are using a Unix PC, then you may want to look into *minicom*, *kermit*, or some other UNIX terminal application. On a Windows workstation, the Hyper Terminal application is located in the following directory:

All Programs/ Accessories/ Communications/ Hyper Terminal.

Start Hyper Terminal, name your connection, and then click OK, as shown in Figure 1-1. I've named my connection "Cisco Console," but any name will do.

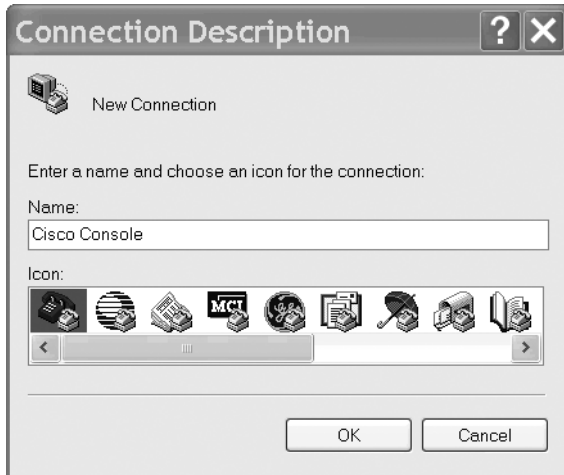


Figure 1-1. *Creating a connection description for Hyper Terminal*

Now, select the COM port (serial communication port) that you're using with the console cable on your PC, and then click OK. This is usually COM1, but it could be COM2, COM3, or even COM4. This will depend on what hardware is installed in your PC. Figure 1-2 shows COM1 as the selected port.



Figure 1-2. *Selecting COM1 as the targeted serial port*

Next, define the properties of the COM port, then click OK. The default for all Cisco routers is 9600 bits per second, 8 data bits, no parity, and 1 stop bit, as shown in Figure 1-3.

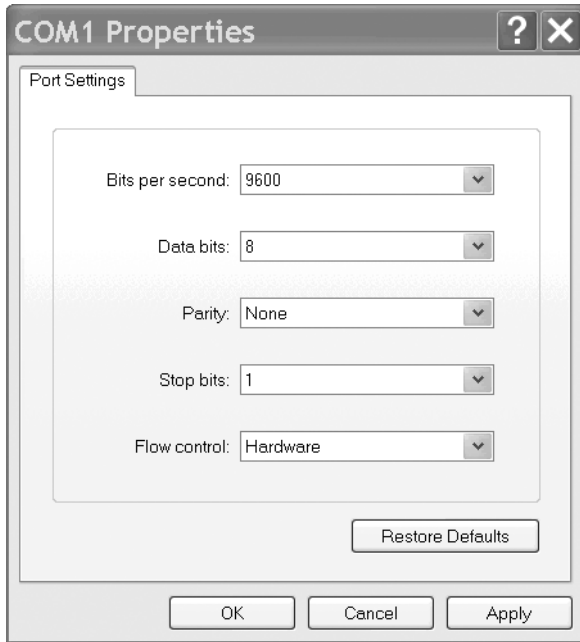


Figure 1-3. *Defining the serial port properties*

Note Cisco Routers support ANSI (American National Standards Institute) terminal emulation. Hyper Terminal also supports ANSI and defaults to auto detect the terminal emulation type, which works very well with Cisco routers. If you use some other terminal emulation software, you may need to manually set the terminal emulation type. If so, be sure to set it to ANSI to ensure that the terminal software works correctly with your router.

Power Up the Router

After you have your console cable plugged in and have started Hyper Terminal, flip the router's power switch to the *ON* position. If all goes well, you should see the Cisco bootstrap message. If not, you may need to check your Hyper Terminal settings or cable.

At startup, a lot of information is displayed. Notice in the following sample output that the router in question has 64 MB (65536 KB) of main memory. Your router also has system flash memory, which stores a compressed image of the IOS, and web flash memory, which stores other configuration files. During startup, the image file in system flash memory is decompressed and loaded into main memory. While decompressing, pound signs march across the screen to indicate progress.

```

System Bootstrap, Version 12.2(8r)YN, RELEASE SOFTWARE (fc1)
TAC Support: http://www.cisco.com/tac
Copyright (c) 2002 by Cisco Systems, Inc.
C800/SOH0 series (Board ID: 28-130) platform with 65536 Kbytes of main memory

program load complete, entry point: 0x80013000, size: 0x6897e8
Self decompressing the image :
#####
#####[OK]

```

Near the end of startup, other information about your hardware is displayed. You can see from the following output that the router being used is a Cisco SOHO91 router with 2 Ethernet interfaces (e0 and e1 mentioned earlier). The system flash memory (where the IOS is stored) is 8 MB, and the web flash memory is 2 MB.

```

Cisco SOHO91 (MPC857DSL) processor (revision 0x300) with 5893kK/65536K bytes of
memory.
Processor board ID AMB080903VX (1784032485), with hardware revision 0000
CPU rev number 7
2 Ethernet interfaces
128K bytes of NVRAM.
8192K bytes of processor board System flash (Read/Write)
2048K bytes of processor board Web flash (Read/Write)

Press RETURN to get started!

```

At this point, depending on how your router is configured, other messages can appear. Pressing the ENTER key should give you the Cisco CLI prompt.

Welcome to the Command Line

The Command Line Interface (CLI) is where you configure your router. The CLI has several prompts, with each one representing a different mode (e.g., *user EXEC mode* and *privileged EXEC mode*). Each mode offers different configuration commands and tools that allow you to manage your router. Before you start configuring your router, I want to introduce a few simple commands to familiarize you with the user interface.

Your First CLI Commands

The first prompt you are presented with is the user EXEC mode prompt. You know you are in user EXEC mode when you see the “>” symbol after the name of your router, which is “Router” by default. There are very few things you can do in user EXEC mode (you cannot configure your router in this mode). Let’s try a few commands to get a feel for the CLI. In all of the CLI examples throughout this book, you’ll type the bolded information after the prompt on the command line, then press the ENTER key. To begin, try the following `show version` command:

```
Router> show version
```

Depending on the model of your router and the IOS version that it is running, you will see output similar to this:

```
Cisco IOS Software, SOHO91 Software (SOHO91-K90Y6-M), Version 12.4(5b),  
RELEASE SOFTWARE (fc2)  
Technical Support: http://www.cisco.com/techsupport  
Copyright (c) 1986-2006 by Cisco Systems, Inc.  
Compiled Wed 19-Apr-06 08:12 by ssearch  
  
ROM: System Bootstrap, Version 12.2(11r)YV3, RELEASE SOFTWARE (fc2)  
  
Router uptime is 1 day, 12 hours, 35 minutes  
System returned to ROM by power-on  
System image file is "flash:soho91-k9oy6-mz.124-5b.bin"
```

The `show version` command tells you three important things about your router. The first is the IOS version (12.4(5b) in this case). The second is the router uptime (how long the router has been running). And third, it tells you the name of the compressed image file in system flash memory, which is “soho91-k9oy6-mz.124-5b.bin”.

Another way to find out about the compressed image file is to issue the `show flash` command. This command is a good way to see how much system flash memory the image is using:

```
Router> show flash
```

As you can see from the following output, the `show flash` command displays a directory of the files in system flash memory and how much space they are using (6.8 MB in this case).

System flash directory:

File Length Name/status

1 6855008 soho91-k9oy6-mz.124-5b.bin

[6855072 bytes used, 1271388 available, 8126460 total]

8192K bytes of processor board System flash (Read/Write)

The reason this information is important is because someday you may want to upgrade your IOS version, and you need to be sure you have enough memory to hold the new image file. IOS upgrades are a beneficial Cisco service. Sometimes they fix bugs, and other times they add features. In any case, every new version will be larger than the last. I'll discuss upgrading your IOS in Chapter 5.

Another handy command is `show ip interface brief`. This command tells you basic information about the status of your interfaces and displays their IP addresses.

Router> `show ip interface brief`

Interface	IP-Address	OK?	Method	Status	Protocol
Ethernet0	10.10.10.1	YES	NVRAM	up	up
Ethernet1	66.238.5.254	YES	NVRAM	up	up

This output contains a lot of valuable information. You know which IP addresses are assigned to your router. The IP address of the LAN switch ports (Ethernet0) is 10.10.10.1, and the IP address of the Internet port (Ethernet1), sometimes referred to as the WAN port, is 66.238.5.254.

“Status” shows whether an interface has been shut down by an administrator. “Protocol” shows whether a cable may be unplugged.

Let's look at another example of the command. As you can see from the following output, this router has a manually shut down interface (Ethernet0):

Router> `show ip interface brief`

Interface	IP-Address	OK?	Method	Status	Protocol
Ethernet0	10.10.10.1	YES	NVRAM	manual administratively down	down
Ethernet1	66.238.5.254	YES	NVRAM	up	down

First, let's look at interface Ethernet0. It has an IP address (that's good), but the status displays the message "manual administratively down" and the protocol is also "down." This tells us that the interface was manually shut down by an administrator. Next, look at interface Ethernet1. It has an IP address and its status is "up," but the protocol is "down." If the Protocol is down, it may indicate that the Ethernet cable is unplugged or there is something wrong with the cable. Check the cable.

If you want to exit user EXEC mode and log out of the router, simply type **exit** and press the ENTER key:

```
Router> exit
```

Now that you're familiar with how to connect to your router and issue a few simple commands, you're going to enable privileged EXEC mode and learn a few IOS configuration commands.

Turn On Privileged EXEC Mode

Privileged EXEC mode, sometimes referred to as *enable mode*, gives you a lot more commands than user EXEC mode and allows you to begin configuring your router. You know you are in privileged EXEC mode when you see the "#" sign after the name of your router. Let's get started; type **enable**, then press the ENTER key. Notice how the prompt changes in the following example:

```
Router> enable
Router#
```

Set the Date and Time

Now that you're in privileged EXEC mode, let's start by setting the date and time on your router. You use the **clock set** command and enter the time in a 24 hour format. Therefore, 1:40 PM is entered as 13:40:00. Also, note the European date order format (Day Month Year). To display the time on your router, use the **show clock** command. Here is an example:

```
Router# clock set 13:40:00 13 April 2007
Router# show clock
```

```
13:40:46.887 AKST Fri Apr 13 2007
```

Get Help

In the Cisco IOS, help is only a question mark away. To see a list of commands, type a question mark, then press the ENTER key. To see a list of command parameters, type the command followed by a question mark, and then press the ENTER key. The following example shows how you can get help on the `clock set` command:

```
Router# clock set ?
```

```
hh:mm:ss  Current Time
```

In the next example, a question mark has been added after the time parameter. Notice when you use a question mark after a parameter, the IOS displays subsequent parameters that can be used for that command. In this example, the next set of parameters is the day of the month and the month of the year.

```
Router# clock set 13:40:00 ?
```

```
<1-31>  Day of the month  
MONTH   Month of the year
```

When configuring your router, you can use the question mark at any time to access help with any IOS command or parameter. That ability is an important feature of the Cisco IOS you will quickly become familiar with.

Using Global Configuration Mode

So far you have learned about user EXEC mode and privileged EXEC mode. The next, and most powerful mode you can use in the Cisco IOS, is *global configuration mode* (a sub-mode of privileged EXEC mode). Global configuration mode allows you to configure pretty much every aspect of your router. While in global configuration mode, you can access even more sub-modes, such as interface configuration mode or VTY configuration mode (I'll be discussing these modes in Chapter 2). What's important to note here is that when you configure your router, you use commands to escalate your privileges on the router, which in turn allows you to access and configure different features of the IOS.

The command to enter global configuration mode is `configure terminal`. You know that you're in global configuration mode when you see the `(config)#` prompt.

Set Your Router's Hostname

Assigning your router a hostname will allow you to easily identify the router on which you're working. When a hostname has been assigned, the name of the router will be displayed on the screen as part of the prompt during the configuration process. If you're an IT consultant who works with multiple clients and routers, the hostname will help you keep track of which client and router you're configuring.

Let's go into global configuration mode and set the hostname for your router:

```
Router> enable
Router# configure terminal
Router(config)# hostname lab-r1
```

And notice the new prompt:

```
lab-r1(config)#
```

The prompt has changed to reflect the newly assigned hostname, "lab-r1."

Set the Privileged EXEC Mode Password

Because privileged EXEC mode is so powerful, it's best to secure it with a password. The Cisco IOS allows you to set several layers of passwords, but this is the big one, and I recommend you set it *before* you put your router online. The following code shows you how to set the privileged EXEC mode password to "cisco":

```
lab-r1> en
lab-r1# config t
lab-r1(config)# enable secret cisco
```

In this example, you might assume I made a typo in the second line, but I didn't. Anyone who's been around the Cisco IOS for any length of time uses abbreviated commands. All IOS commands can be abbreviated, which is great because no one wants to type in "configure terminal" when they can simply type "config t".

Now that you've set your privileged EXEC mode password, try it out. You'll need to log out of privileged EXEC mode and go back to user EXEC mode using the exit command:

```
lab-r1(config)# exit
lab-r1# exit
lab-r1> enable
```

Password:

Did you get a password prompt? Go ahead and type in the password **cisco** to enter privileged EXEC mode again.

Display and Save Your Configuration

The `show running-configuration` command, (abbreviated `sh run`) and the `show startup-configuration` command (abbreviated `sh start`) are two useful commands that are used often when setting up and managing your router. When your router is turned on, the IOS looks for a startup configuration file. If it finds one, it loads the configuration into your router's main memory (running configuration). The running configuration is the active configuration on your router.

The `sh run` command will display all the details of your router's running configuration in main memory. This is where you look to see what features are enabled and how they are configured. Here is how to issue the command:

```
lab-r1# sh run
```

Following this paragraph is a partial example of output from the `show run` command. Use the spacebar to page through the configuration. Try this command now on your router but don't worry if you don't understand the details of the output. Some of the information will be familiar to you, such as the `hostname` command, while other configuration information will be default settings that may seem cryptic at this time. Not to worry—you'll become familiar with the details of your configuration as you work through this book.

```
Current configuration : 11075 bytes
!
version 12.4
no service pad
service timestamps debug datetime msec
service timestamps log datetime localtime
!
!
hostname lab-r1
!
boot-start-marker
boot-end-marker
.
.
.
```

The `show startup-configuration` command (abbreviated `sh start`) will display the contents of your router's startup configuration.

After you power up your router, the IOS loads into memory and is executed. The first thing the IOS does is look for a startup configuration. If no startup configuration is found, the IOS will start a system configuration dialog that will guide you through basic router setup. However, if a startup configuration is found, it is read and loaded into the router's main memory. Here's how to display the contents of the startup configuration on your router:

```
lab-r1# sh start
```

Depending on your router's current configuration, the output of the `sh start` command appears similar to the output of the `sh run` command in the previous example.

The last command I want to introduce you to in this chapter is `copy running-configuration startup-configuration` (abbreviated `copy run start`). It allows you to save your running configuration file so that it loads the next time you start your router. For example,

```
lab-r1# copy run start
```

```
Destination filename [startup-config]? {press ENTER}
```

```
Building configuration...  
[OK]
```

```
lab-r1#
```

Great! Now you know the basics of the Cisco IOS. In Chapter 2, you'll start learning how to setup your router for a small business, so make sure you are comfortable with logging in, changing modes, and entering a few commands before moving on.

Summary

By now you should have a feel for how the CLI works. In this chapter, you've learned about the ports on the back of the router, how to connect to the console port, and how to login using Hyper Terminal. You have also learned about three IOS modes: user EXEC mode, privileged EXEC mode, and configuration mode. You've learned how to get help by typing a question mark, and how to enter a few commands into the router. You have also learned that Cisco commands can be abbreviated.

Here's a quick recap of Chapter 1.

Ports

Your router has three types of ports:

Console Port: Located on the rear of the router, the console port is accessed using a console cable and terminal emulation software, such as Hyper Terminal or minicom. The COM settings are 9600, 8, n, 1 (9600 baud, 8 data bits, no parity and, 1 stop bit) and ANSI terminal emulation.

LAN Ethernet Port (E0): The interface name for the four 10/100 BaseT LAN switch ports on the rear of a Cisco SOHO91 or 831 series router. These ports are used for workstations.

WAN Ethernet Port (E1): The interface name for the 10 BaseT Internet WAN port on the rear of a Cisco SOHO91 or 831 series router. This interface is used by a DSL or cable modem.

Alternatively, the Ethernet port names for Cisco's newer 850 and 870 series routers are

LAN FastEthernet Port (VLAN1): The interface name for the four 10/100 BaseT switch ports on the rear of Cisco 850 and 870 series routers (FE0, FE1, FE2, FE3). These ports are used for workstations.

WAN FastEthernet Port (FA4): The interface name for the 10/100 BaseT Internet WAN port on the rear of Cisco 850 or 870 series routers. This port is used by a DSL modem or a cable modem.

To access your router through the console port, perform the following steps:

1. Plug the console cable into your computer's DB9 serial port and the RJ45 end into the router's console port.
2. Start Hyper Terminal or minicom and set the communication parameters to 9600 baud (bits per second), 8 data bits, no parity, and 1 stop bit.
3. Power on your router and wait for the IOS to load.
4. Press the ENTER key to get a command prompt (Router>).

User EXEC Mode Commands

User EXEC mode has very *limited* functionality. You cannot configure your router in this mode. You know you are in user EXEC mode when you see the > symbol after the name of your router. Some useful commands in this mode follow:

`show version`—Displays basic information about your IOS version and router hardware.

`show flash`—Displays the name and size of your compressed IOS image file.

`show ip interface brief`—Displays basic information about the status of your interfaces and displays their IP addresses. Router and protocol status information that you might see includes the following:

- Router status of “manual administratively down” indicates the interface has been shut down by an administrator.
- Router status of “up,” but Protocol “down” indicates a cable problem. Check if the cable is unplugged.

`show clock`—Displays the date and time on your router.

`exit`—Exits the current mode returning you to the previous mode.

Privileged EXEC Mode Commands

The privileged EXEC mode allows you to configure your router. Due to its expanded capabilities, this mode should be password protected. You know you are in privileged EXEC mode when you see the # sign after the name of your router. Some useful commands related to or in that mode are:

`enable`—Enters privileged EXEC mode from user EXEC mode.

`clock set 13:40:00 13 April 2007`—Allows you to set the date and time on your router.

`show clock`—Displays the date and time on your router.

Global Configuration Mode Commands

Global configuration mode is the most powerful mode available to you on a Cisco router. From this mode you can configure most every aspect of your router. You know you are in global configuration mode when you see (config)# in the prompt after the name of your router. Some global configuration mode commands related to, or used in that mode are:

`configure terminal`—Enters global configuration mode, which is a sub-mode of privileged EXEC mode.

`hostname lab-r1`—Allows you to set the hostname on your router. In this example it's set to lab-r1.

`enable secret cisco`—Allows you to set the privileged EXEC mode password on your router. In this example, it has been set to cisco.

Display and Save Your Configuration

Issue the following commands from privileged EXEC mode to display and save your router's configuration:

`sh run`—Displays all the details of your router's running, or active, configuration.

`sh start`—Displays all the details of the startup configuration that will be loaded at startup.

`copy run start`—Saves your running configuration file to a startup configuration file, which will be loaded the next time you issue the `reload` command or power-cycle your router.



Configuring Your Router

If you don't know NAT from DHCP, or a DMZ from a 1959 Buick, don't panic! I'm about to describe what all of these technologies are, why they're useful, and how to configure them on your router.

In this chapter, you'll learn how to use the IOS to configure your router to be an Internet broadband router and firewall. We will use a Cisco 851 router that uses the interface configuration name `vlan1` for the LAN ports and `fa4` for the WAN port. You will be configuring the router for a nonauthenticated Internet connection, such as a cable modem. In Chapter 3, I'll explain how to set up an authenticated Point-to-Point Protocol over Ethernet (PPPoE) connection using Digital Subscriber Lines (DSL). The topics I am now going to cover include how to configure an interface with an IP address, how to manage all of your network IP addresses with Dynamic Host Control Protocol (DHCP), and how to set up Network Address Translation (NAT) and configure a demilitarized zone (DMZ) with an access control list (ACL) for common services like e-mail and web hosting.

Erasing the Startup Configuration

When a router is shipped from Cisco, it has a default configuration pre-installed on the device. The default configuration has many preset options, such as an IP address assigned to the LAN interface, and other IOS features that allow a user to start configuring and managing the router using the SDM and a web browser. These preconfigured settings may conflict with your site's existing network configuration and may make the initial configuration process using the CLI somewhat cumbersome. For example, if you don't want one of the preconfigured options enabled on the router, you would need to change the configuration to disable that feature, which could be quite a hassle. Therefore, it's a good idea to erase the factory's default startup configuration before you begin configuring your router for the first time. This allows you to start with a clean slate and configure the router with all the features you need and leave off any that you don't want.

Caution You'll begin by erasing your router's current configuration! You may want to save your existing configuration before you begin, in case you need to restore it later. See Chapter 5 to learn how to backup and restore files to a flash drive or a TFTP server.

I strongly believe that hands-on experience is the best way to learn this material. Therefore, you really should work with a router. If you are currently using your Cisco router in a production environment, then you may want to **BACKUP** the existing configuration to a server before you begin, or acquire another router to work with during this tutorial. In either case, you need to erase the router's existing configuration.

Connect your console cable to your router as described in Chapter 1, start HyperTerminal, and log in to your router. Enable privileged EXEC mode using the enable command, and erase your startup configuration file with the erase start command. Then, reset the power on your router to start with a clean slate, as in the following example:

```
Router> ena
Router# erase start
```

```
Erasing the nvram file system will remove all configuration files!
Continue? [confirm]
```

Now press the ENTER key to confirm erasing the configuration. This command is irreversible so be sure to use caution here. If you change your mind and do not want to erase the configuration, press the Ctrl+C key now. If you press the ENTER key, you should see the following message:

```
Erase of nvram: complete
```

Now, reset the power on your router or type **reload** at the console prompt. After the router restarts, you'll see this dialog message:

```
--- System Configuration Dialog ---
```

```
Would you like to enter the initial configuration dialog [yes/no]
```

Since you're learning how to use the CLI to configure a Cisco router, you'll want to answer "no" to the dialog and press the ENTER key, which will provide you with a command prompt. If you answer "yes" to this question, the router will begin a dialog asking you a series of questions that will provide the router with a basic configuration, such as the router's hostname, privileged EXEC mode password, and the IP address of the LAN interface.

Learning Some CLI Tips and Tricks

Before you begin configuring your router, I want to introduce a few tips and tricks to help make the experience more pleasant.

Use Keyboard Shortcuts

There are some keyboard shortcuts that I find very useful when using the Cisco CLI. The up-arrow key, in particular, comes in handy. If you press the up-arrow key repeatedly, the console will cycle through and display the "command history," meaning the commands you recently typed in at the CLI. Table 2-1 shows some useful keyboard shortcuts (the first five are the most useful).

Table 2-1. *Keyboard Shortcuts*

Up arrow	Displays your previous command (history)
TAB key	Completes a partially typed CLI command
Ctrl+A	Places the cursor at the beginning of a line
Ctrl+E	Places the cursor at the end of a line
Ctrl+Z	Takes you back to privileged EXEC Mode
Ctrl+F	Moves the cursor forward one character
Ctrl+B	Moves the cursor backward one character
Esc+F	Moves the cursor forward one word
Esc+B	Moves the cursor backward one word at a time
Ctrl+R	Redisplays the current command line
Ctrl+U	Erases (undoes) an entire line
Ctrl+W	Erases a word (behind the cursor)

Suppress Console Messages

When you manage your router from the console port, it displays a lot of chatty and often distracting messages. You can suppress the output of those messages with the global

configuration mode command `no logging console`. I highly recommend using this command while configuring your router. You can disable console logging using the following commands:

```
Router> en
Router# config t
Router(config)# no logging console
```

You can re-enable console logging at any time with the `logging console` command.

```
Router> en
Router# config t
Router(config)# logging console
```

Undo the Effects of a Command

Now is a great time to formally introduce you to the `no` command. You’ve already used it with the `no logging console` command. You can place the word “no” before almost any configuration command to negate it. For example, if you decide you don’t want a hostname assigned to your router, you can use the configuration mode command `no hostname` to undo that configuration setting.

Configuring Your LAN Interface

Listing 2-1 displays all the steps necessary to set your router’s hostname and configure an IP address on your LAN interface `vlan1`.

Listing 2-1. *How to Configure Your LAN Interface*

```
Router> ena
Router# config t
Router(config)# hostname lab-r1
lab-r1(config)# int vlan1
lab-r1(config-if)# descr LAN switch ports on inside interface
lab-r1(config-if)# ip address 192.168.1.1 255.255.255.0
lab-r1(config-if)# no shut
lab-r1(config-if)# exit
lab-r1(config)# exit
lab-r1#
```

Let’s go over these commands in more detail.

Step 1: Assign a Hostname to Your Router

First, log in to privileged EXEC mode using the `enable` command, then enter global configuration mode using the command `configure terminal` (abbreviated `config t`) and use the `hostname` command to assign a hostname to your router. In Listing 2-1, the hostname is set to “lab-r1” (lab router one).

Step 2: Start Interface Configuration Mode

To configure an interface, you must be in interface configuration mode. Use the global configuration mode command `interface vlan1` (abbreviated `int vlan1`) to start interface configuration mode on the `vlan1` interface. Did you notice the prompt change? You can always tell when you are in interface configuration mode by the prompt (`config-if`).

Step 3: Add a Description to Your Interface

You can add a description to your interface using the command `description` (abbreviated `descr`) while in interface configuration mode. This feature can be used to permanently place reminders or other information about the interface in your router’s configuration file.

Step 4: Assign an IP Address to Your Interface

The command used to assign an IP address to an interface is `ip address`. The private address and mask `192.168.1.1 255.255.255.0` (shown in Listing 2-1) are assigned to interface `vlan1`. Private addresses are reserved for LANs. The network address `192.168.1.0` is the most commonly used, so we’ll use it here, but you can also use `172.16.0.0` or `10.0.0.0` (Cisco is fond of `10.10.10.0 255.255.255.0`).

Note To learn more about IP addresses and subnet masks please refer to Chapter 6.

Step 5: Bring Up the Interface

Before an interface can be used, it must be brought up. The command `no shutdown` (abbreviated `no shut`) will bring up an interface. You may wonder, why shut down an interface? Sometimes you may want to manually shut down an interface, while you configure it, and then bring it back up when you’re done. To take an interface down, use the

shutdown command while in interface configuration mode. When you shut down an interface with the shutdown command, its status is referred to as “administratively down.”

Tip When configuring an interface, it’s a good habit to bring up the interface when you’re done. Otherwise, you may forget to bring it up later and will need to troubleshoot the interface when it’s not functioning as expected.

Step 6: Exit from Interface Configuration Mode

Use the `exit` command to exit from interface configuration mode. It’s worth noting here that the `exit` command is always available to you, and it will always take you back to the previous mode. You can also use the key combination `Ctrl+Z` to exit directly to privileged EXEC mode.

Step 7: Check Your Work

Throughout the configuration process you should periodically check your work using the privileged EXEC mode command `show running-configuration` (abbreviated `sh run`).

```
lab-r1> ena
lab-r1# sh run
```

Use the space bar to scroll through the configuration until you see “interface Vlan1,” as shown in the following example:

```
interface Vlan1
  description LAN switch ports on inside interface
  ip address 192.168.1.1 255.255.255.0
  hold-queue 32 in
```

Notice that you can see the description you provided for `vlan1`, “LAN switch ports on inside interface” as well as the IP address you assigned the interface. You may also see other commands in your configuration that you did not configure on your router. This is perfectly normal. These commands are defaults set by the IOS and will vary from one IOS version to another. If you upgrade your IOS, you may find new commands and options set by the IOS in your configuration.

Configuring a DHCP Server

DHCP is a great tool for managing your IP addresses. It will assign IP addresses to your hosts and keep track of all those addresses for you. Without DHCP, you would need to assign IP addresses manually and keep track of all those addresses to ensure that they are correct and without duplicates. We refer to a manually assigned address as a “static IP address.” As a rule, only shared devices, such as servers, printers, routers, and switches need a static IP address. All other devices should use DHCP-assigned IP addresses.

Listing 2-2 presents an overview of how to configure a DHCP server. The subsections that follow the listing go into the details.

Listing 2-2. *How to Set Up a DHCP Server*

```
lab-r1> ena
lab-r1# config t
lab-r1(config)# ip dhcp pool MYNET
lab-r1(dhcp-config)# network 192.168.1.0 255.255.255.0
lab-r1(dhcp-config)# domain-name geekvenue.net
lab-r1(dhcp-config)# default-router 192.168.1.1
lab-r1(dhcp-config)# dns-server 66.238.5.2 66.238.5.3
lab-r1(dhcp-config)# netbios-name-server 192.168.1.2
lab-r1(dhcp-config)# lease 0 2
lab-r1(dhcp-config)# exit
lab-r1(config)# ip dhcp excluded-address 192.168.1.1 192.168.1.25
```

Step 1: Define the DHCP Pool Name

The first step is to define a name for your DHCP pool. The global configuration mode command `ip dhcp pool` is used to define a DHCP pool and places you in `dhcp pool configuration` mode. You can confirm you’re in `dhcp pool configuration` mode by the prompt (`dhcp-config`). Cisco provides pool naming so you can define and keep track of multiple DHCP pools. This feature allows enterprise networks, which have multiple subnets, to manage those subnets with DHCP. However, since this book is about using a router for a small business, we will have only one pool and name it “MYNET” using the command `ip dhcp pool MYNET`.

Step 2: Define the Network Address for DHCP

The command `network` defines which network address DHCP will be using. In Listing 2-2, the private network address and mask are `192.168.1.0 255.255.255.0`.

Step 3: Define Your Domain Name

A DHCP server provides your hosts with more than their IP addresses. The `domain-name` command defines which domain name all of your hosts will be assigned. If you don't have a registered domain name (e.g., `mycompany.com`), then using a `.local` name such as `mycompany.local` or `mycompany.lan` will suffice (both are commonly used).

Step 4: Define the Default Gateway

The command `default-router` defines where your gateway is: that is, it tells your workstations where to find a route to the Internet. However, since your Cisco router *is* your Internet router, it is also your gateway and its IP address is used as the default router. In Listing 2-2, the IP address is `192.168.1.1`.

Step 5: Define Your DNS Servers

Domain Name System (DNS) lookups are used to find the IP address of a host on a network, and DNS servers provide the information. This is how it works: when you type `www.cisco.com` into your browser, your PC performs a DNS lookup to match `www.cisco.com` to the IP address for that web server. The address returned is then used to take you to that web site. Use the command `dns-server` to define your DNS servers. The DNS servers `66.238.5.2` and `66.238.5.3` in Listing 2-2 are fictional. If you run your own DNS servers on your LAN, enter their IP addresses here; otherwise, use those provided to you by your Internet Service Provider.

Step 6: Define a WINS Server (Optional)

If you have a Microsoft WINS (Windows Internet Name Service) server on your network that you want accessible to your workstations, use the command `netbios-name-server` and the IP address of your WINS server. The configuration in Listing 2-2 is using a fictional WINS server at address `192.168.1.2`. If you don't have a WINS server or even know what one is, then skip this step.

Step 7: Define a DHCP Lease Time

The command `lease` defines how long your hosts may keep their DHCP-assigned IP addresses. One way that DHCP manages all your IP addresses is by leasing out IP addresses to your hosts for a short period of time. The time, defined in Listing 2-2, is 0 days and 2 hours. Therefore, every two hours a host asks the server if it can keep its DHCP-assigned IP address, which the server typically allows. However, if your router doesn't hear from that host two hours after leasing it an address, the DHCP server

assumes the host is no longer on the network and releases its address back into the pool for some other host to use.

Step 8: Define a DHCP-Excluded Address Range

Use the command `ip dhcp excluded-address` to define the addresses that will not be allocated by your DHCP server. These will be the static IP addresses you assign your servers, printers, switches, and routers. It's best to define a block of addresses right from the start. That way you won't need to reconfigure your router later to reserve (exclude) more addresses.

In Listing 2-2, the first 25 IP addresses are reserved. The starting address is 192.168.1.1 and the ending address is 192.168.1.25. Reserve as many static IP addresses as you need, but 25 should be plenty for a small business.

Step 9: Test DHCP Using a Workstation

Before you continue, it's a good idea to test your DHCP server. Make sure that your PC is configured to use DHCP, and plug the laptop or workstation PC into one of the four Ethernet switch ports on the back of your router.

To check if your workstation has successfully received an IP address from your router's DHCP server, use the Windows command `ipconfig`. First, go to the Windows command prompt. Click "Start/run" and type **cmd** into the "open:" field then click the "OK" button. At the "C:\>" command prompt, issue the `ipconfig` command.

```
C: \> ipconfig
```

Notice from the following command result, that basic information about your workstation's IP configuration is provided.

Windows IP Configuration

Ethernet Adapter Local Area Connection

```
Connection-Specific DNS Suffix: . . . . geekvenue.net
IP Address: . . . . . 192.168.1.26
Subnet Mask: . . . . . 255.255.255.0
Default Gateway:. . . . . 192.168.1.1
```

To get more detailed information, including information about your DHCP server, DNS servers, when your lease was obtained, and when it will expire, use the Windows command `ipconfig/all`.

Tip If your workstation is running a UNIX operating system such as Linux, FreeBSD, or Macintosh OSX, use the command `ifconfig` from the UNIX console to display your workstation's IP configuration.

Step 10: Check Your DHCP Status with the IOS

You can use the IOS command `show ip dhcp binding` (abbreviated `sh ip dhcp bind`) to check DHCP lease information on your router.

```
lab-r1> ena
lab-r1# sh ip dhcp bind
```

You will be presented with output similar to the following:

Bindings from all pools not associated with VRF:			
IP address	Client-ID/ Hardware address/ User name	Lease expiration	Type
192.168.1.26	0100.Obd.deed.9	Dec 17 2008 12:28 PM	Automatic

Notice that the first IP address assigned by your DHCP server is 192.168.1.26. That's because you reserved addresses 1 through 25 for static IP hosts, such as servers, printers, routers, and switches. Also note that this lease will expire on "December 17 2008 at 12:28 PM." When an IP address lease expires, the host that received the lease will need to request a renewal of the lease to continue using the IP address. If the host does not request a renewal, the server releases the address back into the pool and may assign it to some other host.

Configuring Telnet on Your Router

A telnet application allows you to log in and configure your router remotely. If you use telnet, then you won't need to sit huddled next to your router in a dark utility closet, configuring it via the console port. It's a handy tool. However, the use of telnet does expose you to some security risks. First, telnet sends all information unencrypted, in clear text, across your network. This means that some clever soul may be able to capture your data with a network analyzer and obtain your passwords, router configuration, or other sensitive information. He or she could even hijack your router. Also, if you don't protect your

router with some sort of ACL, then anyone from any host can log in remotely—even from the Internet! If you don't know what an ACL is, don't worry, I'm going to explain it to you in the next section, "Securing VTY."

The good news is that you can take precautions to minimize your risk. Later, in Chapter 5, I'll introduce you to a much better alternative to telnet, known as Secure Shell (SSH), which will encrypt all of your data.

Listing 2-3 shows you all of the IOS commands to set up telnet on your router, and Listing 2-4 in the next section on "Securing VTY" shows how to secure it with an ACL. Security is vital when configuring an Internet router, so pay close attention to information and concepts covered in this section.

Listing 2-3. *How to Configure Telnet on Your VTY Line*

```
lab-r1> ena
lab-r1# config t
lab-r1(config)# enable secret cisco
lab-r1(config)# line vty 0 4
lab-r1(config-line)# password cisco2
lab-r1(config-line)# login
lab-r1(config-line)# exit
```

Step 1: Set Your Privileged EXEC Mode Password

The IOS requires you to set the privileged EXEC mode password before allowing you to log in using telnet. Use the global configuration mode command `enable secret` to set the password.

Note This was covered in Chapter 1 in the section "Set the Privileged EXEC Mode Password."

Step 2: Set Your VTY Login Password

Your router's telnet server listens for connections on what Cisco calls a VTY line (Virtual Terminal line). The IOS requires that you set a VTY password before you can log in via the VTY line. Use the command `line vty 0 4` to start VTY configuration mode, which is where you configure your router's telnet server. There are five virtual VTY lines, labeled 0 through 4, on a Cisco 800 series router. You can confirm you're in VTY configuration mode when you see the `(config-line)` prompt. Use the `password` command to set the VTY password. In Listing 2-3, the password is set to "cisco2". Finally, use the `login` command to enable password checking on the VTY line.

At this point, you should be able to plug a workstation into one of your router's Ethernet switch ports and use the telnet application to connect to your router using IP address 192.168.1.1 (your router's address). At the Windows command prompt, type the following:

```
c: /> telnet 192.168.1.1
```

You should receive a password prompt, at which you enter your VTY password.

```
Password: cisco2
```

After logging in to the router, you will be in user EXEC mode. You can use the enable command to enter privileged EXEC mode.

```
lab-r1> enable
```

Next, you will be prompted for your privileged EXEC mode password.

```
Password: cisco
```

```
lab-r1#
```

After you type in the privileged EXEC mode password, "cisco" in this case, you are placed in privileged EXEC mode as indicated by the # sign in the prompt.

Securing VTY

Now that telnet is working, it's a good idea to restrict which hosts have access to your router's VTY lines. The general rule is to only allow the PC's on your private network access to the router. Why? Because later on, when your router is connected to the Internet, it will prevent anyone on the Internet from gaining access to it.

In the Cisco IOS you control access to your router with an ACL. An ACL is a powerful IOS tool that allows you to secure your router using a combination of IP addresses and protocols. In simple terms, an ACL is a list of rules that either permits or denies a host (or group of hosts) access to another host (or group of hosts) through your router. It's a filter. After you create your ACL, it's applied to a VTY line or Ethernet interface, and you specify whether it's filtering incoming or outgoing traffic.

I'll go into more detail about ACLs later in this chapter. For now, let's look at the ACL for telnet line by line in Listing 2-4.

Listing 2-4. *How to Secure VTY (Telnet) with an ACL*

```
lab-r1(config)# ip access-list standard VTY-ACL
lab-r1(config-std-nacl)# permit 192.168.1.0 0.0.0.255
lab-r1(config-std-nacl)# exit
lab-r1(config)# line vty 0 4
lab-r1(config-line)# access-class VTY-ACL in
```


Step 1: Create and Name Your ACL

The first thing you do is create and name your ACL. The name “VTY-ACL” is used in Listing 2-4. You can use any name that makes sense to you, but it’s a good idea to use a name that will remind you of what the ACL is used for. The command for this is

```
lab-r1(config)# ip access-list standard VTY-ACL
```

Notice from the following output that the prompt has changed to (config-std-nacl), indicating that you are in ACL configuration mode. The following is the prompt for a standard named ACL:

```
lab-r1(config-std-nacl)#
```

Now that you’ve named and created your ACL, the IOS puts you in ACL configuration mode, allowing you to create ACL rules. The rule `permit 192.168.1.0 0.0.0.255` allows traffic from all IP hosts on network 192.168.1.0.

Note The first part of the command (192.168.1.0) is a starting address and the second part (0.0.0.255) is a wildcard mask, not a subnet mask!

In the following rule, a zero (0) in the wildcard mask represents an exact match and the number (255) represents all hosts on our network:

```
lab-r1(config-std-nacl)# permit 192.168.1.0 0.0.0.255
```

Step 2: Apply Your ACL to VTY

An ACL can be applied to an interface or VTY line, but in this case we need to apply it to our VTY line. Exit from ACL configuration mode, go to VTY configuration mode, and apply the rule using the `access-class` command. Notice in Listing 2-4 that you need to specify the direction in which the ACL is applied. In this case, apply the rule to incoming traffic using the `in` statement with these commands:

```
lab-r1(config)# line vty 0 4
lab-r1(config-line)# access-class VTY-ACL in
```

Now is a good time to use the `show running-configuration` command (abbreviated `sh run`) to check your progress so far.

Configuring Your WAN Interface—Dynamic IP

The next step is to set up your Wide Area Network (WAN) interface. Any router interface that connects to the Internet can be referred to as a WAN port. There are two ways to set an IP address on your WAN port. The first method (illustrated in Listing 2-5) is to use a DHCP-assigned address from your ISP, and the second (illustrated in Listing 2-6) is to use a static IP address that you request from your ISP. You request a static IP address when you are hosting Internet services (e-mail, www, etc.) on your private network. The set of commands in Listing 2-5 will configure your WAN interface to use a dynamically-assigned IP address.

Listing 2-5. *How to Configure Your WAN Interface for DHCP*

```
lab-r1> ena
lab-r1# config t
lab-r1(config)# int fa4
lab-r1(config-if)# descr WAN interface to ISP using DHCP
lab-r1(config-if)# ip address dhcp client-id FastEthernet4
lab-r1(config-if)# duplex auto
lab-r1(config-if)# speed auto
lab-r1(config-if)# no shut
lab-r1(config-if)# exit
lab-r1(config)# ip domain-lookup
lab-r1(config)# exit
lab-r1#
```

Step 1: Start Interface Configuration Mode

To configure an interface, you must be in interface configuration mode. Use the global configuration mode command `interface FastEthernet4` (abbreviated `int fa4`) to start interface configuration mode for the FastEthernet4 interface (WAN port).

Step 2: Add a Description to Your Interface

You can add a description for your WAN interface just as you did for your LAN interface in Listing 2-1.

Step 3: Configure Your WAN Interface to Use DHCP

The command `ip address dhcp client-id FastEthernet4` used in Listing 2-5, instructs your router to request a DHCP-assigned address from your ISP using WAN interface fa4.

Step 4: Set the Duplex and Speed on Your Interface

Next, the `duplex auto` statement configures your router to automatically set its duplex to full or half, based on the type of cable modem your ISP provides. An interface set to use half duplex means the interface can send or receive data at any given time, but not both at the same time. Full duplex, which essentially doubles your interface's bandwidth, allows the interface to send and receive data simultaneously.

Use the command `speed auto` to ensure that the port automatically adjusts its speed between 10 and 100 Mbps (if your router supports it).

Step 5: Bring Up the Interface

Bring up the interface with the command `no shutdown` (abbreviated `no shut`).

Step 6: Enable Domain Lookup

Finally, use the command `ip domain-lookup` to permit your router to perform DNS name lookups.

At this point, you should be able to plug your cable modem into interface fa4 on your router and *ping* an Internet web site from the router's CLI. Ping is a utility that allows you to test whether a host is reachable on a network. The ping command sends an "echo-request" to a host; if the host is reachable, it replies with an "echo-reply." Some network security policies prevent echo-replies so you may need to ping a couple of sites. Try pinging your ISP first, if you receive an echo-reply, then you know TCP/IP is configured and working correctly.

When you ping a host from the Cisco CLI, a row of exclamation points (!!!!!) indicates a successful reply from the host. Each exclamation point represents a single packet that has been replied to successfully. On the other hand, dots (.....) indicate that you have received no response from the host. Each dot represents a single packet that has been lost and was not replied to successfully. You can use the ping command to ping a host by IP address or by a fully qualified domain name, as in this example:

```
lab-r1# ping www.yourisp.net
```

In the following example, you can see that your ISP's fully qualified domain name has been resolved to IP address "66.238.5.101" by a DNS server at IP address "66.238.5.2"; the ISP's host successfully replied to all five packets your router sent to it (as indicated by the five exclamation points).

```
Translating "www.yourisp.net"...domain server (66.238.5.2) [OK]
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 66.238.5.101, timeout is 2 seconds:
```

```
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 180/208/228 ms
```

```
lab-r1#
```

Configuring Your WAN Interface—Static IP

The only difference between using DHCP or a static IP address on your WAN interface is that with a static IP, you must configure your WAN interface manually with the IP information that your ISP assigns you. When you request a static IP address from your ISP, make sure you receive the following information: your static IP address and mask, the ISP's default gateway address, and DNS server addresses. The following is the static IP information from my fictitious Internet Service Provider, ACME ISP:

Static IP Address: 66.238.5.254 255.255.255.0

Default Gateway: 66.238.5.1

Acme DNS1: 66.238.5.2

Acme DNS2: 66.238.5.3

The commands in Listing 2-6 can configure your router using the static IP information provided by ACME ISP.

Listing 2-6. *How to Configure Your WAN Interface Using a Static IP Address*

```
lab-r1> ena
lab-r1# config t
lab-r1(config)# int fa4
lab-r1(config-if)# descr WAN interface to ISP using a STATIC IP
lab-r1(config-if)# ip address 66.238.5.254 255.255.255.0
lab-r1(config-if)# duplex auto
lab-r1(config-if)# speed auto
lab-r1(config-if)# no shut
```

```
lab-r1(config-if)# exit
lab-r1(config)# ip route 0.0.0.0 0.0.0.0 66.238.5.1
lab-r1(config)# ip domain-lookup
lab-r1(config)# ip name-server 66.238.5.2 66.238.5.3
lab-r1(config)# exit
lab-r1#
```

Step 1: Start Interface Configuration Mode

Start interface configuration mode for interface fa4 by using the global configuration mode command `interface fastethernet4` (abbreviated `int fa4`).

Step 2: Add a Description to Your Interface

Optionally, you can add a description to your interface with the `description` command (abbreviated `descr`) just as you’ve done previously.

Step 3: Assign an IP Address to Your Interface

Use the command `ip address` to assign an IP address to your interface. The fictional ISP address in Listing 2-6 is 66.238.5.254 255.255.255.0. Replace this with a real IP address assigned to you by your ISP.

Step 4: Set the Duplex and Speed on Your Interface

The command `duplex auto` configures your router to automatically set its duplex to full or half duplex, based on the type of cable modem your ISP provides. Use the `speed auto` command to adjust the port speed automatically between 10 and 100 Mbps.

Step 5: Bring Up the Interface

Next, bring up the interface with the command `no shutdown` (abbreviated `no shut`).

Step 6: Assign the Default Gateway

Now, use the `ip route 0.0.0.0 0.0.0.0 {gateway-address}` command to configure your router’s default gateway address (provided by your ISP). Cisco often refers to this as the “gateway of last resort.” A gateway of last resort is where your router sends packets that are not destined for your local network. In the case of an Internet router, the gateway address is always provided by your ISP. I like to think of it as the onramp to the Internet.

Step 7: Enable Domain Lookup

The last step is to enable domain lookup using the command `ip domain-lookup`, and assign DNS servers using the command `ip name-server`. The name servers used in the Listing 2-6 example are 66.238.5.2 and 66.238.5.3.

At this point, you should be able to successfully ping a web site.

Configuring NAT on Your Router

There are just a few more things to be done before your router is ready to be used on the Internet. You need to configure NAT and apply a firewall to protect your network from the outside world.

NAT is a clever technology that allows all the computers on your network to access the Internet using only one registered IP address. It works by tagging host data before it leaves your network bound for the Internet. When a reply with information (say a web site you requested) returns to your router, its data contains the same tags. NAT then looks in its address translation table to determine to which computer on your private network the data should be directed. It then sends the data on to the correct workstation using the workstation's IP address. Shazam! Your user sees the web page they requested in their browser.

NAT was created out of the legitimate fear that the Internet was going to run out of IP addresses. It's a wonderful tool because you can have hundreds of PCs on your LAN, all using one registered IP address. All of the other addresses (on your LAN) are *private* IP addresses. Private IP addresses are simply IP addresses that have been reserved for use on private networks. They're not routable on the Internet, except via NAT, so multiple disparate private networks can use the same private network address (the most common is 192.168.1.0 255.255.255.0).

The Internet Assigned Numbers Authority (IANA) has reserved the following network addresses for use on any private network:

10.0.0.0 through 10.255.255.255

172.16.0.0 through 172.31.255.255

192.168.0.0 through 192.168.255.255

The zeros in these network numbers may be replaced with other numbers to form a valid private network address such as 10.10.10.0.

Listing 2-7 displays all the commands needed to configure NAT on your router.

Listing 2-7. How to Configure NAT

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# ip access-list extended NAT-ACL
lab-r1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 any
lab-r1(config-ext-nacl)# exit

lab-r1(config)# ip nat inside source list NAT-ACL interface fa4 overload

lab-r1(config)# int vlan1
lab-r1(config-if)# ip nat inside

lab-r1(config-if)# int fa4
lab-r1(config-if)# ip nat outside
```

Step 1: Create and Name an Extended ACL for NAT

The first step is to create an ACL that allows all of your hosts to use NAT. This is an *extended* ACL, not a standard ACL like you used earlier for securing your VTY lines (telnet). An extended ACL allows you to filter not only by IP address but also by protocol. The global configuration command `ip access-list extended NAT-ACL` creates an extended access list named “NAT-ACL”. You can confirm you’re working with an extended ACL because the prompt changes to `(config-ext-nacl)`.

Step 2: Create an ACL Rule

Only one rule is required to permit all of your hosts access via NAT. The extended access rule `permit ip 192.168.1.0 0.0.0.255 any` will allow all of your private IP hosts access.

Step 3: Configure Inside Address Translation

The global configuration mode command `ip nat inside source list NAT-ACL interface fa4 overload` tells NAT how to translate addresses. In this case, you’re instructing NAT to translate your inside addresses, based on your ACL “NAT-ACL” and “overload” them on to your single outside IP address on interface fa4 (your WAN interface).

Step 4: Apply NAT to Your Interfaces

For NAT to work properly, it's important that it knows which interface is the inside (private) interface and which is the outside (public) interface. Go into interface configuration mode and issue the command `ip nat inside` on your private LAN interface `vlan1` and the command `ip nat outside` on your public WAN interface `fa4` (see Listing 2-7).

Securing Your Interfaces

Because the worldwide Web is filled with all sorts of worldwide weenies, who may want to bring down your network, there are a few standard features of the IOS that you'll want to disable on your interfaces. I think Cisco should disable these features by default, but I'm sure it has its reasons for doing things the way it does.

Listing 2-8 displays the commands that should be applied to both your LAN and WAN interfaces to ensure a high level of security on your router.

Listing 2-8. *How to Secure Your Interfaces*

```
lab-r1> ena
lab-r1# config t
lab-r1(config)# int fa4
lab-r1(config-if)# no ip unreachable
lab-r1(config-if)# no ip redirects
lab-r1(config-if)# no ip proxy-arp

lab-r1(config-if)# int vlan 1
lab-r1(config-if)# no ip unreachable
lab-r1(config-if)# no ip redirects
lab-r1(config-if)# no ip proxy-arp
lab-r1(config-if)# exit
lab-r1(config)# exit
```

Step 1: Disable IP Unreachable Messages

The interface configuration command `no ip unreachable` prevents a router from generating messages about unreachable destinations. An excessive amount of Internet control message protocol (ICMP) unreachable replies could overload a router, resulting in a denial of service (DOS) attack.

Step 2: Disable IP Redirects

IP redirects are used to inform a host of a better route to a destination. IP redirected routes override a router's routing tables and could be used in a DOS attack. Use the interface configuration command `no ip redirects` to disable this feature.

Step 3: Disable Proxy ARP

In a very small nutshell, address resolution protocol (ARP) proxying is when a router, sitting between two networks, replies to an ARP request on behalf of one network to the other. It's very unlikely you will need this feature in a small business network. In fact, it could be used in a malicious manner and should therefore be disabled. Use the interface command `no ip proxy-arp` to disable this feature.

Creating a Basic Firewall

In this section, you're going to learn how to secure your Internet router using the Cisco IOS firewall. Cisco uses a *stateful* packet inspection firewall that works at the application layer. Unlike traditional packet filtering, which examines a packet based on the information in the packet header, stateful inspection tracks the state of all packets as they traverse the interfaces of your router to ensure they've not been the victim of tampering. Cisco refers to this as Context Based Access Control (CBAC). Traditional firewalls are more like an ACL, permitting or denying traffic based on IP address and protocol, not inspecting the packets for tampering. Listing 2-9 shows the essential commands to create and configure a basic firewall.

Listing 2-9. *How to Configure a Basic IOS Firewall*

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# ip inspect name IPFW tcp timeout 3600
lab-r1(config)# ip inspect name IPFW udp timeout 15
```

To configure a basic firewall, issue the global configuration mode command `ip inspect name` to create and name your firewall. In Listing 2-9 the name used is "IPFW". Next, define which protocols will be inspected (TCP and UDP in this case). Optionally, define a protocol timeout; 3,600 seconds is the recommended timeout for TCP (connection-oriented protocols) and 15 to 30 seconds for UDP (connectionless protocols).

Creating an Advanced Firewall

The Cisco IOS firewall is not only designed to examine protocols. It can also examine data for specific applications. Anytime a new version of the Cisco IOS becomes available, the list of supported applications expands. Listing 2-10 gives a few firewall rules that inspect some common applications.

Listing 2-10. *How to Configure an Advanced IOS Firewall*

```
lab-r1(config)# ip inspect name IPFW cuseeme
lab-r1(config)# ip inspect name IPFW ftp
lab-r1(config)# ip inspect name IPFW tftp
lab-r1(config)# ip inspect name IPFW rcmd
lab-r1(config)# ip inspect name IPFW realaudio
lab-r1(config)# ip inspect name IPFW smtp
lab-r1(config)# ip inspect name IPFW h323
lab-r1(config)# exit
lab-r1#
```

Step 1: Create Application Rules

Application layer rules are created the same way as basic firewall rules except you use the global configuration mode command `ip inspect name {firewall-name} {application-name}`. Use the IOS command `ip inspect name IPFW ?` to see a list of all the supported applications.

Step 2: Apply the Rules Outbound

After you name your firewall and create all the protocol and application inspection rules, you need to apply the firewall to an interface. Since you're using your router as an Internet gateway, your firewall "IPFW" is applied to the WAN interface, as shown in Listing 2-11.

Listing 2-11. *How to Apply a Firewall to an Interface*

```
lab-r1> ena
lab-r1# config t
lab-r1(config)# int fa4
lab-r1(config-if)# ip inspect IPFW out
lab-r1(config-if)# exit
lab-r1(config)# exit
lab-r1#
```

Go to interface configuration mode and issue the command `ip inspect IPFW out` to apply the firewall named IPFW to outbound traffic on your WAN interface fa4. This begs the obvious question: how can applying a firewall to *outbound* traffic on your WAN

interface protect you from bad guys coming in from the Internet on your WAN interface? The answer is that your IOS firewall (IPFW) will inspect the traffic leaving your workstations, exiting the router's WAN port on interface fa4, and allow that same traffic back in through your WAN interface fa4, regardless of which ACL is configured on the WAN port. Cisco refers to this as "pinhole security."

Creating an ACL for Your WAN Interface

There are two concepts to consider when creating ACLs: *default to permit* and *default to deny*. Default to permit means that all traffic is permitted unless it is explicitly denied. By contrast, a default to deny ACL denies all traffic unless it is explicitly permitted. When creating an Internet firewall, you always want to follow the default to deny standard. It's easier to configure, because it requires fewer rules, and it's a lot more secure because you don't need to create a rule for everything you want to deny.

Now that you know a little bit about ACLs and how the Cisco CBAC firewall works, let's tighten up your WAN interface with a firewall ACL that permits replies to outbound ping and traceroute requests, but denies all other incoming traffic (unless of course it's returning traffic inspected by the IOS firewall). Traceroute is a utility that tracks the route a packet takes as it travels across an IP network. It's important to note here that ping and traceroute are IP applications that use the ICMP that the CBAC firewall cannot handle. Therefore, to allow ping and traceroute packets from your router to return from the Internet, you must explicitly allow them in with a set of ACL rules.

Listing 2-12 gives the rules for an ACL that denies all traffic except for returning replies to outbound ping and traceroute requests. There is also a rule included for Generic Routing Encapsulation (GRE), which is commonly used with Point-to-Point Tunneling Protocol (PPTP) for remote access to Microsoft Windows servers:

Listing 2-12. *How to Create a Firewall ACL for Your WAN Interface*

```
lab-r1> enable
lab-r1# config t

lab-r1(config)# ip access-list extended IPFW-ACL
lab-r1(config-ext-nacl)# permit icmp any any administratively-prohibited
lab-r1(config-ext-nacl)# permit icmp any any echo-reply
lab-r1(config-ext-nacl)# permit icmp any any packet-too-big
lab-r1(config-ext-nacl)# permit icmp any any time-exceeded
lab-r1(config-ext-nacl)# permit icmp any any traceroute
lab-r1(config-ext-nacl)# permit gre any any
lab-r1(config-ext-nacl)# deny ip any any
lab-r1(config-ext-nacl)# exit
lab-r1(config)#
```

Step 1: Allow Ping and Traceroute

The ACL in Listing 2-12 denies all IP traffic except for replies to outbound ping and traceroute requests. All you need to do now is apply the ACL to inbound traffic on your WAN interface FastEthernet4.

Step 2: Apply the ACL Inbound

Listing 2-13 shows how to apply your firewall ACL to incoming Internet traffic on your WAN interface. This will ensure that the firewall inspects all inbound traffic.

Listing 2-13. *How to Apply a Firewall ACL to Your WAN Interface*

```
lab-r1> ena
lab-r1# config t
lab-r1(config)# int fa4
lab-r1(config-if)# ip access-group IPFW-ACL in
lab-r1(config-if)# exit
lab-r1(config)# exit
lab-r1#
```

Use the interface configuration command `ip access-group IPFW-ACL in` to apply the firewall ACL named “IPFW-ACL” to your WAN interface. The `in` statement used with the `ip access-group` command ensures that your router filters incoming traffic from the Internet through your WAN interface (fa4 in this case).

That’s all there is to setting up a firewall. You might want to examine all of the applications you use on your network and apply the appropriate IOS firewall statements to better secure your network, but the rules I have provided are suitable for most small business environments.

Configuring a Basic DMZ

The last configuration I would like to cover in this chapter is how to create a DMZ host. This is a fancy way of saying you’re going to open a hole in your firewall ACL that will allow traffic from the Internet to access a host that provides a service on your private network.

You can create ACL rules that will allow access to any internal service from the Internet. SMTP e-mail and HTTP are two of the most common.

In Listing 2-14, it is assumed that you have an e-mail server on your private network. I’m going to show you how to allow incoming e-mails to be delivered to that server from the Internet. There are two steps to the process. First, modify your WAN interface’s ACL (IPFW-ACL) to allow SMTP (Simple Mail Transport Protocol) in via the WAN interface. SMTP is the protocol used by e-mail servers to accept incoming messages. Next, use NAT to redirect that traffic to the mail server on your private network.

Listing 2-14. *How to Create a New IPFW-ACL*

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# no ip access-list extended IPFW-ACL

lab-r1(config)# ip access-list extended IPFW-ACL
lab-r1(config-ext-nacl)# permit tcp any host 66.238.5.254 eq smtp
lab-r1(config-ext-nacl)# deny ip any any
lab-r1(config-ext-nacl)# exit
lab-r1(config)# exit
lab-r1#
```

Step 1: Remove the Existing IPFW-ACL

You begin by removing your existing IPFW-ACL using the global configuration mode command `no ip access-list extended IPFW-ACL`. This statement will *remove* the ACL from your router and delete all the rules. If you have a complex ACL, you may want to modify it instead.

Tip The easiest way to modify ACL rules is to keep a copy of the rules in a text document and use copy/paste to create them on your router (if your terminal software supports the copy and paste function). It can save you a lot of typing time.

Step 2: Create a New IPFW-ACL

Create a new ACL that allows SMTP e-mail traffic to access your WAN interface using the command `permit tcp any host 66.238.5.254 eq smtp`. This command permits any host to access the static IP address of your WAN interface (66.238.5.254 in this example) but only allows them to connect to the TCP port for SMTP, which is logical port number 25. SMTP always listens on TCP port 25. We could replace “SMTP” in the statement with its TCP port number “25” and the ACL will work identically.

Notice in Listing 2-14 that there’s a `deny ip any any` statement at the end of the ACL. This rule was added to illustrate a point: in Cisco IOS, a deny statement at the end of an ACL is completely unnecessary. There is *always* an implied deny at the end of any ACL. That’s because Cisco adheres to the default to deny approach to security with ACLs.

Step 3: Configure NAT to Forward Traffic to a LAN Host

Now that you have opened a hole in your IPFW-ACL for incoming e-mail, you need to tell the router what to do with the traffic. Typically, it’s forwarded to a static IP host on your

private network. In this case it should be forwarded to your e-mail server. We refer to that host as being a DMZ host. Any private host that is directly reachable from the Internet is a DMZ host. The commands in Listing 2-15 are used to configure your router to forward incoming e-mails to SMTP (TCP port 25) on your e-mail server, which in this example is at static IP address 192.168.1.3 on your private network:

Listing 2-15. *How to Configure Port Forwarding with NAT*

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# ip nat inside source static tcp 192.168.1.3 25 interface fa4 25
lab-r1(config)# exit
lab-r1#
```

Step 4: Apply the Inside Source Rule

In Listing 2-15, the global configuration mode command `ip nat inside source static` is used to redirect incoming WAN traffic on port 25, to port 25 of the DMZ host at static IP address 192.168.1.3 (which is your e-mail server). In this step, NAT is being used as a traffic cop to redirect acceptable traffic from the Internet to the DMZ host on your LAN. E-mail messages destined for TCP port 25 on the WAN interface are redirected to TCP port 25 on the e-mail server, making the delivery of e-mail messages transparent to the e-mail servers that send them.

Note Port 25 is a logical TCP/IP port and has nothing to do with the physical ports on your router. All Internet applications use port numbers to ensure that client applications and servers (e.g., web servers and web browsers) can communicate with one another.

Table 2-2 shows some common TCP/IP services and their TCP/IP port numbers that you may want to use to set up your network:

Table 2-2. *Common TCP/IP Services and Their TCP/IP Port Numbers*

Service	Port Number
Secure Shell	22
Email Server	25
Web Server	80

Service	Port Number
POP3	110
POP3s (secure pop)	995
Secure Web Server	443
MS Terminal Server (RDP)	3389
MS SBS Remote Web Workplace	4125
Point-to-Point Tunneling Protocol (PPTP)	1723

Remember, first create an ACL (IPFW-ACL) that allows a service like SMTP e-mail, and apply it to your WAN interface, then use a NAT rule to redirect that traffic to the appropriate host on your private LAN. That host needs to be assigned a static IP address.

Saving Your Configuration

After making any configuration changes to your router, you will want to save your running configuration. If you don't, the next time you restart your router those changes will not be in your configuration and your router will not operate as expected.

Use the privileged EXEC mode command `copy running-configuration startup-configuration` (abbreviated `copy run start`) to save your running configuration file to your startup configuration file.

```
lab-r1> ena
lab-r1# copy run start
```

You will then be prompted for a destination file name.

```
Destination filename [startup-config]?
```

Press the ENTER key to use the default name "startup-config".

Caution You do not want to change the default startup configuration name "startup-config" because it is the name of the configuration your IOS looks for and loads when you power up or reload your router.

Restoring the Default Configuration

Cisco always stores a factory default configuration file in the flash memory of its routers. If you ever want to restore your router's configuration to the factory default, just as it was when you pulled it out of the box, all you need to do is copy the default configuration file (stored in flash or webflash) to the startup-config and restart your router. Use the command `show webflash` or `show flash` to display the exact name of the configuration file. The default configuration file name will always end with ".cfg". On older Cisco 800 series routers, the name of the file is "ConfigExp.cfg" and is stored in the router's webflash. On newer 800 series routers, like the 851 or 871, the name of the file is "sdmconfig-8xx.cfg" and is stored in the router's flash memory. Use the `copy flash:{config-name.cfg} start` to copy the default image to your router's startup configuration.

```
lab-r1> ena
lab-r1# copy flash:sdmconfig-8xx.cfg start      ! Cisco 851 or 871
lab-r1# copy webflash:ConfigExp.cfg start      ! Cisco SOHO91 or 831
```

Verifying Your Setup

In this final section, you'll learn some of the more common commands that can be used to gather information about your router's configuration. The Cisco IOS provides a wide range of `show` commands that allow you to verify your router's setup and display statistics.

Sometimes you may want to know basic information about an interface (such as which IP address your ISP has provided you via DHCP), while other times you may want to know how many matches a rule in an ACL has received or whether an interface is experiencing a high number of errors. Many of the details from the output of these commands are beyond the scope of this book so I won't go into much depth here. However, it's a good idea to be aware these commands exist and understand what sort of information they provide. I'll only touch upon the basics (you may spot some obvious problems).

Check Your Interfaces

You've already learned that the `show ip interface brief` command gives you a quick overview of your interface's IP address and status (whether it's up or down). There are two other interface-related commands that are useful and provide a lot more detail. The `show ip interface` command (abbreviated `sh ip int`) will display all of the configuration options that are configured on your interfaces, and the `show interface` command (abbreviated `sh int`) will display interface statistics. The following is an example of the brief command:

```
Router# show ip interface brief
```


Notice that the output from this command is very limited.

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet4	66.238.5.254	YES	NVRAM	up	up
Vlan1	10.10.10.1	YES	NVRAM	up	up

To see more detailed information about your interfaces, use the privileged EXEC mode command `show ip interface`. This command can be used on its own, to display information about all interfaces, or it can be used to display information about a specific interface like `vlan1` in the next example, which is only a snippet of the information (e.g., line status, IP address, NAT, firewall inspection) that the command provides. This command will also show you which ACLs are applied to an interface and in which direction they are filtering packets.

Router# `show ip interface vlan1`

The following is abbreviated output from the command:

```
Vlan1 is up, line protocol is up
  Internet address is 10.10.10.1/24
  Broadcast address is 255.255.255.255
  .
  .
  .
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Policy routing is disabled
  Network address translation is enabled, interface in domain inside
  BGP Policy Mapping is disabled
  Inbound inspection rule is IPFW
```

Use the privileged EXEC mode command `show interface` to look at packet statistics on an interface. Information gathered from this command could be useful in determining a hardware error on an interface. In particular, you want to look for a high number of input errors, output errors, or CRC errors. This information can be used to help determine if you have a faulty interface or bad cabling.

Router# `show interface fa4`

The following is a sample of WAN interface fa4:

```

FastEthernet4 is up, line protocol is up
  Hardware is PQICC_FEC, address is 0012.8032.a178 (bia 0012.8032.a178)
  Description: fa4 is WAN interface to ISP using a STATIC IP
  Internet address is 66.238.5.254 /27
  .
  .
  .
  5 minute input rate 45000 bits/sec, 96 packets/sec
  5 minute output rate 14000 bits/sec, 1 packets/sec
    7212591 packets input, 584921939 bytes, 0 no buffer
    Received 6951540 broadcasts, 0 runts, 0 giants, 0 throttles
    1 input errors, 0 CRC, 0 frame, 1 overrun, 0 ignored
    0 input packets with dribble condition detected
    255150 packets output, 158972122 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 babbles, 0 late collision, 0 deferred
    0 lost carrier, 0 no carrier
    0 output buffer failures, 0 output buffers swapped out
  .
  .
  .

```

Check NAT

There are several commands available that can be used to verify NAT. The privileged EXEC mode command `show ip nat statistics` (abbreviated `sh ip nat stat`) does what the command's name implies: it displays NAT statistics. You can see from the following sample output that there are quite a few active translations (147) on this router.

```

Total active translations: 147 (0 static, 147 dynamic; 147 extended)
Outside interfaces:
  FastEthernet4
Inside interfaces:
  Vlan1
Hits: 509796 Misses: 10941
CEF Translated packets: 462574, CEF Punted packets: 114608
Expired translations: 12201

```

Dynamic mappings:

-- Inside Source

[Id: 1] route-map nonat interface FastEthernet4 refcount 11

Queued Packets: 0

If you would like to see all of those translations, use the privileged EXEC mode command `show ip nat translations` (abbreviated `sh ip nat tran`). Notice in the following example that NAT is currently forwarding an e-mail message, using port 25, to your e-mail server at the private IP address 192.168.1.3. There is also a web server that is serving web pages from the private IP address 192.168.1.4 on HTTP port 80.

Pro	Inside global	Inside local	Outside local	Outside global
tcp	66.238.5.254:25	192.168.1.3:25	123.23.198.189:59462	123.23.198.189:59462
tcp	66.238.5.254:25	192.168.1.3:25	---	---
tcp	66.238.5.254:80	192.168.1.4:80	60.208.64.177:56840	60.208.64.177:56840
tcp	66.238.5.254:80	192.168.1.4:80	60.208.64.177:64877	60.208.64.177:64877
tcp	66.238.5.254:80	192.168.1.4:80	62.163.22.82:12319	62.163.22.82:12319

Check Your ACLs

There are several ways to display your ACLs. The `show run` command will provide you with a quick look at the ACLs, but if you use the privileged EXEC mode command `show access-list {acl-name}`, as in the command statement `show access-list VTY-ACL`, the output will display how many times a particular rule has been met. The Cisco IOS refers to these as matches.

Router# `sh access-list VTY-ACL`

You can see from the following output that there have been 16 matches to the permit rule in the ACL named “VTY-ACL”. This information comes in handy when you learn about performance tuning your ACLs in Chapter 5.

Standard IP access list VTY-ACL

10 permit 192.168.1.0, wildcard bits 0.0.0.255 (16 matches)

Check Your Firewall

The privileged EXEC mode command `show ip inspect interfaces` is a great tool for doing a quick security inspection of your router. It will display all of your firewall names and rules for all of your interfaces, as well as which ACLs are applied to those interfaces. Try it now on your router.

```
Router# sh ip inspect interfaces
```

Notice from the following output that all of the application rules of your CBAC firewall named “IPFW” are displayed, but the rules for your firewall ACL “IPFW-ACL” are not. The output only displays the name of the ACL, and in what direction it has been applied on the interface. Also note from this output that an outbound ACL on interface FastEthernet 4 has not been set as shown by the line that reads “Outbound access list is not set”. When it comes to securing your router, it’s just as important to know which ACLs have not been set, as it is to know which ones have been set.

Interface Configuration

Interface Vlan1

Inbound inspection rule is IPFW

cuseeme alert is on audit-trail is on timeout 3600

ftp alert is on audit-trail is on timeout 3600

rcmd alert is on audit-trail is on timeout 3600

realaudio alert is on audit-trail is on timeout 3600

smtp max-data 20000000 alert is on audit-trail is on timeout 3600

tftp alert is on audit-trail is on timeout 30

udp alert is on audit-trail is on timeout 15

tcp alert is on audit-trail is on timeout 3600

h323 alert is on audit-trail is on timeout 3600

Outgoing inspection rule is not set

Inbound access list is not set

Outgoing access list is not set

Interface FastEthernet4

Inbound inspection rule is not set

Outgoing inspection rule is not set

Inbound access list is IPFW-ACL

Outgoing access list is not set

Summary

You covered a lot of ground in this chapter. You’ve learned all of the steps to configure your router as an Internet gateway and firewall. You’ve learned how to erase your startup configuration file, configure your LAN and WAN interfaces, set up a DHCP server, and

configure and secure your router's VTY lines with an ACL. You've also learned how to configure NAT, set up a basic DMZ, configure an IOS Firewall, save your configuration, and restore your router's default configuration. Here's a brief recap of all of the concepts introduced in Chapter 2.

Erase the Startup Configuration

The startup configuration is the configuration file that loads into main memory when the router is started. Use the privileged EXEC mode command `erase start` to erase the startup configuration, as shown in the following example:

```
Router> ena
Router# erase start
```

Configure an IP Address on Your LAN Interface

You must be in interface configuration mode to assign an IP address to an interface. Use the `ip address` command to assign the address and mask. You can also use the `description` command to keep notes about the interface.

```
Router> ena
Router# config t
Router(config)# hostname lab-r1
lab-r1(config)# int vlan1
lab-r1(config-if)# descr LAN switch ports on inside interface
lab-r1(config-if)# ip address 192.168.1.1 255.255.255.0
lab-r1(config-if)# no shut
lab-r1(config-if)# exit
lab-r1(config)# exit
lab-r1#
```

Configure a DHCP Server

DHCP manages all of your host IP addresses and provides a variety of other IP information to your workstations. Use the `ip dhcp pool` command to name and enable a DHCP server on your router. In this example the DHCP pool name is "MYNET". The `network` command defines the network address that DHCP will use to assign host addresses, the `default-router` command defines the gateway address (which is your router), the `dns-server` command defines your DNS servers, and the `lease` command is used to determine how many days and hours your workstation may keep its address (24 hours by default). Finally, the `ip dhcp excluded-address` command defines the address range that will be excluded by DHCP for leases. These addresses are reserved for your static IP hosts, such as servers, printers, routers, and switches.

```
lab-r1> ena
lab-r1# config t
lab-r1(config)# ip dhcp pool MYNET
lab-r1(dhcp-config)# network 192.168.1.0 255.255.255.0
lab-r1(dhcp-config)# domain-name geekvenue.net           ! Your domain name
lab-r1(dhcp-config)# default-router 192.168.1.1          ! This router's addr.
lab-r1(dhcp-config)# dns-server 66.238.5.2 66.238.5.3    ! Provided by ISP
lab-r1(dhcp-config)# netbios-name-server 192.168.1.2     ! Your WINS
lab-r1(dhcp-config)# lease 0 2                          ! 0 Days & 2 Hours
lab-r1(dhcp-config)# exit
lab-r1(config)# ip dhcp excluded-address 192.168.1.1 192.168.1.25
```

Configure Telnet on Your Router

Use telnet as an alternative to configuring your router via the console port. It allows you to log in remotely using a telnet client application and configure your router over an IP network. You must set your privileged EXEC mode password and your telnet password before you can log in via the VTY line.

```
lab-r1> ena
lab-r1# config t
lab-r1(config)# enable secret cisco           ! privileged EXEC Mode Password
lab-r1(config)# line vty 0 4                  ! Telnet Virtual Terminal
lab-r1(config-line)# password cisco2         ! Telnet Password
lab-r1(config-line)# login
lab-r1(config-line)# exit
```

Secure VTY with an ACL

You should limit access to your router via VTY by means of an ACL. In the following example, the ACL named “VTY-ACL” only allows users on the local (private) network 192.168.1.0 to log in to the router using telnet. The privileged EXEC mode command `ip access-list standard {acl-name}` creates and names a standard ACL. The `permit` command defines which IP addresses will be allowed. Use the `access-class {acl-name}` in command to apply the ACL to inbound traffic on your VTY line (telnet).

```
lab-r1> ena
lab-r1# config t
lab-r1(config)# ip access-list standard VTY-ACL
lab-r1(config-std-nacl)# permit 192.168.1.0 0.0.0.255    ! All local hosts
lab-r1(config-std-nacl)# exit
lab-r1(config)# line vty 0 4
lab-r1(config-line)# access-class VTY-ACL in
```

Configure Your WAN Interface—Dynamic IP

You can configure your WAN interface to use a static IP address or use DHCP. Here is how to set up your WAN interface using DHCP and a nonauthenticated broadband connection, such as a cable modem. Use the interface configuration command `ip address dhcp client-id {interface}` to use a DHCP address on your WAN interface.

```
lab-r1> ena
lab-r1# config t
lab-r1(config)# ip domain-lookup                ! Enables DNS lookup
lab-r1(config)# int fa4
lab-r1(config-if)# descr WAN interface to ISP using DHCP
lab-r1(config-if)# ip address dhcp client-id FastEthernet4 !Enable DHCP Client
lab-r1(config-if)# duplex auto
lab-r1(config-if)# speed auto
lab-r1(config-if)# no shut
lab-r1(config-if)# exit
```

Configure Your WAN Interface—Static IP

Use the interface configuration command `ip address {ip-address} {subnet-mask}` to configure your router's WAN interface to use a static IP address given to you by your ISP. Use the configuration command `ip route 0.0.0.0 0.0.0.0 {gateway-address}` to enter your ISP-provided gateway address, the `ip domain-lookup` command to allow DNS lookups by the router, and the `ip name-server {name-server-addresses}` command to enter your ISP-provided DNS addresses.

```
lab-r1> ena
lab-r1# config t
lab-r1(config)# ip domain-lookup
lab-r1(config)# int fa4
lab-r1(config-if)# descr WAN interface to ISP using a STATIC IP
lab-r1(config-if)# ip address 66.238.5.254 255.255.255.0          ! From ISP
lab-r1(config-if)# duplex auto
lab-r1(config-if)# speed auto
lab-r1(config-if)# no shut
lab-r1(config-if)# exit
lab-r1(config)# ip route 0.0.0.0 0.0.0.0 66.238.5.1              ! From ISP
lab-r1(config)# ip name-server 66.238.5.2 66.238.5.3             ! From ISP
lab-r1(config)# exit
lab-r1#
```

Secure Your Interfaces

To help secure your router, it is highly recommended that you disable the following services on both your LAN and WAN interfaces:

```
lab-r1> ena
lab-r1# config t
lab-r1(config)# int fa4
lab-r1(config-if)# no ip unreachable
lab-r1(config-if)# no ip redirects
lab-r1(config-if)# no ip proxy-arp

lab-r1(config-if)# int vlan1
lab-r1(config-if)# no ip unreachable
lab-r1(config-if)# no ip redirects
lab-r1(config-if)# no ip proxy-arp
lab-r1(config-if)# exit
lab-r1(config)# exit
```

Configure NAT on Your Router

NAT allows your inside (private) IP hosts to access the Internet through your router using a single, public IP address. Create an extended ACL using the `ip access-list extended {acl-name}` command. Use the extended ACL `permit ip` statement to permit all of your hosts. Use the global configuration command `ip nat inside source list {acl-name} interface {interface} overload` to enable NAT on your WAN interface. Use the interface commands `ip nat inside` and `ip nat outside` to define which interface will be used for inside and which interface will be used for outside network address translation.

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# ip access-list extended NAT-ACL
lab-r1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 any      !All local hosts
lab-r1(config-ext-nacl)# exit
lab-r1(config)# ip nat inside source list NAT-ACL interface fa4 overload

lab-r1(config)# int vlan1                                     ! Your LAN Switch Port
lab-r1(config-if)# ip nat inside

lab-r1(config-if)# int fa4                                    ! Your WAN Port
lab-r1(config-if)# ip nat outside
```


Create an Advanced Firewall

Use the global configuration mode command `ip inspect name {firewall-name} {protocol or application-name}` to name your firewall and create firewall inspection rules. The Cisco CBAC firewall doesn't work with ICMP packets, so you need to add ping and traceroute rules to your WAN ACL if you want to allow those applications from your private network to the Internet. Apply the IPFW-ACL to your WAN interface.

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# ip inspect name IPFW tcp
lab-r1(config)# ip inspect name IPFW udp
lab-r1(config)# ip inspect name IPFW cuseeme
lab-r1(config)# ip inspect name IPFW ftp
lab-r1(config)# ip inspect name IPFW tftp
lab-r1(config)# ip inspect name IPFW rcmd
lab-r1(config)# ip inspect name IPFW realaudio
lab-r1(config)# ip inspect name IPFW smtp
lab-r1(config)# ip inspect name IPFW h323

lab-r1(config)# int fa4                                ! Your WAN Interface
lab-r1(config-if)# ip inspect IPFW out                  ! Apply CBAC to WAN Interface
lab-r1(config-if)# exit
lab-r1(config)# exit

lab-r1(config)# ip access-list extended IPFW-ACL        ! Create a Named ACL

! Allow outbound Ping and Traceroute
lab-r1(config-ext-nacl)# permit icmp any any administratively-prohibited
lab-r1(config-ext-nacl)# permit icmp any any echo-reply
lab-r1(config-ext-nacl)# permit icmp any any packet-too-big
lab-r1(config-ext-nacl)# permit icmp any any time-exceeded
lab-r1(config-ext-nacl)# permit icmp any any traceroute
lab-r1(config-ext-nacl)# permit gre any any             ! Allow PPTP
lab-r1(config-ext-nacl)# deny ip any any               ! Deny ALL Traffic
lab-r1(config-ext-nacl)# exit

lab-r1(config)# int fa4                                ! WAN Interface
lab-r1(config-if)# ip access-group IPFW-ACL in          ! ACL on WAN Interface
lab-r1(config-if)# exit
lab-r1(config)# exit
lab-r1#
```

Set Up a Basic DMZ

To set up a basic DMZ, create a firewall ACL on your WAN interface that allows desired traffic inbound from the Internet. Use extended ACL statement `permit tcp any host {wan-ip-address} eq {protocol or protocol-port-number}` to allow the desired traffic.

Use the global configuration command `ip nat inside source static tcp 192.168.1.3 25` to redirect that traffic to your e-mail server based on the TCP/IP port number.

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# ip access-list extended IPFW-ACL
lab-r1(config-ext-nacl)# permit tcp any host 66.238.5.254 eq smtp
lab-r1(config-ext-nacl)# permit icmp any any administratively-prohibited
lab-r1(config-ext-nacl)# permit icmp any any echo-reply
lab-r1(config-ext-nacl)# permit icmp any any packet-too-big
lab-r1(config-ext-nacl)# permit icmp any any time-exceeded
lab-r1(config-ext-nacl)# permit icmp any any traceroute
lab-r1(config-ext-nacl)# permit gre any any
lab-r1(config-ext-nacl)# deny ip any any
lab-r1(config-ext-nacl)# exit
lab-r1(config)# int fa4
lab-r1(config-if)# ip access-group IPFW-ACL in
lab-r1(config-if)# exit
lab-r1(config)# ip nat inside source static tcp 192.168.1.3 25 interface fa4 25
lab-r1(config)# exit
lab-r1#
```

Save Your Configuration

After making changes to your router's configuration, you save them using the `copy running-configuration startup-configuration` command (abbreviated `copy run start`).

```
lab-r1> ena
lab-r1# copy run start
Destination filename [startup-config]? {press ENTER}
```

Restore the Default Configuration

To restore your router's configuration back to the factory default, copy the default configuration file (stored in flash) to the startup-config and restart your router. Use the `sh flash` or `sh webflash` command to display the name of your router's configuration file.

```
lab-r1> ena
lab-r1# copy flash:sdmconfig-8xx.cfg start      ! Cisco 851 or 871
lab-r1# copy webflash:ConfigExp.cfg start      ! Cisco Soho91 or 831
```

Verify Your Setup

There are many ways to use the Cisco IOS `show` command to display configuration settings and statistics on your router. This information can be used as general information or used for performance tuning and troubleshooting router issues.

There are three ways to gather information about your interfaces:

`show ip interface brief`—Displays basic up/down status information about your interfaces as well as each interface's IP addresses.

`show ip interface {interface name}`—Displays configuration options that are enabled for an interface.

`show interface {interface name}`—Displays statistics about packets that have traveled through the interface.

The following two privileged EXEC mode commands can be used to verify that NAT is functioning on your router:

`show ip nat statistics`—Displays statistical information about translated packets.

`show ip nat translations`—Displays the state of your translation table (what is being translated by NAT).

The following privileged EXEC mode commands can be used to verify the ACLs running on your router:

`show running-configuration`—Displays what ACLs are assigned to which interfaces and in what direction they're applied.

`show access-list {acl-name}`—Displays an ACL and shows how many times each rule in the ACL has been matched.

`show ip inspect interfaces`—Gives a quick overview of the firewalls and ACLs configured on all your interfaces. This command does not display detailed information about the ACLs, only which ACLs are used and the direction they are filtering, but it does show detailed information about the protocols and applications configured in your firewalls.



Configuring DSL Using PPPoE

I would like to preface this chapter by stating that DSL is strange and mysterious voodoo. DSL takes Ethernet—a well-defined and documented standard—and squeezes it into a serial line protocol known as Point-to-Point Protocol (PPP). PPP has been used with old-fashioned analog modems for decades. DSL is a relative newcomer.

There have been a few different DSL technologies introduced over the years, and there will certainly be more to come. The first one that I worked with was Internet DSL (IDSL). The current DSL standards with which I'm familiar are Consumer-grade DSL (CDSL) and Asymmetric DSL (ADSL). CDSL offers a fast download speed (downstream speed) of about 1 Mbps but a much slower upload speed (upstream speed). ADSL also has a relatively slow upstream speed, compared to its downstream speed, as the word asymmetric in its name implies. However, ADSL is much faster and has a theoretical maximum downstream speed of 6.1 Mbps! Telephone companies (telcos) that offer DSL rely on the fact that most people download a lot more information than they upload and take advantage of this fact to use their own bandwidth more efficiently (even telcos have to pay for bandwidth).

When configuring your router, there are a couple of key differences between DSL and cable modem broadband. One is that DSL usually requires a username and password and needs at least 8 bytes of each data frame to carry PPP information. Whereas a cable modem requires a separate coaxial cable to carry data, sometimes voice, and of course, CNN Headline News simultaneously, DSL carries its voice and data signals simultaneously over existing copper telephone lines.

There are several reasons why you may want to use DSL, so it's a good idea to know something about it. One reason is sometimes DSL is the only service available at a commercial site and another is that it can often deliver bandwidth more consistently than cable. However, bandwidth reliability by both services can fluctuate, so your mileage may vary. More important, it's a good idea to know how to configure it on your Cisco router.

Introducing PPPoE

Point-to-Point Protocol over Ethernet (PPPoE) is the protocol used to configure an authenticated DSL connection to an Internet Service Provider (ISP), and it's more complicated

than setting up an Internet connection using a cable modem. For starters, PPPoE uses a *virtual* WAN interface (called a dialer) along with the *physical* WAN interface.

Note When using PPPoE, the virtual interface is where you configure outside NAT, your WAN ACLs and other WAN parameters that would normally be applied to the physical WAN interface.

DSL using PPPoE requires you to authenticate to your ISP with a username and password. There are two types of authentication: Challenge-Handshake Authentication Protocol (CHAP) and Password Authentication Protocol (PAP). CHAP is the most commonly used by ISPs and is more secure than PAP. PAP sends information in clear text, whereas CHAP uses encryption to keep your DSL password secure.

As mentioned earlier, PPPoE needs a little space to carry some overhead. PPPoE requires a minimum of 8 bytes from each data frame to carry PPP information. As a result you need to modify (reduce) the frame size on your dialer interface. This is commonly called the Maximum Transmission Unit (MTU) size.

Overview of the Steps

As mentioned earlier, configuring the Cisco PPPoE client on your router is more complicated than setting up a cable modem broadband connection. The following list walks you through the steps:

1. Collect information from your ISP.
2. Enable Virtual Private Dialup Networking (VPDN), create a VPDN dial group, and configure it to use PPPoE.
3. Prepare the physical WAN interface for PPPoE.
4. Configure the virtual WAN interface.
5. Configure NAT on the virtual WAN interface.
6. Set the default gateway to the virtual WAN interface.
7. Adjust the Maximum Segment Size (MSS) on the LAN interface.

The following sections will take a detailed look at each step in the process.

Collecting Information from Your ISP

Before you can configure your PPPoE client, you must gather some information from your ISP. Unlike setting up a cable modem, DSL using PPPoE requires authentication. Often the username is in the form of an e-mail address for your ISP (e.g., jason@acmeisp.com). You also need to determine by which protocol your ISP would like you to authenticate. It could be CHAP, PAP, or possibly both. (I've never actually known an ISP to use both, but I'm aware that some may so I've included it here as a possibility.) If you have any doubt about which authentication type to use, you can configure both as in the example in Listing 3-3 and your router will automatically authenticate using the protocol required.

If you request a static IP address, it usually ties to your authentication. In other words, it's a DHCP-assigned address that never changes because it's associated with your login information. As it relates to DSL, Cisco refers to this as a *negotiated* IP address.

Enabling Virtual Private Dialup Networking

It may seem odd, but PPPoE uses dialup networking even though it's a broadband connection on an Ethernet interface. Depending on your router's IOS version, you may need to enable a dialup networking group on your router. Cisco calls this a Virtual Private Dialup Network (VPDN) connection. For IOS releases earlier than 12.4, you need to execute the commands shown in Listing 3-1. For release 12.4 on, those commands are optional (and you can skip ahead to the section, "Preparing the Physical WAN Interface"), but executing them will cause no harm.

Listing 3-1. *How to Configure VPDN*

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# vpdn enable
lab-r1(config)# vpdn-group PPPOE
lab-r1(config-vpdn)# request-dialin
lab-r1(config-vpdn-req-in)# protocol pppoe
lab-r1(config-vpdn-req-in)# ^Z
lab-r1#
```

Step1: Enable VPDN on Your Router

Begin by enabling VPDN on your router with the global configuration mode command `vpdn enable`.

Step 2: Create a VPDN Group

Next, issue the `vpdn-group` command to create a VPDN dialer group. You can name the group anything you would like. The dialer group in Listing 3-1 has been named "PPPOE"

with the command `vpdn-group PPPoE`. Notice that this command places you in VPDN configuration mode and the prompt has changed to `(config-vpdn)#`.

Step 3: Request a VPDN Tunnel to Your ISP

Now issue the command `request-dialin`. This command configures your router to request a dialin VPDN tunnel to your ISP. The tunnel is used to forward packets to your ISP's Network Access Server (NAS). Notice that the prompt changes to `(config-vpdn-req-in)#` when configuring a dialin request. From this prompt, you can configure which protocol the dialin request will use.

Step 4: Set the VPDN Protocol to PPPoE

Next, configure the VPDN requested dialin to use PPPoE by issuing the command `protocol pppoe`.

Finally, you can exit all the way back to privileged EXEC mode by using the shortcut key-stroke `Ctrl+Z (^Z)`. You can also type `exit` three times to return to privileged EXEC mode.

Preparing the Physical WAN Interface

The PPPoE client on a Cisco router uses two WAN interfaces: a physical WAN interface, and a virtual WAN interface called a *dialer interface*. The physical WAN interface is the Ethernet WAN port on the back of your router (fa4 in Listing 3-2). The commands in Listing 3-2 prepare the physical WAN interface for PPPoE:

Listing 3-2. How to Configure the Physical WAN Interface

```
lab-r1> ena
lab-r1# config t
lab-r1(config)# interface fa4
lab-r1(config-if)# descr Physical WAN Interface used by DSL (Dialer1)
lab-r1(config-if)# no ip address
lab-r1(config-if)# pppoe enable group global
lab-r1(config-if)# pppoe-client dial-pool-number 1
lab-r1(config-if)# no shut
lab-r1(config-if)# exit
lab-r1(config)#
```

Step 1: Start Interface Configuration Mode

Begin by entering interface configuration mode for your physical WAN interface. As shown in Listing 3-2, you use the global configuration command `interface fa4` to configure the physical WAN interface FastEthernet 4 (fa4).

Step 2: Add a Description to Your Interface

If you would like to add a description, you can use the `description` command (abbreviated `descr`). You may wish to include the DSL circuit number in the interface description. This is often the nine-digit telephone number for the telephone line that the circuit is using.

Step 3: Remove Any Assigned IP Address

Remove any assigned IP address using the `no ip address` command. You do not want an IP address configured on the physical WAN interface because DSL using PPPoE requires assigning the WAN IP address to the virtual WAN interface instead.

Step 4: Enable PPPoE on the Physical Interface

The command `pppoe enable group global`, configured on the physical WAN interface, will enable PPPoE sessions using Cisco's global parameters, which communicates with your ISP's DSL equipment. That equipment could be a NAS, a Digital Subscriber Line Access Multiplexer (DSLAM) or some other DSL technology. In IOS version 12.4, this single `pppoe` statement replaces all the VPDN commands in Listing 3-1.

Note PPPoE on Cisco routers works with ECI-high focus, Alcatel 7300, and Lucent Stinger DSLAM chipsets that are likely used by every ISP in the United States and abroad, making compatibility between your router and your ISP's DSL equipment virtually guaranteed.

Step 5: Enable the Cisco PPPoE Client

Use the command `pppoe-client dial-pool-number 1` to enable the Cisco PPPoE dialer client on the interface. I like to think of the client as a dial script used to bring up the DSL connection. The number "1" identifies this as pool number one, so that it can be associated with the virtual WAN interface, tying the physical WAN interface to the virtual WAN interface. This allows them to work together.

Step 6: Bring Up the Interface

Finally, bring up the interface using the `no shutdown` command (abbreviated `no shut`), then type `exit` to leave interface configuration mode.

Configuring the Virtual WAN Interface

Now that you've configured the physical WAN interface, you need to configure the virtual WAN interface. The virtual WAN interface (dialer 1 in Listing 3-3) is where you apply normal WAN attributes such as outside NAT, WAN ACLs like "IPFW-ACL" (introduced in

Chapter 2), and PPP authentication (either CHAP or PAP). This is also where you set the MTU size for data frames on the interface.

Listing 3-3. *How to Configure the Virtual WAN Dialer Interface*

```
lab-r1> ena
lab-r1# config t
lab-r1(config)# interface dialer 1
lab-r1(config-if)# ip mtu 1492
lab-r1(config-if)# dialer pool 1
lab-r1(config-if)# dialer-group 1
lab-r1(config-if)# ip address negotiated
lab-r1(config-if)# encapsulation ppp
lab-r1(config-if)# ppp authentication chap pap callin
lab-r1(config-if)# ppp chap hostname jason@acmeisp.com
lab-r1(config-if)# ppp chap password MyISPpassword
lab-r1(config-if)# ppp pap sent-user jason@acmeisp.com password MyISPpassword
lab-r1(config-if)# no shut
lab-r1(config-if)# exit
lab-r1(config)#
```

Step 1: Start Interface Configuration Mode

Start interface configuration mode using the global configuration mode command `interface dialer 1` (abbreviated `int dial 1`). This allows you to configure your virtual interface. Your treatment of a virtual interface is similar to that of a physical interface. Virtual interfaces can be shutdown, assigned IP addresses, descriptions, etc.

Step 2: Set the Dialer's MTU Size

Because DSL carries the PPP protocol inside Ethernet frames, you need to reserve some space in the data frames on the dialer interface for the protocol. This is often referred to as *PPP overhead*. Typically DSL requires 8 bytes for PPP overhead; a normal Ethernet frame contains 1,500 bytes. To accommodate PPP, you reduce the standard frame size (also known as the MTU) by 8 bytes, making it 1,492. Use the command `ip mtu 1492` to set the frame size to 1,492 bytes. Some ISPs require an even smaller MTU size, depending on their implementation of DSL. Also, you may find that you can achieve better performance by reducing the MTU more than your ISP requires. In general, though, follow your ISP's recommendations.

Step 3: Set the Dialer Pool

Use the commands `dialer pool 1` and `dialer-group 1` to associate the virtual dialer interface with the PPPoE client configured on the physical interface (fa4 in this example). The reason they are associated is because the dialer pool number configured on interface fa4 is also set to "1."

Step 4: Request a DHCP Address Using PPPoE

Even if you have requested a static IP address from your ISP, chances are it will be assigned to your router based on your authentication. Because you are using PPPoE, you will receive your static IP address based on your PPPoE username. Therefore, when configuring PPPoE on your router, you use the command `ip address negotiated` to request an IP address from your ISP.

Step 5: Set PPP Encapsulation on the Interface

Before you can use PPPoE, you must configure the router's virtual WAN interface to encapsulate data using PPP. PPP is the protocol used to transport IP traffic over point-to-point links. It is responsible for aspects, such as link quality, error detection, and data compression negotiation. Use the command `encapsulation ppp` (abbreviated `encap ppp`) to enable the PPP protocol on your virtual WAN interface.

Step 6: Configure CHAP and PAP Authentication

DSL using PPPoE requires you to authenticate to your ISP before it will bring up your DSL circuit. The ISP will usually require either CHAP or PAP authentication. In this example, you will be configuring both CHAP and PAP on your router.

Tip Your ISP's help desk may not be able to tell you what form of authentication they use (it happens all the time). Configuring both authentication types will solve this problem. Also, many ISPs use an e-mail address as the username for authentication.

The command `ppp authentication chap pap callin` enables both CHAP and PAP authentication protocols on your interface. The IOS commands used to authenticate using PAP are different from the commands used to authenticate using CHAP.

There are two steps to configuring CHAP:

1. `ppp chap hostname jason@acmeisp.com`—uses the username assigned to you by your ISP, to authenticate using CHAP. In this example, ACME ISP has assigned you the login name “jason@acmeisp.com”.
2. `ppp chap password MyISPpassword`—sets the CHAP password to “MyISPpassword”.

That's all there is to CHAP authentication. PAP authentication is simpler still. Only one line is needed to configure PAP authentication. The following command sets the username to “jason@acmeisp.com” and the password to “MyISPpassword”:

```
ppp pap sent-user jason@acmeisp.com password MyISPpassword
```

Configuring NAT on the Virtual WAN Interface

NAT, which was introduced in Chapter 2, is configured differently when using PPPoE and DSL. The difference is that NAT is configured on the virtual WAN interface (dialer 1) and not the physical WAN interface. Listing 3-4 shows the necessary commands to configure NAT on your virtual WAN interface.

Listing 3-4. *How to Configure NAT on the Dialer Interface*

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# ip access-list extended NAT-ACL
lab-r1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 any
lab-r1(config-ext-nacl)# exit

lab-r1(config)# ip nat inside source list NAT-ACL int dialer 1 overload

lab-r1(config)# int vlan1
lab-r1(config-if)# ip nat inside
lab-r1(config-if)# int dialer 1
lab-r1(config-if)# ip nat outside
```

Step 1: Create and Name an Extended ACL for NAT

The first step is to create an ACL that allows all of your local hosts to use NAT. The global configuration command `ip access-list extended NAT-ACL` creates an extended access list named “NAT-ACL.” You can confirm that you are working with an extended access control list because the prompt changes to `(config-ext-nacl)`.

Step 2: Create an Access Control List Rule

Only one rule is required in the “NAT-ACL” to permit all of your LAN hosts access to the Internet via NAT. The extended access rule `permit ip 192.168.1.0 0.0.0.255 any` permits Internet access to all your private IP hosts on network 192.168.1.0.

Step 3: Configure Inside Address Translation

The global configuration command `ip nat inside source list NAT-ACL int dialer 1 overload` tells NAT how to translate addresses. In this case, you’re instructing NAT to translate your inside addresses using your ACL “NAT-ACL” then “overload” them on to the virtual WAN interface “dialer 1.”

Step 4: Apply NAT to Your Interfaces

When using PPPoE, the virtual WAN interface is where you will configure outside NAT. For NAT to work properly, it’s important that it knows which interface is the inside (private) interface, and which is the outside (public) interface. As shown in Listing 3-4, switch

to interface configuration mode and issue the command `ip nat inside` on your private LAN interface (vlan1), and the command `ip nat outside` on your public WAN interface (dialer 1)—not the physical interface.

Setting the Default Gateway

When using PPPoE, the router's default gateway (the gateway provided by your ISP) is set differently than when using a cable modem. The difference is that you configure the default gateway to use your virtual WAN interface (dialer1), not an IP address. Rather than routing packets from your private network to your ISP using IP routing, PPPoE instead forwards packets destined for the Internet to your ISP through the VPDN tunnel. From there, the packets use the IP protocol to route to the Internet. Listing 3-5 shows the commands to set the default gateway.

Listing 3-5. *How to Set the Default Route to the Dialer Interface*

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# ip route 0.0.0.0 0.0.0.0 dialer 1
lab-r1(config)# exit
```

As you can see from the command `ip route 0.0.0.0 0.0.0.0 dialer 1`, the default route has been set to the virtual dialer interface (dialer 1), rather than using the physical WAN interface or an IP address.

Adjusting the MSS on the LAN Interface

Remember I stated that DSL is strange and mysterious voodoo? The need to change the MTU size on your WAN interface is one reason. Another is the need to change the maximum segment size (MSS) on your LAN interface.

It works like this; All Ethernet devices have an MTU size (the default being 1,500 bytes). If the MTU size on your workstation PCs is not changed to match the decreased MTU size on the WAN interface—which is required for DSL using PPPoE—the router in between the PCs and the Internet will drop many of the packets received from the Internet before they reach the PCs. In this situation, users browsing the Internet may receive only partial content from a web page: perhaps only the text but no images, or some text and some images. When this happens, rest assured the users will call you quicker than a flash of light to complain!

Decreasing the MSS on the LAN interface of your router solves the problem, without having to configure the MTU size on all your workstations. It's quite an elegant solution because you only need to configure one interface on your router. The commands in Listing 3-6 show how to set the MSS on your LAN interface.

Listing 3-6. *How to Set the MSS on the LAN Interface*

```
lab-r1# config t
lab-r1(config)# interface vlan1
lab-r1(config-if)# ip tcp adjust-mss 1412
```

Use the interface command `ip tcp adjust-mss {size}` to set the MSS on the LAN interface of your router. A setting of 500 is as small as you can go, 1,460 is the maximum. Note in Listing 3-6, I recommended that you set the MSS size to 1,412 for best results, but you can adjust it up or down as needed.

General Troubleshooting

You've just completed all the configuration steps necessary for a PPPoE connection to your ISP, plugged in your DSL modem, powered it all up—and you don't have a connection to the Internet! I have firsthand knowledge of how frustrating this can be (having been down this road many times before), and I sympathize with anyone in this situation. The best thing to do now is take a short break, collect your thoughts, and come back refreshed and ready to begin troubleshooting the problem. Most of the time it's a simple mistake in the configuration, an unplugged cable, or some other simple and annoying little thing, but even little things can stop you dead in your tracks. In this section, I'm going to explain some of the less technical methods you can use to troubleshoot a non-functional DSL circuit. In the section, "Using the Cisco Debugger," I'll explain the more technical aspects of troubleshooting a DSL circuit on your Cisco router.

Check That the DSL Circuit Has Been Activated

Many ISPs require that you telephone them and request that they activate your DSL circuit before it can be used. If you're unsure, call your ISP and ask if the circuit is activated. They will usually ask you for the circuit ID, which is often the nine-digit telephone number for the line where the circuit resides.

Tip Because DSL signals can create a lot of noise on the telephone line they share—which is very distracting when it's a voice line—many ISPs prefer to use your company's fax line for their DSL circuit. Therefore, the circuit number is often the telephone number of your company fax line.

Check Your Username and Password and MTU

I know this seems obvious, but it happens all the time. An improperly configured username or password will prevent your router from successfully bringing up the DSL circuit.

Be sure to check that you are using the correct credentials for both CHAP and PAP in your IOS configuration. Remember, passwords are case sensitive. Also check the MTU size configured on your router. It should be configured on the virtual WAN interface (dialer 1). An MTU size of 1,492 is common, but your ISP may require a smaller size.

Verify That the Circuit Is Functional

The simplest way to test whether a DSL circuit is functional is by using a PC. If the PPPoE software on your laptop (which is very simple to set up) cannot bring up the connection, then it may be something wrong with the circuit itself or the login credentials supplied by your ISP. In either case, your ISP's help desk will want to step you through the DSL configuration process over the phone. It's not very likely they are going to help you with your Cisco router configuration, but they will gladly tell you how to configure Windows to work with their DSL. If the circuit works with Windows, then it's functional.

Print a Copy of Your Router's Configuration

A paper copy of the `show running-config` command output (abbreviated `show run`) can be a handy tool for troubleshooting all aspects of an IOS configuration. I like to use a pen and tick off each configuration setting to ensure that I've not missed anything. If you're an IT consultant, you can save a copy of your client's configuration and refer to it later if they're having problems or if they would like you to modify the configuration of their router. I'm a big fan of well-documented networks, whether they are servers, switches, or routers, and you should be, too! It's an important part of good network administration.

Use the IOS to Troubleshoot PPPoE

Use the command `show ip interface brief` to verify that the virtual WAN interface has received an IP address from your ISP. If it has received an IP address, then you know that PPPoE is operational.

Tip Many IOS `show` commands, like the `show ip interface brief` command, will work in either user EXEC mode or privileged EXEC mode.

If the PPPoE client was able to establish a connection with the ISP and you have an IP address on interface dialer 1, but you are still unable to browse a web site from a client, then the problem may be a misconfigured ACL, a DNS configuration problem, or an improperly configured default gateway on your router. Remember, the default gateway should be set to the virtual interface (dialer 1). The problem could also be that one of your

interfaces is shut down. If so, use the interface configuration mode command `no shutdown` (abbreviated `no shut`) to bring up the interface.

Another possibility is your NAT configuration. You should verify that your NAT-ACL is configured properly, the `ip nat inside` command is applied to your LAN interface, and the `ip nat outside` command is applied to your virtual WAN interface (dialer 1).

Using the Cisco Debugger

Okay, you've tried all the standard troubleshooting techniques and your configuration looks good, but your DSL circuit still isn't up and running. Now is the time to pull out the big gun—the Cisco IOS debugger!

The Cisco IOS has an extremely comprehensive debugger. This is a fabulous tool, but it provides so much information that it can be intimidating to new users and seasoned ones alike. Not to worry! In this section, I'll explain everything you need to know.

When using the IOS debugger to troubleshoot DSL and PPPoE, there are three main issues you need to look for and debug:

- Are you receiving a PPPoE response?
- Are PPP and LCP coming up properly?
- Are you successfully authenticating?

The first thing you need to do is enable logging on your router. Cisco provides several ways to log and view debugging information on a router. Log information can be sent to a server, called a *syslog* server, where the information can be viewed, or it can be logged to memory on the router itself and viewed from the IOS. You are going to enable *buffered logging* on your router. Once enabled, the log messages generated by the debugger will be buffered (stored) using some of the router's memory.

Enable Buffered Logging

The commands shown in Listing 3-7 will enable buffered logging. This means the logger will reserve some of your router's main memory to store logged events.

Listing 3-7. *How to Enable Buffered Logging*

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# no logging console
lab-r1(config)# no logging monitor
lab-r1(config)# logging buffered 4096 debugging
```

```
lab-r1(config)# exit
lab-r1#
```

Step 1: Disable Console and Monitor Logging

Use the global configuration commands `no logging console` and `no logging monitor` to prevent logging messages from popping up on the screen while logged on to the router via the console port or a VTY line. I find it highly distracting when information randomly appears on the screen, so I always disable these features before beginning a debugging session.

Step 2: Enable Buffered Logging

The global configuration command `logging buffered 4096 debugging` starts the logger, reserves 4,096 bytes of memory (4 KB) for log data, and sets the logging level to debugging for log events. The Cisco logger provides other logging levels, but debugging will provide you with the most information and is needed for troubleshooting PPPoE.

Check for PPPoE Response

When troubleshooting PPPoE, the first thing to check for is a successful PPPoE negotiation. This comes in the form of a PPPoE Active Discovery Initiation (PADI) and a PPPoE Active Discovery Offer (PADO). When the Cisco PPPoE client is enabled on a router, it will send a PADI out the router's WAN interface to your ISP and your router should receive a PADO back from your ISP's DSL equipment.

To check for a PPPoE response from your ISP, use the `debug pppoe events` command and check your log for PADI and PADO events.

Step 1: Debug PPPoE Events

Before you can begin any logging and debugging, you must specify the conditions you want to debug. In this case, it's PPPoE events. Listing 3-8 uses the privileged EXEC mode command `debug pppoe events` to start the debugger so that you can check for the success or failure of the initial PPPoE negotiation.

Listing 3-8. *How to Debug PPPoE Negotiation*

```
lab-r1> enable
lab-r1# debug pppoe events
```

Step 2: Check the Log

After you configure the debugger to start debugging events, use the privileged EXEC mode command `show log` to display the information stored in your router's logging buffer.

```
lab-r1# show log
```


The following output provided by the `show log` command indicates a problem:

```
*Jan 18 12:22:45.719: Sending PADI: Interface = FastEthernet4
*Jan 18 12:22:45.719: padi timer expired
```

A PADI has been sent by your router out the WAN interface `fa4`, as indicated by the log event `Sending PADI: Interface = FastEthernet4` but has *timed out* waiting for your ISP to respond. This is evident by the log event.

```
*Jan 18 12:22:45.719: padi timer expired
```

An expired PADI timer could mean your WAN interface is not plugged into the DSL modem, your DSL modem is not plugged into the phone line, the DSL modem has a problem, or the DSL circuit is not activated or functional. In this situation, look for something that is not plugged in or is plugged in incorrectly. Check that you're using the correct cables. Check that your DSL modem is turned on. If all that looks good, then contact your ISP to ensure that the circuit has been activated and correctly provisioned.

Let's have a look at some other debug PPPoE event logs. In the following output, the router has successfully negotiated a PADI with the ISP and is functioning properly:

```
*Jan 18 11:58:15.243: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet4,
changed state to up
*Jan 18 11:58:16.255: PPPoE: we've got our pado and the pado timer went off
*Jan 18 11:58:16.255: OUT PADR from PPPoE Session
*Jan 18 11:58:16.551: PPPoE 7833: I PADS R:0030.8822.f34d L:0012.8032.a178 Et1
*Jan 18 11:58:16.551: IN PADS from PPPoE Session
*Jan 18 11:58:16.559: %DIALER-6-BIND: Interface Vi2 bound to profile Di1
*Jan 18 11:58:16.563: PPPoE: Virtual Access interface obtained.
*Jan 18 11:58:16.563: PPPoE : encap string prepared
*Jan 18 11:58:16.563: [0]PPPoE 7833: data path set to Virtual Access
```

Your router's PADI received a PADO from your ISP and a virtual access interface was set up (bound) to your dialer interface. You will know it was successful because of the log event that reads

```
*Jan 18 11:58:16.255:  PPPoE: we've got our pado and the pado timer went off
```

I just love it when the PADO timer goes off! This tells you the DSL modem is set up correctly and your ISP has activated the DSL circuit. However, it does not indicate that the circuit is up and functional, only that a circuit appears to be available.

Now that you've collected that bit of information, you can stop debugging PPPoE events by using the privileged EXEC mode command `no debug pppoe events`.

```
lab-r1# no debug pppoe events
```

Debug the PPP

There are two components you need to debug when troubleshooting PPP. The first is *PPP negotiation*. This is where the hardware of both the router and the ISP decide how best to communicate with each other based on the rules defined by the protocol itself. The second part is the *PPP authentication*, which is the username and password required for DSL using the PPPoE protocol. Here are the steps to debug PPP negotiation and authentication.

Step 1: Debug PPP Negotiation

In this step, you will check for Layer 2 connectivity between the router and your ISP. I won't go into the details of what Layer 2 is, but you should know that Cisco uses the Link Control Protocol (LCP) to set up a data link layer connection from your router to the ISP and that prior to the PPPoE authentication, you need to have an *open LCP link state*.

The debug `ppp negotiation` command in Listing 3-9 configures the debugger to log information that can help you troubleshoot problems with PPP's initial negotiation and with the link state.

Listing 3-9. How to Debug PPP Negotiation

```
lab-r1> enable
lab-r1# debug ppp negotiation
```

Step 2: Check the Log

Use the privileged EXEC mode command `show log` to display the information in your router's log buffer.

```
lab-r1# show log
```

The following output indicates a *successful* PPP negotiation because the LCP link state is open, as indicated by the log line that reads `LCP: State is Open`.

```
*Jan 18 12:01:24.047: Vi2 LCP: I CONFREQ [ACKrcvd] id 115 len 14
*Jan 18 12:01:24.047: Vi2 LCP:   AuthProto PAP (0x0304C023)
*Jan 18 12:01:24.047: Vi2 LCP:   MagicNumber 0x7B04C481 (0x05067B04C481)
*Jan 18 12:01:24.047: Vi2 LCP: O CONFACK [ACKrcvd] id 115 len 14
*Jan 18 12:01:24.051: Vi2 LCP:   AuthProto PAP (0x0304C023)
*Jan 18 12:01:24.051: Vi2 LCP:   MagicNumber 0x7B04C481 (0x05067B04C481)
*Jan 18 12:01:24.051: Vi2 LCP: State is Open
```

Let's take a moment to decipher some of the information in this log. The log event that reads

```
*Jan 18 12:01:24.047: Vi2 LCP: I CONFREQ [ACKrcvd] id 115 len 14
```

informs you that a configuration request was sent to you from your ISP because of the statement CONFREQ and the I indicating it was an *inbound* request. The line that reads

```
*Jan 18 12:01:24.047: Vi2 LCP: O CONFACK [ACKrcvd] id 115 len 14
```

indicates a successful acknowledgment of that request because of the statement CONFACK. The O indicates the acknowledgment was sent *outbound* (from you to your ISP). However, the most important information is the line that reads

```
*Jan 18 12:01:24.051: Vi2 LCP: State is Open
```

which indicates that PPP has properly negotiated a link from your router to your ISP. If the LCP link state was *closed*, that may indicate a possible hardware problem, such as your router's WAN interface, a DSL modem, or the actual DSL line.

Once you are finished debugging PPP negotiation, don't forget to terminate debugging with the privileged EXEC mode command `no debug pppoe negotiation`.

```
lab-r1# no debug ppp negotiation
```

Step 3: Debug PPP Authentication

The last thing to check for is authentication. Basically, you're checking to determine if the router is configured with the correct username and password. It's worth noting that this is often the most commonly misconfigured parameter.

Tip Usernames and passwords are always case sensitive.

Begin by configuring the debugger. Listing 3-10 uses the privileged EXEC mode command `debug ppp authentication` to start debugging and logging information related to PPP authentication.

Listing 3-10. How to Debug PPP Authentication

```
lab-r1> enable
lab-r1# debug ppp authentication
```

Step 4: Check the Log

Now use the privileged EXEC mode command `show log` to display the information in your router's log buffer.

```
lab-r1# show log
```

The following output indicates that you have an authentication failure, most likely caused from an improperly configured username or password. As you can see from the output, this router is attempting to authenticate using PAP. The username being sent is `jason@acmeisp.com` and your ISP is sending back an AUTH-NAK. The NAK is a *negative* acknowledgment that alerts us when a username or password is incorrect.

```
*Jan 18 12:03:44.151: %LINK-3-UPDOWN: Interface Virtual-Access2, changed state to up
*Jan 18 12:03:44.259: Vi2 PPP: No authorization without authentication
*Jan 18 12:03:44.263: Vi2 PAP: Using hostname from interface PAP
*Jan 18 12:03:44.263: Vi2 PAP: Using password from interface PAP
*Jan 18 12:03:44.263: Vi2 PAP: 0 AUTH-REQ id 1 len 40 from "jason@acmeisp.com"
*Jan 18 12:03:44.547: Vi2 PAP: I AUTH-NAK id 1 len 5
*Jan 18 12:03:45.139: %DIALER-6-UNBIND: Interface Vi2 unbound from profile Di1
*Jan 18 12:03:45.143: %LINK-3-UPDOWN: Interface Virtual-Access2, changed state to ➡
down
```

Here's the output from another debug ppp authentication command; this time the authentication is successful. This output indicates that the router has successfully authenticated using PAP to the ISP. We know this because of the AUTH-ACK (authentication acknowledgment) log event that displays in the log just after the username jason@acmeisp.com was sent.

```
*Jan 18 12:13:06.927: Vi2 PPP: No authorization without authentication
*Jan 18 12:13:06.927: Vi2 PAP: Using hostname from interface PAP
*Jan 18 12:13:06.927: Vi2 PAP: Using password from interface PAP
*Jan 18 12:13:06.927: Vi2 PAP: O AUTH-REQ id 1 len 39 from "jason@acmeisp.com"
*Jan 18 12:13:07.195: Vi2 PAP: I AUTH-ACK id 1 len 5
*Jan 18 12:13:08.243: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-
Access2, changed state to up
```

Stop Debugging and Logging

Now that you've finished logging and debugging, it's a good idea to stop the debugger and logger, as debugging can consume a lot of CPU cycles and memory. Listing 3-11 displays the commands to stop debugging.

Listing 3-11. How to Stop the Debugger and Logger

```
lab-r1> enable
lab-r1# undebug all

lab-r1# clear log
Clear logging buffer [confirm] <press ENTER>

lab-r1# no logging buffered 4096 debugging
```

Step 1: Stop All Debugging

The following example uses the privileged EXEC mode command undebug all to terminate any debugging running on the router. It's a quick way to ensure you discontinue all debugging.

```
lab-r1> enable
lab-r1# undebug all
```

Step 2: Clear the Log

Use the privileged EXEC mode command clear log to clear all information from the logging buffer.

```
lab-r1# clear log
```

Confirm the action at the following prompt by pressing the ENTER key.

```
Clear logging buffer [confirm]
```

Step 3: Stop Buffered Logging

Use the privileged EXEC mode command `no logging buffered 4096 debugging` to discontinue the buffered logger and free the memory it's using.

A Word About ISPs

You've carefully used all of the troubleshooting techniques provided in this chapter and you suspect the problem is with your ISP. The next challenge is getting your ISP to acknowledge that it may have a problem. The first thing you need to do is call its technical support department on the telephone. If you're a residential customer, you can expect to wait 45 minutes or more before you reach a warm body. If you're a business customer, your ISP should offer expedited phone support, but I've found that you may still wait a long time before they provide you with any service. The word "service" is used quite loosely by ISPs these days. In fact, ISPs are probably the most frustrating aspect of configuring and troubleshooting a router connection. Ten years ago, you could call your ISP, ask to speak to the Network Operations Center (NOC), and be transferred to people who actually knew the nuts and bolts of their complex network. They would help you with all aspects of setting up and troubleshooting your circuit, including DNS setup, IP addresses, and anything else you needed help with. They talked the talk and walked the walk. It was great.

Today, Internet providers are more like corporate computer superstores. You can find a lot of people there who can help you with the basics, but if you need useful answers to complex questions, an expert is nowhere to be found.

It's possible you'll find the ISP technical support person is a grumpy curmudgeon who is more interested in getting you off the phone than solving your problem. Not all of them are like this, of course; but I've dealt with my share of them who are. They're very busy people who spend most of their time dealing with unknowledgeable users, and that can have the unfortunate effect of making them impatient or condescending (especially if you catch them at the end of their work day). However, no matter how bad they seem, you need to be nice to them. Let's face it, when you're having problems with your Internet connection, they have you over a barrel, so you may as well buck up and play nice.

The bottom line is this—don't let your ego (or theirs) get in the way of solving a problem for you or your client. Even if you think the person on the phone is dumber than a box of pet rocks, you need them. Who knows, they may be smarter than you think.

The best scenario is to find a contact person you like and trust (hang on to this person). A good contact at your ISP, who has your best interests in mind, is worth their weight in gold. Better yet, if you're lucky enough to find a technical support person at your ISP who understands Cisco routers, get their name and telephone number and store it with your router documentation!

Summary

This chapter covered the basics of DSL technology and the PPPoE. It covered all the commands necessary to configure PPPoE and PPPoE authentication. Basic DSL and PPPoE troubleshooting was covered as well as advanced PPPoE troubleshooting using the Cisco IOS Debugger.

Because most of the material regarding basic router setup (such as configuring your router's hostname, LAN interfaces, firewalls, etc.) was covered in Chapter 2, this chapter covered only the Cisco CLI commands specific to setting up a PPPoE connection.

What You Need from Your ISP

To configure PPPoE, you need the following information from your ISP: a username, a password, and the MTU size for the WAN interface.

Enable VPDN and Create a Dial Group (If Necessary)

PPPoE uses dialup networking; therefore, you may have to configure your router to use Cisco's VPDN. However, in recent versions of the IOS (12.4) these commands are not necessary.

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# vpdn enable
lab-r1(config)# vpdn-group PPPOE ! You can use any group name
lab-r1(config-vpdn)# request-dialin
lab-r1(config-vpdn-req-in)# protocol pppoe
lab-r1(config-vpdn-req-in)# ^Z
lab-r1#
```

Prepare the Physical WAN Interface

Enable PPPoE and the Cisco PPPoE client on the physical WAN interface.

```
lab-r1# config t
lab-r1(config)# interface fa4
lab-r1(config-if)# descr Physical WAN Interface used for DSL
lab-r1(config-if)# no ip address
lab-r1(config-if)# pppoe enable group global
lab-r1(config-if)# pppoe-client dial-pool-number 1
lab-r1(config-if)# no shut
lab-r1(config-if)# exit
lab-r1(config)#
```

Configure the Virtual WAN Interface

You must set the MTU size as per your ISP's requirements. You need to enable PPP encapsulation, and either CHAP, PAP or both authentication methods (as in the following example).

```
lab-r1(config)# interface dialer 1
lab-r1(config-if)# ip mtu 1492                                ! Provided by ISP
lab-r1(config-if)# dialer pool 1
lab-r1(config-if)# dialer-group 1
lab-r1(config-if)# ip address negotiated                      ! DHCP using PPPoE
lab-r1(config-if)# encapsulation ppp
lab-r1(config-if)# ppp authentication chap pap callin
lab-r1(config-if)# ppp chap hostname jason@acmeisp.com
lab-r1(config-if)# ppp chap password MyISPpassword
lab-r1(config-if)# ppp pap sent-user jason@acmeisp.com pass MyISPpassword
lab-r1(config-if)# no shut
lab-r1(config-if)# exit
lab-r1(config)#
```

Configure NAT on the Virtual WAN Interface (Dialer 1)

When using PPPoE, you must configure outside NAT on the virtual WAN interface.

```
lab-r1(config)# ip access-list extended NAT-ACL
lab-r1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 any  ! LAN hosts
lab-r1(config-ext-nacl)# exit
lab-r1(config)# ip nat inside source list NAT-ACL int dialer 1 overload
```



```
lab-r1(config)# int vlan1                ! LAN Interface
lab-r1(config-if)# ip nat inside
lab-r1(config-if)# int dialer 1          ! Virtual WAN Interface
lab-r1(config-if)# ip nat outside
lab-r1(config-if)# exit
```

Assign the Default Gateway to Use the Virtual WAN Interface

Set the router's default gateway address to the virtual WAN interface (dialer 1).

```
lab-r1(config)# ip route 0.0.0.0 0.0.0.0 dialer 1
```

Adjust the MSS on the LAN Interface

Setting MSS ensures that the router doesn't drop data on the LAN interface due to the decreased MTU size set on the WAN interface.

```
lab-r1(config)# interface vlan1
lab-r1(config-if)# ip tcp adjust-mss 1412
```

Troubleshooting

Check your IOS configuration, usernames, and password. Check that everything is plugged in correctly and that you are using the correct cables. If it all looks good, you can use the IOS debugger to log events that will help you verify the DSL circuit.

You can use this space for notes and to record information about your DSL circuit. Refer to this information later when you need to contact your ISP.

ISP NAME: _____

ISP PHONE NUMBER: _____

USERNAME: _____

CONTACT NAME: _____

PASSWORD: _____

STATIC IP: _____

SUBNET MASK: _____

MTU SIZE: _____

CIRCUIT NO.: _____

NOTES:



Configuring a VPN Using IPSec

A *Virtual Private Network* (VPN) is an inexpensive method that securely joins two or more private networks together using a publicly accessible network. The Internet is usually the public network, and the private networks are often a corporate office and a branch office, or a personal network used by a telecommuter. VPNs use strong encryption to ensure that your private information remains secure as it passes through the Internet. With the exception of some internetwork lag, the two networks are virtually transparent to the end users.

Before the Internet and VPNs, a company was required to pay for a dedicated leased line from one site to another. Telephone companies typically charge by the mile for leased lines, which as you can imagine, can become rather pricey depending on the distance between sites. On the other hand, bandwidth on the Internet costs the same no matter how far away one site may be from another.

The Cisco IOS fully supports site-to-site VPN connections using *IP Security* (IPSec), the de facto standard in VPN encryption. In this chapter, you'll learn how to use the IOS to join two or more sites together using IPSec. You will also learn how to secure those connections using encryption and ACLs.

This chapter will go into some detail about how encryption works with a VPN, and will explain the IOS commands used to set up a VPN between two Cisco routers. If you are not interested in the details and simply want to get your VPN up and running, I suggest you skip forward to the Summary section where you will find a down and dirty list of all the commands needed to set up your VPN. You may even want to look at the IOS examples there first, before reading the chapter, to get an overview of all the IOS commands.

Preparing Your Sites

In this chapter, I will refer to a Cisco 831 router that uses interface names *e0* for the LAN ports and *e1* for the WAN interface. Before you can join two sites together using a VPN, you need to make sure that the local network (at each site) is configured with a unique network number. This ensures that your router knows which packets are destined for hosts on your local network, and which should be forwarded through the VPN to the remote site.

It works like this: whenever a host sends data addressed to anything other than another host on the local network, it is sent to the default gateway. Since your router is the default gateway, it will receive those data packets. The router then uses the packet's destination network address to determine whether the packets should be sent to the remote site via the VPN or some other remote location (i.e., www.cnn.com). If destined for the remote site, the router then encrypts the data using IPsec and forwards it to the remote site through the VPN tunnel where it is decrypted by the remote site's router and forwarded to the appropriate host on the remote network.

A VPN connection between two sites is often referred to as a *Site-to-Site*, *Gateway-to-Gateway*, or *LAN-to-LAN* VPN tunnel, and the routers at each end are referred to as VPN endpoints.

There are two parts to VPN encryption known as "Phase 1" and "Phase 2." Phase 1 is the *Internet Key Exchange* (IKE) negotiation phase. IKE is used to manage your VPN connections using Security Associations, or SAs. Phase 2 is the IPsec phase, in which your data is encrypted using IPsec and tunneled between the VPN endpoints through the Internet.

Setting Up the VPN

Before you begin configuring a VPN, take a look at the diagram in Figure 4-1. It shows a VPN between a branch office and a corporate office. In this example, the branch office will be the router "lab-r1" which is configured like the router in Chapter 2. During this tutorial, you will modify that configuration to support a VPN tunnel between the two sites.

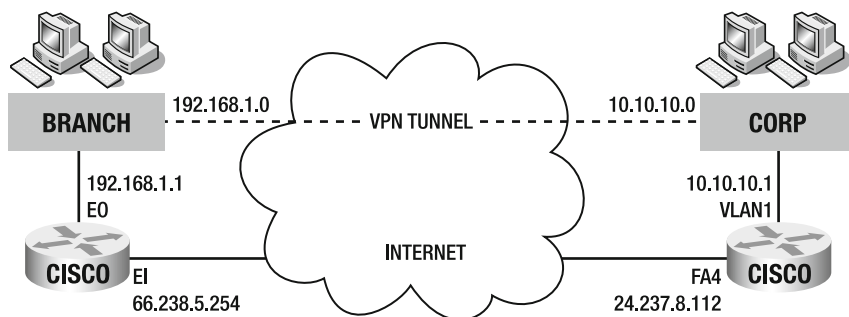


Figure 4-1. VPN diagram

Because NAT cannot be processed in a VPN tunnel, you begin by removing your existing ACL for NAT and configuring a new (VPN-friendly) ACL used by NAT that prevents packets from being processed by NAT as they traverse from the branch office through the VPN to the corporate office. You do this by denying the private network address at the branch office access to the private network address at the corporate

office. Once this ACL is created, it will be used in conjunction with a routing policy that is applied to NAT on the router.

Step 1: Create a VPN-Friendly ACL for NAT

The commands in Listing 4-1 display the steps to delete your old ACL used by NAT, named “NAT-ACL,” and create a new one, which denies all hosts from network 192.168.1.0 to network 10.10.10.0, but allows all hosts on network 192.168.1.0 to anything else. This will prevent packets from being processed by NAT through the VPN, but will allow it to process all packets destined for the Internet.

Listing 4-1. *How to Create a New ACL for NAT*

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# no ip access-list extended NAT-ACL

lab-r1(config)# ip access-list extended NAT-ACL
lab-r1(config-ext-nacl)# deny ip 192.168.1.0 0.0.0.255 10.10.10.0 0.0.0.255
lab-r1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 any
lab-r1(config-ext-nacl)# exit
```

Because ACLs filter packets in the order in which the ACL rules are created, it's important to make sure that the deny statement comes before the permit statement. If you don't do this, the permit rule will be matched first, allowing the VPN packets to be processed by NAT (which doesn't work).

To use your new ACL with the VPN, you need to create a routing policy that will be incorporated into your NAT statement.

Step 2: Define a VPN Routing Policy for Your WAN Interface

Routing policies are beyond the scope of this book but you need to use them to get your VPN up and running. The route-map NO-NAT permit 10 command (shown in Listing 4-2) is used to define a routing policy named “NO-NAT” that uses the “NAT-ACL” you created in Listing 4-1. This policy is applied to NAT on your WAN interface.

Listing 4-2. *How to Create a VPN Routing Policy for Your LAN Interface*

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# route-map NO-NAT permit 10
lab-r1(config-route-map)# match ip address NAT-ACL
```

Now that you've defined a VPN routing policy, you need to remove your existing `ip nat` statement and create a new one that uses the new routing policy named "NO-NAT."

Step 3: Apply Your VPN Routing Policy to NAT

You have to remove the existing NAT statement before you can create a new one that uses a routing policy. Begin by using the global configuration mode command `no ip nat inside source list NAT-ACL int e1 overload`, shown in Listing 4-3, to remove your existing NAT statement. Next, use the new `ip nat` statement to configure NAT using the "NO-NAT" routing policy you created in Listing 4-2 and apply it to the WAN interface `e1`.

Listing 4-3. *How to Create a New NAT Statement*

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# no ip nat inside source list NAT-ACL int e1 overload
lab-r1(config)# ip nat inside source route-map NO-NAT int e1 overload
```

Step 4: Define a VPN Routing Policy for Your LAN Interface

In addition to the NO-NAT routing policy that you apply to the `ip nat` statement on your WAN interface, you must also create a routing policy that is applied to your LAN interface. This policy prevents VPN data that is *returning* from your remote LAN from being processed by NAT as it passes through the VPN tunnel. In Listing 4-4 I have named this policy "NONAT-LAN," but you can use any name that makes sense to you.

The NONAT-LAN routing policy is a bit more complicated than the previously created NO-NAT policy. The LAN routing policy requires that you create a virtual loopback interface, an ACL that I have named "NONAT-LAN-ACL," and the routing policy itself. The ACL permits your local private network access to the remote private network. Once you create the policy, you apply it to the LAN interface of your router. It will redirect VPN originated traffic to the `loopback0` interface and from there it will be routed according to your routing table, thus sending the traffic to your WAN interface, which will take care of the encryption process back through the tunnel.

Listing 4-4 displays all the commands to configure the NONAT-LAN routing policy.

Listing 4-4. *How to Create a VPN Routing Policy for Your LAN Interface*

```
lab-r1# config t
lab-r1(config)# int loopback0
lab-r1(config-if)# ip address 1.1.1.1 255.255.255.252
lab-r1(config-if)# exit
```

```
lab-r1(config)# ip access-list ext NONAT-LAN-ACL
lab-r1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 10.10.10.0 0.0.0.255
lab-r1(config-ext-nacl)# exit

lab-r1(config)# route-map NONAT-LAN
lab-r1(config-route-map)# match ip address NONAT-LAN-ACL
lab-r1(config-route-map)# set interface loopback0
lab-r1(config-route-map)# exit

lab-r1(config)# int e0
lab-r1(config-if)# ip policy route-map NONAT-LAN
lab-r1(config-if)# exit
lab-r1(config)#
```

Configuring IKE Phase 1

To configure IPsec, you need to define an *Internet Security Association Key Management Protocol* (ISAKMP) policy. ISAKMP is the protocol used to negotiate a cryptographic key exchange that is used when the routers set up a VPN connection between one another. This is often referred to as the “Phase 1” portion of the VPN setup. The commands in Listing 4-5 configure Phase 1 IKE negotiation.

Listing 4-5. *How to Configure Phase 1 IKE Negotiation*

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# crypto isakmp policy 1
lab-r1(config-isakmp)# encryption aes 256
lab-r1(config-isakmp)# hash sha
lab-r1(config-isakmp)# authentication pre-share
lab-r1(config-isakmp)# group 2
lab-r1(config-isakmp)# exit
lab-r1(config)#
```

Step 1: Create a Key Exchange Policy

You can create more than one ISAKMP policy on your router, which can be used with different VPN connections. Multiple policies are distinguished by their policy number. In Listing 4-5, the policy number has been set to “1” with the command `crypto isakmp policy 1`. This command puts you in ISAKMP configuration mode, which allows you to configure the policy. You can confirm you’re in ISAKMP configuration mode by the prompt (config-isakmp).

Step 2: Define the Encryption Type

Cisco routers support several types of VPN encryption. The most common are the *Data Encryption Standard* (DES) and *Triple Data Encryption Standard* (3DES [pronounced “triple dez”]). 3DES is a triple strength version of DES. Cisco routers also support the newest encryption standard known as the *Advanced Encryption Standard* (AES). As of this writing, this is the strongest form of VPN encryption available. Use the command `encryption aes 256` to invoke this encryption method.

Note AES provides stronger encryption than 3DES, yet uses less of your router’s processing power to encrypt and decrypt data. Therefore, I recommend that you always use AES to secure your VPN tunnels. However, if you need to set up a VPN using older equipment (routers that don’t support AES) you can replace the command statements that use “aes 256” with “3des”.

Step 3: Define a Cryptographic Hash Function

A *hash function* takes a string of characters, no matter what their length, and produces a fixed length string known as a digital fingerprint. In the case of your VPN, the *Secure Hash Algorithm* (SHA) creates a digital fingerprint of your VPN’s preshared key. Use the command `hash sha` to use a SHA hash function. You could also use a *Message Digest Algorithm* (md5) hash, but SHA is considered a more secure standard. Each router participating in a VPN must use the same hash function.

Step 4: Define Your IKE Key Type

The command `authentication pre-share` instructs the IOS to use a *preshared key* for the IKE (phase 1) portion of the VPN setup. Each VPN router must use the same key type.

Step 5: Define Your IKE Key Size

The group command defines the size of your IKE key. Larger IKE keys are more secure but take the router longer to encrypt and decrypt, which can impact router performance. Use the group 2 command to use a 1024-bit *Diffie-Hellman* (DH) key. You can also use group 1 for a 768-bit key or group 3 for a 1536-bit key. Each router participating in the VPN must use the same group number.

DIFFIE-HELLMAN KEY PROTOCOL

Whitfield Diffie and Martin Hellman introduced the concept of Public Key/Private Key encryption in 1976. This form of encryption allows two parties to establish a secure form of communication using two sets of keys. A private set and a public set. For the first time, this allowed two parties to share information securely and publicly without the need to provide all the encryption keys to each other (which themselves could then be compromised).

This is how it works: party A wants to share encrypted data with party B. Each party (A and B) has a *private* key that they keep secret, and a *public* key that can be shared over the Internet. When the two parties want to share information securely, they exchange only their public keys. Party A takes party B's public key and uses its own private key to encrypt the data. That data is then sent to party B. Party B uses party A's public key and its own private key to decrypt the data. Only party B's private key can be used to decrypt the data received from party A. This is what revolutionized encryption. As long as the *private* keys are not compromised, it is extremely difficult to crack this form of encryption. It revolutionized computer security as we know it today and set the stage for e-commerce using the Internet.

Step 6: Create a Preshared Key

In this step, you set the VPN's preshared key. I like to think of the preshared key as the VPN's password. You must use the exact same preshared key on both routers participating in the VPN. The key is case sensitive, so keep that in mind when setting up your real VPN. You associate the key with the static IP address of the router at the other end of the VPN. Since you're configuring the branch router, you should use the static IP address of the corporate router as shown in Listing 4-6.

Note The VPN preshared key is always case sensitive and must be the same on both routers participating in a VPN connection.

Listing 4-6. How to Create a Preshared Key

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# crypto isakmp key Fre@ksh0w! address 24.237.8.112
lab-r1(config)# crypto ipsec security-association lifetime seconds 28800
```

In Listing 4-6, the global configuration command `crypto isakmp key` sets the VPN pre-shared key to “Fre@ksh0w!”. It’s a silly password, but it illustrates an important point. Anytime you create passwords, you want them to be *complex*. A complex password should be comprised of at least eight characters, include mixed case letters, numbers, and a symbol. This ensures that the password is difficult to crack.

Next, the pre-shared key is associated with the static IP address of the Corporate router at address “24.237.8.112”. This is the static IP address of the participating router for this VPN connection.

To protect your encryption keys from system crackers, the router periodically regenerates the SA encryption keys. By rotating the encryption keys that are used by IKE to set up the VPN connection, you make your VPN a more difficult moving target for hackers. Use the global configuration command `crypto ipsec security-association lifetime seconds` to set the SA interval. Eight hours (28,800) is a common interval for most routers, but you can reduce it for higher security.

Configuring IPsec Phase 2

IKE Phase 1 determines the initial communication setup between VPN routers. Phase 2 determines how data will be encrypted between the VPN routers.

You begin by creating an ACL that permits the private networks to communicate with one another through the VPN. Next, you create a *transform set* that determines the encryption algorithm and hash function. The transform set often uses the same encryption and hash as IKE Phase 1, but they can be different. Next, you create a *crypto map* for the site to which you want your router to connect (corporate, in this case). Finally, using the `crypto map` command, you apply the VPN to your WAN interface. Listing 4-7 shows the commands needed to configure Phase 2.

Listing 4-7. *How to Configure Phase 2 IPsec Data Encryption*

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# ip access-list extended VPN-ACL
lab-r1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 10.10.10.0 0.0.0.255
lab-r1(config-ext-nacl)# exit

lab-r1(config)# crypto ipsec transform-set SET1 esp-aes 256 esp-sha-hmac
```

```
lab-r1(config)# crypto map CORP-VPN 10 ipsec-isakmp
lab-r1(config-crypto-map)# set peer 24.237.8.112
lab-r1(config-crypto-map)# set transform-set SET1
lab-r1(config-crypto-map)# set pfs group2
lab-r1(config-crypto-map)# match address VPN-ACL
lab-r1(config-crypto-map)# exit

lab-r1(config)# int e1
lab-r1(config-if)# crypto map CORP-VPN

lab-r1(config-if)# ^Z
lab-r1#
```

Step 1: Create a VPN-ACL

Begin by creating an ACL that will be used to control which hosts are permitted to use the VPN tunnel. The ACL in Listing 4-7, named “VPN-ACL”, permits all hosts on network 192.168.1.0 to access all hosts on network 10.10.10.0. Like any ACL, you can modify this ACL to permit or deny hosts to the VPN tunnel.

Step 2: Create a Transform Set

The *Encapsulating Security Payload* (ESP) protocol is the protocol that provides the IPsec packet encryption. The `crypto ipsec transform-set` command used in Listing 4-7 defines which ESP encryption type will be provided for IPsec. In this example, the set has been named “SET1” and it uses AES 256 bit encryption with an SHA hash function. You can name your transform set anything you would like, and you can create multiple transform sets that can be applied to different VPN connections.

Step 3: Create a Crypto Map

The global configuration command `crypto map` ties all your IPsec VPN parameters together for a particular VPN connection. In this example, the connection is to the corporate office. The command `crypto map CORP-VPN 10 ipsec-isakmp` creates a VPN named “CORP-VPN” that uses a sequence number of 10 (not important here) and uses ISAKMP keying. The `crypto map` command places you in crypto map configuration mode. The prompt (`config-crypto-map`) confirms you are in crypto map configuration mode.

Step 4: Set the VPN Peer

The command `set peer 24.237.8.112` is used to define the endpoint for this VPN connection. In this case, the endpoint is the corporate router at IP address 24.237.8.112. This is the corporate router's static WAN IP address.

Step 5: Set the Transform Set

The command `set transform-set SET1` is used to define which transform set will be used for this VPN connection. In this example, you use the previously created transform-set "SET1".

Step 6: Set the PFS Group

As mentioned earlier, frequently regenerating encryption keys during a VPN session helps secure your data from system crackers. Perfect Forward Security (PFS) ensures that a previously used session key cannot be used to generate any new keys. The crypto map configuration command `set pfs group2` is used to set the DH key size to 1024-bit encryption.

Step 7: Apply Your VPN ACL

The command `match address VPN-ACL` is used to control which hosts will have access through the VPN. In this example, you use the ACL you created, named "VPN-ACL", to allow all branch hosts on network 192.168.1.0 to access all corporate hosts on network 10.10.10.0.

Step 8: Apply the Crypto Map

Use the global configuration command `int e1` to enter configuration mode for your router's WAN interface. The command `crypto map CORP-VPN` applies your newly created VPN, named "CORP-VPN", to your router's WAN interface.

At this point, everything is configured at the branch office for a VPN connection to the corporate office.

Modifying Your IPFW-ACL

The final step is to modify your WAN ACL named "IPFW-ACL" to allow the set up of IPsec VPNs via the WAN interface.

Tip The easiest way to modify ACL rules is to keep them in a text document and use copy/paste to configure them on your router.

Listing 4-8 shows all the ACL rules necessary to allow any remote site to set up a VPN through the router's WAN interface. I have also included the ACL rules that allow you to use the ping and traceroute utilities to troubleshoot your VPN.

Listing 4-8. *How to Modify Your IPFW-ACL*

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# ip access-list extended IPFW-ACL
lab-r1(config-ext-nacl)# permit udp any any eq isakmp
lab-r1(config-ext-nacl)# permit udp any eq isakmp any
lab-r1(config-ext-nacl)# permit esp any any

! ACL rules to allow outbound ping and traceroute
corp(config-ext-nacl)# permit icmp any any administratively-prohibited
corp(config-ext-nacl)# permit icmp any any echo-reply
corp(config-ext-nacl)# permit icmp any any packet-too-big
corp(config-ext-nacl)# permit icmp any any time-exceeded
corp(config-ext-nacl)# permit icmp any any traceroute
corp(config-ext-nacl)# permit gre any any
corp(config-ext-nacl)# deny ip any any
corp(config-ext-nacl)# exit
corp(config)#
```

The first two ACL rules in Listing 4-8, `permit udp any any eq isakmp` and `permit udp any eq isakmp any`, which are applied to the WAN interface of the router, will allow any site to set up an IKE connection with this router using ISAKMP. They also allow this router to initiate an IKE negotiation with any router. Finally, the `permit esp any any` command allows IPsec encrypted packets in and out of this router using the ESP protocol.

That's all there is to configuring one side of a VPN. Remember to save your router configuration with the privileged EXEC mode command `copy run start`. To configure the other endpoint router, all you need to do is change the LAN IP addresses used in Listings 4-1 and 4-7, and the static IP address for the endpoint router used in Listings 4-6 and 4-7. It's a piece of cake!

Tip You can tighten security by replacing the `any` statements in these ACL rules with the static IP addresses of your branch and corporate routers. This will ensure that only those sites can establish a VPN with each other.

Verifying Your VPN Connection

To bring up the VPN tunnel, use the ping utility from a workstation on your LAN. Try ping-ing a host at the corporate network from a PC at the branch network. If you receive a reply, then you're up and running. If you don't, then something has gone wrong and you will want to start troubleshooting (described in the next section, "Troubleshooting").

You can check the status of your VPN tunnel (whether or not your VPN tunnel is up) on a Cisco router with the privileged EXEC mode commands `show crypto isakmp sa` (abbreviated `sh crypt isakmp sa`) and `show crypto ipsec sa` (abbreviated `sh crypt ipsec sa`), as in the next two examples.

```
lab-r1> enable
lab-r1# sh crypt isakmp sa
```

The following output shows an *active and connected VPN tunnel* between the branch office at IP address 66.238.5.254 and the corporate office at IP address 24.237.8.112. The connection ID is 1005.

IPv4 Crypto ISAKMP SA					
Dst	src	state	conn-id	slot	status
24.237.8.112	66.238.5.254	QM_IDLE	1005	0	ACTIVE

Next, use the `show crypto ipsec sa` command to verify your encryption and the reliability of the tunnel.

```
Router> enable
Router# sh crypt ipsec sa
```

Below is sample output from the `show` command. Focus on the items highlighted in bold. First, look at the statistics for encrypted and decrypted packets. Also look for any send or receive errors. There will usually be a couple, but a high number could indicate a problem with your WAN links. Next, look at the "inbound and outbound sas" to ensure that your VPN is secured with AES encryption. Security is very important on a VPN connection and encryption is the key component in keeping your information safe from prying eyes as it traverses the Internet between your sites.

```
interface: Ethernet1
  Crypto map tag: CORP-VPN, local addr. 66.238.5.254
```

```

protected vrf:
local ident (addr/mask/prot/port): (192.168.1.0.0/255.255.255.0/0/0)
remote ident (addr/mask/prot/port): (10.10.10.0.0/255.255.255.0/0/0)
current_peer: 24.237.8.112:4500
    PERMIT, flags={origin_is_acl,}

#pkts encaps: 19211, #pkts encrypt: 19211, #pkts digest 19211
#pkts decaps: 14384, #pkts decrypt: 14384, #pkts verify 14384
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0
#pkts not decompressed: 0, #pkts decompress failed: 0
#send errors 2, #recv errors 0

local crypto endpt.: 66.238.5.254, remote crypto endpt.: 24.237.8.112
path mtu 1500, media mtu 1500
current outbound spi: 3627C7D0

inbound esp sas:
    spi: 0x1A7E43C6(45432352)
        transform: esp-256-aes esp-sha-hmac ,
        in use settings ={Tunnel UDP-Encaps, }
        slot: 0, conn id: 2000, flow_id: 1, crypto map: CORP-VPN
        sa timing: remaining key lifetime (k/sec): (4497963/3515)
        IV size: 8 bytes
        replay detection support: Y

inbound ah sas:

inbound pcp sas:

outbound esp sas:
    spi: 0x3727C7D0(8560801)
        transform: esp-256-aes esp-sha-hmac ,
        in use settings ={Tunnel UDP-Encaps, }
        slot: 0, conn id: 2001, flow_id: 2, crypto map: CORP-VPN
        sa timing: remaining key lifetime (k/sec): (4497963/3515)
        IV size: 8 bytes
        replay detection support: Y

outbound ah sas:

outbound pcp sas:

```

Troubleshooting

It's great when everything goes well the first time, we all love that. However, having done this a few times, I've had plenty of opportunities to make a lot of silly mistakes that have prevented me from successfully establishing a VPN tunnel. Don't panic if your VPN doesn't work right from the start. It's probably a very simple mistake.

Use the privileged EXEC mode command `sh run` to display your router's running configuration. This section covers a few common items to check.

General Network Settings

Check your Internet connectivity by either pinging a host on the Internet or browsing a web site. This will verify that your interfaces are up and that IP network connectivity is working properly.

Note Be sure that your "IPFW-ACL" has the rules for ICMP ping and traceroute that were introduced in Chapter 2.

Check the "IPFW-ACL" on your WAN interface. You must allow ISAKMP to set up the Phase 1 negotiation, and ESP to allow Phase 2 IPsec encryption (see Listing 4-8).

Check your "NO-NAT" routing policy and "NAT-ACL." Remember, you don't want local LAN packets to be processed by NAT as they pass through the VPN tunnel (see Listings 4-1 and 4-2). NAT through a VPN tunnel doesn't work.

Check that you have applied the VPN to your WAN interface using the `crypto map` command (e.g., `crypto map CORP-VPN`).

IKE Phase 1 Settings

Check your preshared key. Make sure the key is the same on both endpoint routers—remember, the key is always case sensitive.

Check that your cryptographic ISAKMP policy is the same on each endpoint. Verify the encryption (AES 256), hash (SHA), group number (Group 2), and authentication type (auth preshare). Once again, these parameters must be identical on both endpoint routers or your VPN will not work.

IPSec Phase 2 Settings

Check that the transform set used by your IPSec crypto map is the same on both endpoints.

Check that you have the same IPSec PFS Group number on both endpoints. You can't use PFS Group 1 on one endpoint and PFS Group 2 on the other. Both endpoints must use the same DH bit size.

Check the address of your peer. Make sure that you have set the correct WAN IP address for the router on the other end of your VPN.

When in Doubt, Print It Out

Making a printout of each router's IOS configuration and performing a line-by-line comparison of the VPN parameters is the best way to find the problem. As I stated earlier, when you're experiencing problems bringing up your VPN tunnel, it's most likely a simple mistake (often a typo somewhere in the configuration). I like to use a highlighter and highlight all the VPN parameters on one of the IOS printouts. This way I don't get bogged down by all the non-VPN IOS commands. Next, I use a pen and work my way through the other printout, putting a tick mark next to all the parameters that are correct. So far, I have always found one of the mistakes mentioned above and I bet you will, too!

Summary

Hopefully, by this point you know more about VPN technology than you ever wanted. Although it's not necessary to understand every detail of VPN encryption and encapsulation, it is good to understand the basic concepts such as Phase 1 IKE negotiation and Phase 2 IPSec encapsulation. Knowing the concepts will help you set up your VPN, troubleshoot VPN issues, and (if you want to) it will even help you set up a VPN from a Cisco router to non-Cisco gear, but no one would want to do that!

Set Up the VPN

If you have opted to skip reading the beginning of this chapter and jump right into setting up your VPN, then there are a couple things you need to know.

First, before you can join your sites together using a VPN, you need to make sure that the local LAN at each site has a unique network number. In the examples below, they are 192.168.1.0 and 10.10.10.0.

Second, each site must use the exact same case-sensitive, preshared key. For security purposes, you should use a somewhat complex preshared key; one that uses a combination of letters, numbers, and symbols. “Fre@kSh0w!” will be the key used in this example. Notice that this key uses upper- and lowercase letters, an @ symbol, a zero in place of the “o,” and an exclamation point.

Lastly, there are only five parameters that will need to be changed in the configurations to create a VPN between your actual sites. They are the LAN and WAN addresses (for each site) and the preshared key. Table 4-1 shows the parameters for this example. Figure 4-2 shows conceptually how the VPN is configured.

Table 4-1. *VPN Parameters for This Chapter's Example*

Office Type	Parameter	Value
Branch office	LAN	192.168.1.0
Branch office	WAN	66.238.5.254
Corporate	LAN	10.10.10.0
Corporate	WAN	24.237.8.112
Both	Preshared Key	Fre@kSh0w!

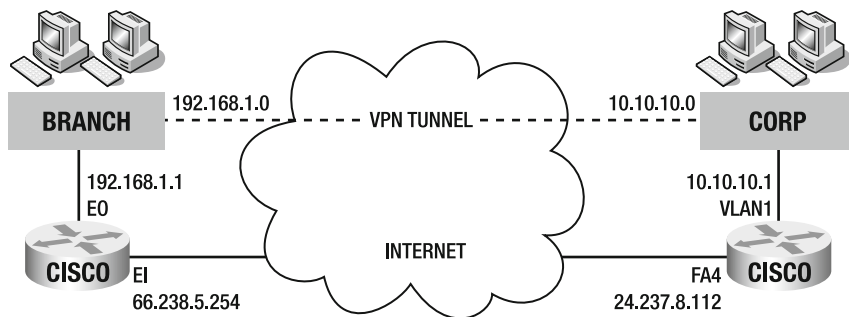


Figure 4-2. *Example VPN configuration*

Since all Cisco routers are not alike, your router's LAN and WAN interface names may differ from the examples given. Table 4-2 gives the router models and the interface names used for the branch office and corporate office routers used in the example to follow.

Table 4-2. *Example Router Models and Interface Names*

Office Type	Router Model	Interface	Network Type
Branch Office	Cisco 831 Router	Ethernet 0 (E0)	LAN
		Ethernet 1 (E1)	WAN
Corporate Office	Cisco 851 Router	VLAN 1 (VLAN1)	LAN
		Fastethernet 4 (FA4)	WAN

The configurations on the next few pages contain all the Cisco IOS commands that you will need to know to set up a VPN between two sites. As I mentioned earlier, you only need to change the IP addresses and preshared key. All other aspects of the configurations are identical. Keeping this in mind helps simplify the process and makes it less intimidating.

There are four ACLs used in the VPN set up. They are named

- **IPFW-ACL**—is applied to the WAN interface on each endpoint router and is used to allow a VPN setup between the two sites.
- **NAT-ACL**—is used with a “route map” routing policy to prevent VPN destined packets from being processed by NAT. All other packets, destined for the Internet, will be processed by NAT.
- **NONAT-LAN-ACL**—is used with a “route map” routing policy to prevent VPN-destined packets from being processed by NAT as they return from an endpoint router. All other packets, destined for the Internet, will be processed by NAT.
- **VPN-ACL**—is used to prevent or allow hosts through the VPN tunnel. In this example, all LAN hosts on each side of the VPN are allowed through the tunnel to all LAN hosts on the other side of the VPN.

Note The Branch and Corporate Office VPN examples are shown next.

Branch Office VPN Configuration

The branch office configuration is based on the following address and interface combinations:

LAN Address: 192.168.1.0 on Interface E0

WAN Address: 66.238.5.254 on Interface E1

Use the following commands to create this branch office configuration:

```
site1> enable
site1# config t

site1(config)# int e0
site1(config-if)# ip address 192.168.1.1 255.255.255.0
site1(config-if)# int e1
site1(config-if)# ip address 66.238.5.254 255.255.255.0
site1(config-if)# exit

site1(config)# ip route 0.0.0.0 0.0.0.0 66.238.5.1           ! Default Route-ISP

site1(config)# ip access-list extended IPFW-ACL              ! Allow VPN Setup
site1(config-ext-nacl)# permit udp any any eq isakmp
site1(config-ext-nacl)# permit udp any eq isakmp any
site1(config-ext-nacl)# permit esp any any

site1(config-ext-nacl)# permit icmp any any administratively-prohibited
site1(config-ext-nacl)# permit icmp any any echo-reply
site1(config-ext-nacl)# permit icmp any any packet-too-big
site1(config-ext-nacl)# permit icmp any any time-exceeded
site1(config-ext-nacl)# permit icmp any any traceroute
site1(config-ext-nacl)# permit gre any any
site1(config-ext-nacl)# deny ip any any
site1(config-ext-nacl)# exit

site1(config)# ip inspect name IPFW tcp timeout 3600        ! CBAC Firewall
site1(config)# ip inspect name IPFW udp timeout 15

site1(config)# ip access-list extended NAT-ACL
site1(config-ext-nacl)# deny ip 192.168.1.0 0.0.0.255 10.10.10.0 0.0.0.255
site1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 any
site1(config-ext-nacl)# exit
```

```
site1(config)# route-map NO-NAT permit 10
site1(config-route-map)# match ip address NAT-ACL
site1(config-route-map)# exit
site1(config)# ip nat inside source route-map NO-NAT int e1 overload

site1(config)# int loopback0
site1(config-if)# ip address 1.1.1.1 255.255.255.252
site1(config-if)# exit

site1(config)# ip access-list ext NONAT-LAN-ACL
site1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 10.10.10.0 0.0.0.255
site1(config-ext-nacl)# exit

site1(config)# route-map NONAT-LAN
site1(config-route-map)# match ip address NONAT-LAN-ACL
site1(config-route-map)# set interface loopback0
site1(config-route-map)# exit

site1(config)# int e0
site1(config-if)# ip policy route-map NONAT-LAN
site1(config-if)# exit

site1(config)# crypto isakmp policy 1                                ! Phase 1 (IKE)
site1(config-isakmp)# encryption aes 256
site1(config-isakmp)# hash sha
site1(config-isakmp)# authentication pre-share
site1(config-isakmp)# group 2
site1(config-isakmp)# exit

site1(config)# crypto isakmp key Fre@kSh0w! Address 24.237.8.112
site1(config)# crypto ipsec security-association lifetime seconds 28800

site1(config)# ip access-list extended VPN-ACL
site1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 10.10.10.0 0.0.0.255
site1(config-ext-nacl)# exit

site1(config)# crypto ipsec transform-set SET1 esp-aes 256 esp-sha-hmac

site1(config)# crypto map CORP-VPN 10 ipsec-isakmp                    ! Phase 2 (IPSec)
site1(config-crypto-map)# set peer 24.237.8.112
site1(config-crypto-map)# set transform-set SET1
site1(config-crypto-map)# set pfs group2
site1(config-crypto-map)# match address VPN-ACL
site1(config-crypto-map)# exit
```

```

site1(config)# int e0                                ! LAN Interface
site1(config-if)# ip nat inside

site1(config-if)# int e1                                ! WAN Interface
site1(config-if)# ip nat outside
site1(config-if)# ip access-group IPFW-ACL in
site1(config-if)# ip inspect IPFW out
site1(config-if)# crypto map CORP-VPN
site1(config-if)# exit
site1(config)#

```

Corporate Office VPN Configuration

Similarly, the corporate office router is configured using the following values as a base:

LAN Address: 10.10.10.0 on Interface VLAN1

WAN Address: 24.237.8.112 on Interface FA4

The following are the commands to carry out that configuration:

```

corp> enable
corp# config t

corp(config)# int vlan1
corp(config-if )# ip address 10.10.10.1 255.255.255.0
corp(config-if )# int fa4
corp(config-if)# ip address 24.237.8.112 255.255.255.224
corp(config-if)# exit

corp(config)# ip route 0.0.0.0 0.0.0.0 24.237.8.1      ! Default Route-ISP

corp(config)# ip access-list extended IPFW-ACL          ! Allow VPN Setup
corp(config-ext-nacl)# permit udp any any eq isakmp
corp(config-ext-nacl)# permit udp any eq isakmp any
corp(config-ext-nacl)# permit esp any any

corp(config-ext-nacl)# permit icmp any any administratively-prohibited
corp(config-ext-nacl)# permit icmp any any echo-reply
corp(config-ext-nacl)# permit icmp any any packet-too-big
corp(config-ext-nacl)# permit icmp any any time-exceeded
corp(config-ext-nacl)# permit icmp any any traceroute
corp(config-ext-nacl)# permit gre any any
corp(config-ext-nacl)# deny ip any any
corp(config-ext-nacl)# exit

```

```
corp(config)# ip inspect name IPFW tcp timeout 3600                ! CBAC Firewall
corp(config)# ip inspect name IPFW udp timeout 15

corp(config)# ip access-list extended NAT-ACL
corp(config-ext-nacl)# deny ip 10.10.10.0 0.0.0.255 192.168.1.0 0.0.0.255
corp(config-ext-nacl)# permit ip 10.10.10.0 0.0.0.255 any
corp(config-ext-nacl)# exit

corp(config)# route-map NO-NAT permit 10
corp(config-route-map)# match ip address NAT-ACL
corp(config-route-map)# exit
corp(config)# ip nat inside source route-map NO-NAT int fa4 overload

corp(config)# int loopback0
corp(config-if)# ip address 1.1.1.1 255.255.255.252
corp(config-if)# exit

corp(config)# ip access-list ext NONAT-LAN-ACL
corp(config-ext-nacl)# permit ip 10.10.10.0 0.0.0.255 192.168.1.0 0.0.0.255
corp(config-ext-nacl)# exit

corp(config)# route-map NONAT-LAN
corp(config-route-map)# match ip address NONAT-LAN-ACL
corp(config-route-map)# set interface loopback0
corp(config-route-map)# exit

corp(config)# int vlan1
corp(config-if)# ip policy route-map NONAT-LAN
corp(config-if)# exit

corp(config)# crypto isakmp policy 1                               ! Phase 1 (IKE)
corp(config-isakmp)# encryption aes 256
corp(config-isakmp)# hash sha
corp(config-isakmp)# authentication pre-share
corp(config-isakmp)# group 2
corp(config-isakmp)# exit

corp(config)# crypto isakmp key Fre@kSh0w! address 66.238.5.254
corp(config)# crypto ipsec security-association lifetime seconds 28800

corp(config)# ip access-list extended VPN-ACL
corp(config-ext-nacl)# permit ip 10.10.10.0 0.0.0.255 192.168.1.0 0.0.0.255
corp(config-ext-nacl)# exit
```

```
corp(config)# crypto ipsec transform-set SET1 esp-aes 256 esp-sha-hmac

corp(config)# crypto map CORP-VPN 10 ipsec-isakmp                ! Phase 2 (IPSec)
corp(config-crypto-map)# set peer 66.238.5.254
corp(config-crypto-map)# set transform-set SET1
corp(config-crypto-map)# set pfs group2
corp(config-crypto-map)# match address VPN-ACL
corp(config-crypto-map)# exit

corp(config)# int vlan1                                          ! LAN interface
corp(config-if)# ip nat inside

corp(config-if)# int fa4                                         ! WAN interface
corp(config-if)# ip nat outside
corp(config-if)# ip access-group IPFW-ACL in
corp(config-if)# ip inspect IPFW out
corp(config-if)# crypto map CORP-VPN
corp(config-if)# exit
corp(config)#
```

Troubleshoot Your VPN

There are three main VPN components to check when troubleshooting your VPN connections. They are your general network settings, Phase 1 IKE negotiation, and Phase 2 IPsec encryption.

General Network Settings

Make sure you have network connectivity to the Internet and are able to browse a web site or ping an Internet host. Check that you have applied the VPN to your WAN interface using the `crypto map` command.

```
corp(config-if)# crypto map CORP-VPN
```

Check all your ACLs. Here is a complete list of rules for your WAN ACL named “IPFW-ACL”:

```
corp(config)# ip access-list extended IPFW-ACL
corp(config-ext-nacl)# permit udp any any eq isakmp
corp(config-ext-nacl)# permit udp any any eq isakmp any
corp(config-ext-nacl)# permit esp any any
corp(config-ext-nacl)# permit icmp any any administratively-prohibited
corp(config-ext-nacl)# permit icmp any any echo-reply
corp(config-ext-nacl)# permit icmp any any packet-too-big
corp(config-ext-nacl)# permit icmp any any time-exceeded
corp(config-ext-nacl)# permit icmp any any traceroute
corp(config-ext-nacl)# permit gre any any
```



```
corp(config-ext-nacl)# deny ip any any
corp(config-ext-nacl)# exit
corp(config)#
```

IKE Phase 1 Settings

Check your preshared key. Make sure the key is the same on both routers. Check your ISAKMP policy. It should be the same on both routers. Verify the encryption (AES), hash (SHA), group number (Group 2), and authentication type (preshare).

IPSec Phase 2 Settings

Check that the transform set used by your IPSec crypto map is the same on both routers. Check that you have the same IPSec PFS Group number on both routers. Check the address of your peer. Make sure you have set the correct WAN IP address for the router on the other end of your VPN.

When in Doubt, Print It Out

A side-by-side comparison of the two router configurations can be very useful. Remember, their VPN configurations should be identical except for the LAN and WAN IP addresses.



Beyond the Basics

By now you should have a good understanding of the IOS command structure and many of the configuration commands and concepts used to set up a router for a small business. In this chapter, I'll cover some of the more advanced features of the Cisco IOS. These features include creating local users on your router, configuring Secure Shell, recovering a lost password, backing up and upgrading the Cisco IOS, as well as backing up your router's configuration. You will also learn about performance tuning your ACLs by ordering the rules so they work more efficiently, and about configuring the Cisco firewall's *E-mail Inspection Engine* (EIE) that can help protect your e-mail server from malicious activity directed from the Internet.

I refer to these commands as “advanced,” but in reality they are no more difficult to understand and implement than any other feature you have used throughout this book. That's the dirty little secret about Cisco routers and the IOS, in particular. Cisco routers are not very difficult to set up and manage if you have just a little knowledge of the concepts. So don't worry, this chapter may be called “Beyond the Basics,” but the material is not difficult to understand and I will guide you every step of the way.

Creating a Local User on the Router

One of the techniques you can use to make your router more secure is to require a username and password prior to gaining access to the router. Even though you may have set your router's privileged EXEC mode password, creating users and making a username and password mandatory will add another level of complexity that helps protect your router from system crackers who are attempting to gain access. When creating users, I recommend that you do not use common names such as *admin* or *administrator*, which make your router vulnerable to a *dictionary attack*. A dictionary attack is an automated, brute force, assault that uses common usernames as well as words from a dictionary list to guess your router's username and password. This is a common method of attack that requires no knowledge on the part of the system cracker. Software automates the process, so even a 10-year-old kid could launch an attack! So when coming up with names, try to think of ones that make sense to you but aren't easy to guess.

Note Cisco usernames and passwords are always case sensitive.

There are two steps to creating a user on a Cisco router. The first step is to create a username and assign the user a privilege level. A *privilege level 15* will grant a user direct access to *privileged EXEC mode* when they log in to the router. A user who is assigned *privilege level 1* has *unprivileged* access only (user EXEC mode) to the router. In large installations, a privileged level 1 account can be used to allow a technician to audit and document some aspects of a router (or a lot of routers in very big installations) but would prevent the user from examining or changing a configuration. However, in the small business environment, privilege level 15 works well for managing a router.

The second step is to configure your VTY lines so only a local user (a user created on the router) with a valid password can gain access to the router. When router administrators use telnet or SSH to access the router, they will be prompted for a username and password. Listing 5-1 shows how to accomplish both of these steps.

Listing 5-1. *How to Create a Local User*

```
lab-r1> ena
lab-r1# config t
lab-r1(config)# username CiscoAdmin privilege 15 secret P@$word!
lab-r1(config)# line vty 0 4
lab-r1(config-line)# login local
lab-r1(config-line)# exit
lab-r1(config)#
```

Step 1: Create a User and Password

Use the global configuration command `username` to create a user. In this example, we'll assign the name "CiscoAdmin" to the new user. The statement `privilege 15` grants full access to the router and will place the user directly into privileged EXEC mode when they log in to the router. The next statement, `secret P@$word!`, will set the user password to "P@\$word!" and encrypt it on the router.

Tip Many of the password commands in the Cisco IOS include the *secret* modifier. This modifier will encrypt those passwords on the fly, rather than store them in clear text in your router configuration.

Step 2: Set the Login to Local

As shown in Listing 5-1, use the global configuration command `line vty 0 4` to start VTY configuration mode. Next, issue the command `login local`. When login is set to local, the router requires a username be provided before granting access to the router via VTY.

Configuring Secure Shell (SSH)

Telnet is a great tool to get you up and running, but it is inherently insecure. Telnet sends all information between you and the router in clear text. This means that some clever soul may be able to intercept that information (your username and password for example) and use it to gain access to your router.

The standard tool used for establishing a secure connection to a Cisco router is *Secure Shell* (SSH). SSH uses a cryptographic algorithm to encode information, making it difficult for someone to intercept your username, password, or other information. As of this writing, there are two versions of SSH (version 1 and version 2). I will cover how to configure the newer and more secure version 2. SSH version 2 (SSHv2) requires a 1024-bit key or greater. If your key is smaller—for example, 512 bits—then you can only use SSHv1. Also, you *must* assign a hostname and domain name to your router before you can generate your cryptographic keys because they become part of the key set.

Step 1: Generate the RSA Keys

As mentioned earlier, you must assign a hostname and domain name to your router before you can generate your encryption keys. You have already configured your router with the hostname “lab-r1” but still you need to assign a domain name. In Listing 5-2 the domain name has been set to “geekvenue.local” using the global configuration mode command `ip domain-name geekvenue.local`. Next, the command `crypto key generate rsa` generates the RSA crypto keys.

Listing 5-2. *How to Generate RSA Keys for SSH*

```
lab-r1> ena
lab-r1# config t
lab-r1(config)# ip domain-name geekvenue.local
lab-r1(config)# crypto key generate rsa
```

To use SSHv2 you must set the key size to 1024 bits or greater.

The name for the keys will be: lab-r1.geekvenue.local
Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose Keys. Choosing a key modulus greater than 512 may take a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

Step 2: Set the VTY Transport Input Type

This is an optional VTY configuration that forces all VTY connections to your router to use the SSH protocol. The VTY configuration statement `transport input ssh`, used in Listing 5-3, forces all VTY connections to use SSH exclusively as the communication protocol. To allow both telnet and SSH connections, use the configuration statement `transport input telnet ssh` instead.

Listing 5-3. *How to Set the VTY Transport Input Type for SSH*

```
lab-r1> ena
lab-r1# config t
lab-r1(config)# line vty 0 4
lab-r1(config-line)# transport input ssh
lab-r1(config-line)# exit
```

Step 3: Use SSH to Log in to the Router

At this point you have a fully functional SSH server listening for incoming connections on your router. This means you can use an SSH client on your workstation to connect to the router. If you are a Windows user, I recommend the Putty SSH GUI client. Putty is a free download and works very well. If you are a UNIX user, an SSH client is nearly always installed with your operating system. The command to connect via SSH from a UNIX command line is

```
ssh -l {username} {ip-address}
```

In the following example, the router's IP address is 192.168.1.1, and the user you created was "CiscoAdmin". Therefore, to log in to the router from a UNIX workstation, you would use the command

```
$ ssh -l CiscoAdmin 192.168.1.1
```

At the password prompt, enter the password assigned to your user, "CiscoAdmin", and you will be logged on to the router in privileged EXEC mode. As you can see in the following sample output, you may receive an "authenticity" warning that can safely be

ignored. This warning indicates that the domain name used to generate the keys cannot be verified. Since you trust yourself, and you generated your router's RSA keys, this is not a problem and you can safely add the key to your SSH client's known hosts list by replying "yes."

```
The authenticity of host '192.168.1.1 (192.168.1.1)' can't be established.  
RSA key fingerprint is af:6f:42:47:43:6a:b1:dd:52:d4:c4:74:6c:b1:8e:ea.  
Are you sure you want to continue connecting (yes/no)? yes
```

```
Warning: Permanently added '192.168.1.1' (RSA) to the list of known hosts.  
CiscoAdmin@192.168.1.1's password: *****
```

```
lab-r1#
```

Recovering a Lost Password

Once you have secured your router with passwords, you have the responsibility of keeping those passwords in a safe place and not losing them. But what happens if you do lose a password? Not to worry, Cisco has developed a method that allows you to recover your router's configuration and reset the password(s). I have found that this process intimidates a lot of people. But overall, it's straightforward and easy to perform; all you need to do is be careful.

Overview of the Process

Begin by attaching your console cable to the router's console port and starting your terminal emulation program (i.e., Hyper Terminal).

Next, power-off and restart your router. While the router is booting, send it a *Ctrl+Break* signal. This will stop the IOS from loading and instead starts something Cisco calls *ROM Monitor Mode* (rommon mode). Rommon allows you to change settings in your router's configuration registers.

Once in rommon mode, you set a register in the router so the IOS will load on power-up, but the startup configuration, "startup-config", will not. When you restart the router, the IOS will load without a configuration, providing you access to the IOS. Once you're in the IOS, you can copy the "startup-config" (the configuration with the lost password) into memory and change the forgotten password. It's really quite clever.

After changing the password, you save the modified startup configuration to the "startup-config", and then reset the configuration register so the startup configuration will again load when you next power-up the router.

Finally, restart the router and log in—only this time you'll know what your password is!

Step 1: Bypass the IOS

Plug a console cable into the router's console port and launch your terminal application. Next, power-off and restart your router. While the router is restarting, send it a Ctrl+Break (you may need to send several for it to catch), as shown in the following example:

```
System Bootstrap, Version 12.2(8r)YN, RELEASE SOFTWARE (fc1)
TAC Support: http://www.cisco.com/tac
Copyright (c) 2002 by cisco Systems, Inc.
C800/SOH0 series (Board ID: 28-130) platform with 32768 Kbytes of main memory

{send a Ctrl+Break}

rommon 1>
```

To send a *Ctrl+Break* using Hyper Terminal, hold down the control key and tap on the break key until you see the rommon prompt, `rommon 1>`. If you are using some other terminal emulation software, you may need to send a Ctrl+Break using a menu option or some other method (refer to the application's documentation or UNIX man page).

Step 2: Modify the Configuration Register

Listing 5-4 shows how to use rommon mode to set the configuration register to 0x2142. This instructs the IOS not to load the startup configuration after the IOS loads. Once you see the rommon prompt, type the command **confreg 0x2142** then press the ENTER key. Next, type **reset** then press the ENTER key again. The IOS will now load, but your router's startup configuration will not.

Listing 5-4. *How to Set the Configuration Register to Bypass the Startup Configuration*

```
rommon 1> confreg 0x2142
You must reset or power cycle for new config to take effect
rommon 2> reset
```

Step 3: Copy the Configuration and Reset Passwords

After invoking reset, your router will load the IOS but will not load a configuration into memory. Therefore, the following prompt will ask if you would like to enter the initial configuration dialog:

```
Would you like to enter the initial configuration dialog? [yes/no]:
```

Answer **NO** to this question and you will receive the user EXEC mode command prompt. From there you can use the enable command to enter privileged EXEC mode, as shown in Listing 5-5.

All you need to do now is copy your startup configuration into memory using the privileged EXEC mode command `copy startup-configuration running-configuration` (abbreviated `copy start run`). You should immediately notice that your command prompt changes to reflect the hostname assigned by the router's original startup configuration. In Listing 5-5 it goes from the Cisco default of "Router" to "lab-r1".

Now that you have the startup configuration in running memory, you can make changes to it. You will now want to change the forgotten password. The most common password lost is the privileged EXEC mode password (sometimes referred to as the "enable password") but it could be a user password or VTY password. The process is the same regardless of which password is lost. Listing 5-5 illustrates how to change the privileged EXEC mode password.

Listing 5-5. How to Start the IOS and Modify Your Startup Configuration

```
Router> ena
Router# copy start run
Destination filename [running-config]? {press ENTER}
2071 bytes copied in 2.952 secs (702 bytes/sec)

lab-r1# config t
Enter configuration commands, one per line. End with CNTL/Z.
lab-r1(config)# enable secret {YourNewPassword}
lab-r1(config)# exit
lab-r1# copy run start
```


After you have changed your passwords and recorded them in a secure location, you issue the privileged EXEC mode command `copy run start` to save your modified running configuration as the startup configuration on your router. You will be prompted for the destination filename.

```
Destination filename [startup-config]?
```

Press the ENTER key to save the configuration using the default name “startup-config”. Once you have modified and saved your startup configuration, you need to reset the configuration register so the IOS will automatically load the startup configuration again when you power up the router.

Step 4: Reset the Configuration Register

After you reset your password and save your configuration, use the global configuration command `config-register 0x2102` to reset the router’s configuration register to its original setting, as shown in Listing 5-6.

Listing 5-6. *How to Set the Configuration Register to Load the Startup Configuration*

```
lab-r1# config t
lab-r1(config)# config-register 0x2102
lab-r1(config)# exit
```

Use the `show version` command to verify the change made to the configuration register.

```
lab-r1#sh ver
```

Note from the following output, that this command displays the configuration register’s current setting and how it will be configured the next time you restart the router or issue the `reload` command.

```
Cisco IOS Software, SOH091 Software (SOH091-K9OY6-M), Version 12.4(5b), RELEASE ►
SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Wed 19-Apr-06 08:12 by ssearch
```

```
Configuration register is 0x2142 (will be 0x2102 at next reload)
```

That's all there is to recovering a lost password! It may seem a little intimidating, but it's really quite simple. If it's your router, you may want to maintain a password list locked in a safe to prevent this from happening to you. However, if you're an IT consultant, you may run into this situation at a client site and you will want to be able to perform this task with confidence. Therefore, I recommend practicing it a few times in your lab. In either case, it's a good skill to have.

Upgrading the IOS

One day you may find that you need to upgrade your router's IOS. Cisco, like any other vendor, is always improving its products by fixing security issues and adding new features. Cisco partners provide various annual service contracts, like *SmartNet*, that include IOS upgrades. If you're using a Cisco router in a small business, it's a good idea to purchase a SmartNet contract for your production router. They are relatively inexpensive, guarantee expedited replacement of failed hardware, provide access to the Cisco *Technical Assistance Center* (TAC), and will also provide you with access to IOS upgrades as they become available. Unlike other vendors, Cisco's TAC has skilled engineers who are very responsive in dealing with problems. One of the nice features of TAC is that it is available 24 hours a day, seven days a week. The only caveat is that you must have an annual contract before they provide that support.

The steps to upgrading the IOS are as follows:

1. Display the name and size of the existing IOS.
2. Back up the existing IOS image to a server.
3. Delete the old IOS image from flash.
4. Install the new IOS image file to flash.
5. Restart your router.

Note Cisco's TAC requires an annual contract before it provides support. A SmartNet contract can only be purchased from a Cisco Partner; it cannot be obtained from Cisco directly.

Step 1: Display the Contents of Flash Memory

Before you upgrade your IOS, you must first display the contents of your router's flash memory using the privileged EXEC mode command `show flash`. You do this so you can see the name of the currently installed IOS image file. You will need to know that name

before you can back up the image file to a server. On an 850 or 870 series router, flash memory contains the IOS image file, web-based administration files, and the router's default configuration file. On older 800 series routers, like the 831, flash memory only contains the IOS image file. Listing 5-7 shows how to display the contents of your router's flash memory.

Listing 5-7. *How to Display the Contents of Flash Memory*

```
lab-r1> enable
lab-r1# sh flash
```

The following output shows the contents of flash memory from a Cisco 851 series router:

```
20480K bytes of processor board System flash (Intel Strataflash)
```

```
Directory of flash:/
```

```
 2  -rwx  10848384  --- -- ---- --:--:--  -----  c850-advsecurityk9-mz.124-4.T7.bin
 3  -rwx   2254    Feb 28 2002 15:07:13 -09:00  sdmconfig-8xx.cfg
 4  -rwx   833024  Feb 28 2002 15:07:29 -09:00    es.tar
 5  -rwx  1052160  Feb 28 2002 15:07:48 -09:00  common.tar
 6  -rwx   1038    Feb 28 2002 15:07:59 -09:00  home.shtml
 7  -rwx   102400  Feb 28 2002 15:08:11 -09:00  home.tar
 8  -rwx  1959936  Feb 28 2002 15:08:39 -09:00  wlanui.tar
 9  -rwx   242285  Feb 28 2002 15:08:54 -09:00  attack-drop.sdf
```

```
19353600 bytes total (4300800 bytes free)
```

The IOS image file is named “c850-advsecurityk9-mz.124-4.T7.bin”, and it is using 10,848,384 bytes (approximately 10 MB) of flash memory. As you can see from the display, the router has 4,300,800 bytes (approximately 4.3 MB) of flash memory free. This is important because you now know that the maximum size of a new image file can be no larger than 14.3 MB.

ABOUT IOS IMAGE NAMES

Let's look at the IOS image name "c850-advsecurityk9-mz.124-4.T7.bin" in more detail. Cisco image names provide a lot of information about the image itself. In this example, you know this is an image for a Cisco 850 series router as denoted by "c850" in the filename. (This image happens to be from a Cisco 851 router.)

The next bit of information "advsecurityk9" tells us there is an advanced security feature set compiled into the IOS. "K9" is Cisco's designation for 3DES/AES security (introduced in Chapter 4). The next part "mz" indicates the file has been mziped (compressed). IOS images are compressed in order to use less space in the router's flash memory. The next information in the filename, "124-4T7", is the actual version of the IOS. "124" indicates that the major release number is "12.4". The next number, "4", indicates the individual release number and "T7" is the new feature release identifier. In this case, the "T" stands for "technology" (Cisco has other feature release identifiers, but this one is very common). The "7" is the number feature identifier. The final suffix for the image is ".bin", which indicates this is a binary file. Cisco folks typically refer to an image such as this one as either "12.4T7" or "12.4 tech 7".

Most of the time this information can be ignored, but it's a good idea to look at it before you install a new IOS. This way you can compare your old and new image files to at least make sure that your new IOS is the correct version for your router. If you install the wrong version (e.g., an IOS designed for some other router) you may hose your router and you don't want that!

Step 2: Back Up the Existing IOS Image File

You should always have a backup copy of your current IOS image on hand. This is especially true if you're about to perform an upgrade of the Cisco IOS on your router. It is important in case something goes wrong with the upgrade. Perhaps the new IOS is corrupt or buggy for your environment. Maybe it's the wrong IOS for your router (let's hope not). At any rate, it's always a good idea to have a way to restore service quickly to your router.

The standard method used to back up files from a Cisco router to a server is the *File Transfer Protocol* (FTP). FTP is a client/server application that allows you to transfer files from a client (your router in this case) to an FTP server that resides on a PC. There are two types of FTP servers: the older version, called *Trivial File Transfer Protocol* (TFTP), and the newer version, called FTP. The main difference between these two protocols is that TFTP does not require any authentication and thus has very little security. On the other hand, FTP requires the client provide authentication before allowing access to the FTP server.

Almost from Cisco's inception, TFTP has been the standard method used for backing up Cisco routers. Cisco even provides a free TFTP server you can download. However, in the past few years, FTP has become the new standard due to the added security. As good as the security is I've found that Cisco's FTP client can have difficulty working with some FTP servers; therefore, I'm going to show you how to use both methods of backing up your Cisco router files.

Method 1: Back Up to a TFTP Server

Begin by copying your IOS image to a TFTP server. You can download a TFTP server from the Web if you run Windows; if you're a UNIX user, it's likely that your UNIX OS has one built in. Once the server is up and running, you use the IP address of the server as the target address for your backup.

Listing 5-8 uses the privileged EXEC mode command `copy flash:{image-name} tftp` to back up your IOS image to a TFTP server.

Listing 5-8. How to Copy the IOS Image from Flash to a TFTP Server

```
lab-r1> enable
lab-r1# copy flash:/c850-advsecurityk9-mz.124-4.T7.bin tftp
Address or name of remote host []? 192.168.1.101
Destination filename [c850-advsecurityk9-mz.124-4.T7.bin]? {press ENTER}
```

Notice from the following output that exclamation points march across the screen to indicate the successful progress of your backup. When the backup is finished copying the file, the size of the file and time it took to copy the file to the server is displayed on the screen.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!

10848384 bytes copied in 43.880 secs (247228 bytes/sec)
```

It's important you know that *TFTP is insecure by nature*, because it uses no authentication. Therefore, I prefer to run the TFTP server on my PC just long enough to back up the IOS then disable it again when I'm finished.

Method 2: Back Up to an FTP Server

A better way to back up your router files is with FTP, because it allows you to store your information on a server that requires a username and password before the information

can be accessed. The process for backing up files with FTP is similar to using TFTP, except that you must provide log in credentials.

■ **Tip** A Macintosh computer running OSX has an FTP server preinstalled by default. Go to “System Preferences”, click the “Sharing” folder, and place a check mark next to “FTP Access”. To stop the FTP server, simply uncheck FTP Access.

The commands in Listing 5-9 show how to copy the IOS image to an FTP server.

Listing 5-9. *How to Copy the IOS Image from Flash to an FTP Server*

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# no ip ftp passive
lab-r1(config)# exit
lab-r1# copy flash:/c850-advsecurityk9-mz.124-4.T7.bin ➡
ftp://jason:MyFtpPassword@192.168.1.101
Address or name of remote host [192.168.1.101]? {press ENTER}
Destination filename [c850-advsecurityk9-mz.124-4.T7.bin]? {press ENTER}
```

The first thing you may have noticed in Listing 5-9 is that I made a configuration change to the Cisco FTP client using the global configuration mode command `no ip ftp passive`. FTP can operate in two modes, *passive* and *active*. Use passive FTP when moving files through a firewall and active FTP when moving files locally. Since you’re copying your files to a local FTP server installed on a workstation that resides on your private network, you want to use active FTP for better client/server compatibility.

Next, use the privileged EXEC mode command `copy flash:{image-name} ftp://{ftp-username}:{ftp-password}@{ip-address}` to copy the IOS image to an FTP server.

■ **Tip** The Cisco FTP client has compatibility issues with some FTP servers. If you have trouble copying files to your FTP server, try both passive and active FTP. Otherwise, use TFTP, which seems to work more consistently.

Step 3: Delete the Old IOS Image

After you have successfully made a backup copy of your old IOS, you can delete it from the router’s flash memory (making room for the new version). Listing 5-10 uses the privileged EXEC mode command `delete flash:{image-name}` to remove an image from flash memory.

Listing 5-10. *How to Delete the Old IOS Image File from Flash*

```
lab-r1> enable
lab-r1# delete flash:/c850-advsecurityk9-mz.124-4.T7.bin
Delete filename [c850-advsecurityk9-mz.124-4.T7]? y
```

Caution When you answer “y” to the “Delete filename” dialog, your currently installed IOS image will be deleted from the router. This is irreversible so be sure you have a backup of the image before you delete it from your router.

Step 4: Install the New IOS Image

After making room in flash memory for the new IOS, copy the updated IOS image from your server to your router’s flash memory. You want to do this right away. Now is not the time to go to lunch or make a phone call!

Caution You do not want to issue the `reload` command or power-cycle your router until the new image file has been copied to your router’s flash memory. Without an image file in flash, your router cannot boot the IOS. Also, if you install an image with the wrong IOS version (e.g., an IOS designed for another router model) your router may not boot. So be careful during the next few steps!

Method 1: Copy from a TFTP Server

Listing 5-11 shows the commands to copy the IOS image file from a TFTP server to your router’s flash memory.

Listing 5-11. *How to Copy the New IOS from a TFTP Server to Flash Memory*

```
lab-r1> enable
lab-r1# copy tftp flash
Address or name of remote host []? 192.168.1.101
Source filename []? c850-advsecurityk9-mz.124-4.T9.bin
Destination filename [c850-advsecurityk9-mz.124-4.T9.bin]? {press ENTER}
```

As you can see from the following output, it's similar to the information you receive when copying to or from a server, regardless of whether you're using FTP or TFTP:

```

Accessing tftp://192.168.1.101/c850-advsecurityk9-mz.124-4.T7.bin...
Loading c850-advsecurityk9-mz.124-4.T9.bin from 192.168.1.101 (via BVI1):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

[OK - 10848384 bytes]

10848384 bytes copied in 126.708 secs (85617 bytes/sec)

```

Method 2: Copy from an FTP Server

To install an IOS image residing on an FTP server, you will want to issue the privileged EXEC mode command `copy ftp://{ftp-username}:{ftp-password}@ flash` as shown in Listing 5-12.

Listing 5-12. *How to Copy the New IOS from an FTP Server to Flash Memory*

```

lab-r1> enable
lab-r1# copy ftp://jason:MyFtpPassword@ flash
Address or name of remote host []? 192.168.1.101
Source filename []? c850-advsecurityk9-mz.124-4.T9.bin
Destination filename [c850-advsecurityk9-mz.124-4.T9.bin]? {press ENTER}
Accessing ftp://jason:MyFtpPassword@192.168.1.101/c850-advsecurityk9-mz.124-4.T9.bin
Erase flash: before copying? [confirm] {press ENTER}
Erasing the flash filesystem will remove all files! Continue?[confirm] {Press ENTER}

```

Note On older 800 series and SOHO models, you may be requested to erase flash memory before being allowed to copy the new IOS image to flash. You should press the ENTER key to answer “yes” to this question before proceeding.

Here is what the output looks like on a Cisco 831 series router when flash memory is being erased prior to copying the IOS image to the router:

```
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
Erase of flash: complete
Accessing tftp://192.168.1.101/c830-advsecurityk9-mz.124-4.T9.bin...
Loading c830-advsecurityk9-mz.124-4.T9.bin
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

[OK - 10848384 bytes]
10848384 bytes copied in 126.708 secs (85617 bytes/sec)
```

In Listing 5-12, the FTP user has been set to “jason” and the password for the user jason is “MyFtpPassword”. These authentication credentials are set when you install the FTP server on your PC. In this example, the IOS image name is “c850-advsecurityk9-mz.124-4.T9.bin”.

Tip You can combine the FTP username, password, and image name into a single command from privileged EXEC mode using `copy tftp://{ftp-username}:{ftp-password}@{ip-address}/{ios-image} flash`.

Step 5: Boot the New Image

The simplest way to have your router load the newly installed IOS is to cycle the power on your router. However, you can also issue the privileged EXEC mode command `reload`, which will cause the router to load the new IOS.

```
lab-r1# reload
Proceed with reload? [confirm] {press ENTER}
```

Tip After restarting your router, you can issue the `show version` command to verify that the new IOS is installed and loaded correctly.

DEALING WITH MULTIPLE IMAGES

Some Cisco router models support storing more than one image in flash memory. In this case, you will need to notify the router which IOS image file to load. This is unnecessary if you're using an 800 series router, but it's good information to keep under your hat just in case. Who knows, someday you may get to sink your teeth into some high-end Cisco gear!

If you have multiple images stored in flash, use the global configuration mode commands `no boot system` and `boot system flash {image-name}` to define the name of the IOS image file to load on startup.

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# no boot system
lab-r1(config)# boot system flash c850-advsecurityk9-mz.124-4.T9.bin
lab-r1(config)# exit
lab-r1# reload
```

Backing Up Your Configuration

Just like the data on your file, database, and e-mail servers, the data on your router should be backed up. There are several ways to back up the configuration and I'm going to cover three of them here: copying the startup configuration to your router's flash memory, copying it to a TFTP server, and copying it to an FTP server.

Method 1: Back Up to Flash Memory

The simplest way to back up your router's running configuration is to copy it to the router's flash memory as shown in Listing 5-13. Most configurations, particularly for a small business, will occupy very little space in flash memory, which allows you to store multiple copies. Use the privileged EXEC mode command `copy run flash` to copy the running configuration to the router's flash memory. The default filename is "running-config" but it can be renamed by typing in a different name at the "Destination filename" prompt. Use the command `show flash` to verify it was copied.

Listing 5-13. *How to Copy the Running Configuration to Flash Memory*

```
lab-r1> enable
lab-r1# copy run flash
Destination filename [running-config]? {Press Enter}
lab-r1# sh flash
```

You can see in the following output from the `sh flash` command that there is now a copy of your router's running configuration stored in the flash directory of your router.

```
Directory of flash:/
 2  -rw- 10848384  --- -- ---- --:--:--  ----- c850-advsecurityk9-mz.124-4.T7.bin
 3  -rw-   2254    Feb 28 2002 15:10:50 -09:00 sdmconfig-8xx.cfg
 4  -rw-   861696  Feb 28 2002 15:11:06 -09:00 es.tar
 5  -rw-  1164288  Feb 28 2002 15:11:25 -09:00 common.tar
 6  -rw-   1038    Feb 28 2002 15:11:36 -09:00 home.shtml
 7  -rw-  113152   Feb 28 2002 15:13:08 -09:00 home.tar
 8  -rw-  1959936  Feb 28 2002 15:13:35 -09:00 wlanui.tar
 9  -rw-   10545   May 26 2008 20:12:29 -09:00 running-config
```

A better alternative is to back up your configuration to a server. If your router mysteriously dies, a copy of your configuration will be safe and secure.

Method 2: Back Up to a TFTP Server

Using your router's flash memory is fine for grabbing a quick backup of your configuration, but what happens if the router hardware fails? Your configuration could be lost forever and difficult to recreate. That's why it's a good idea to keep a copy of the configuration somewhere else. You can use the same FTP or TFTP server that you used earlier to back up the IOS to store a copy of your router's configuration. Listing 5-14 shows how to copy your running configuration to a TFTP server.

Listing 5-14. *How to Copy Your Running Configuration to a TFTP Server*

```
lab-r1> enable
lab-r1# copy run tftp
Address or name of remote host []? 192.168.1.101
Destination filename [lab-r1-config]? {press ENTER}
```

As you can see from the following output, configuration files are very small. This configuration uses 10,545 bytes of memory (10.5 Kb).

```
!!!!
10545 bytes copied in 1.416 secs (7447 bytes/sec)
```

Method 3: Back Up to an FTP Server

Alternatively, you can copy the running configuration to an FTP server using the global configuration command `copy run ftp://{ftp-username}:{ftp-password}@{ip-address}`. Listing 5-15 presents the commands that copy your router's running configuration to an FTP server.

Listing 5-15. How to Copy Your Running Configuration to an FTP Server

```
lab-r1> enable
lab-r1# copy run ftp://jason:MyFtpPassword@192.168.1.101
Address or name of remote host [192.168.1.101]? {press ENTER}
Destination filename [lab-r1-config]? {press ENTER}
```

The following is the output from this command:

```
Writing lab-r1-config !!!
10545 bytes copied in 2.748 secs (3997 bytes/sec)
```

That's all there is to making a copy of your router's configuration. It's the little things that are important and yet, somehow, often are forgotten. Having a backup of your router's configuration can expedite problem solving and quickly get your router back in service in the event that your router ever needs to be replaced due to a hardware problem or upgrade. That's why I recommend you get into the habit of making a backup every time you make a change to your configuration.

SAVING THE FTP USERNAME AND PASSWORD

You can store your FTP username and password in your router's IOS configuration using the global configuration commands `ip ftp username {ftp-username}` and `ip ftp password {ftp-password}`. When your FTP credentials are stored in your configuration, you can use the shorter command `copy run ftp` and you will be prompted for the address of your FTP server as in the following example:

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# no ip ftp passive
lab-r1(config)# ip ftp username jason
lab-r1(config)# ip ftp password MyFtpPassword
lab-r1(config)# exit
lab-r1# copy run ftp
Address or name of remote host []? 192.168.1.101
Destination filename [lab-r1-config]? {press ENTER}
```

Tuning Your ACLs for Performance

ACL performance tuning is probably not necessary for routers in the small business arena where rules are relatively short and simple. By contrast, larger installations with complex networking needs may have dozens or even hundreds of rules in their ACLs. That's when ACL tuning can really improve a router's performance. However, this doesn't mean that small business administrators should completely overlook it.

Step 1: Display ACL Rule Matches

Performance tuning consists of displaying your ACL rules and establishing how many times each permit rule has been matched. In Listing 5-16, we use the command `sh ip access-list IPFW-ACL` to display all the rules and matches for the ACL named "IPFW-ACL."

Listing 5-16. *How to Display Your ACL List Matches*

```
lab-r1> enable
lab-r1# sh ip access-list IPFW-ACL
```

The following output displays the matches for all the rules in your firewall ACL named "IPFW-ACL" that is used throughout this book:

```
Extended IP access list IPFW_ACL
 10 permit tcp any host 66.238.25.254 eq www (724 matches)
 20 permit tcp any host 66.238.25.254 eq smtp (9629 matches)
 30 permit tcp any host 66.238.25.254 eq 443 (32 matches)
 40 permit icmp any any administratively-prohibited (321 matches)
 50 permit icmp any any echo-reply
 60 permit icmp any any packet-too-big
 70 permit icmp any any time-exceeded (85 matches)
 80 permit icmp any any traceroute
 90 permit icmp any any unreachable (22 matches)
100 deny ip any any log (19918 matches)
```

The concept is simple: you want the rules with the highest number of matches to be at or near the top of your ACL list. That way the matches can be processed more quickly without waiting for preceding rules to be examined. Remember, ACLs are processed in the order in which they are created. The rule at the top of the list is processed first and the rule at the bottom is processed last.

Note that the SMTP e-mail server rule has the highest number of permit matches. You know this because of two bits of information:

- SMTP is the simple mail transport protocol (an e-mail server).
- It has 9629 matches, which is more than any other permit rule.

Note The last rule, “deny ip any any log”, *denies and logs all traffic* to your router and will typically have more matches than any other rule used in the WAN ACL “IPFW-ACL”. However, this rule must remain at the end of your ACL to ensure that no unwanted traffic gets through. It’s your main line of defense against bad guys.

Now that you have collected some information about your ACL matches, you need to reorder the ACL rules so they work more efficiently.

Step 2: Reorder the ACL Rules

After collecting some information (a baseline) about matches in your WAN ACL “IPFW-ACL”, you need to reorder the ACL rules so the most frequently matched rules are placed at the beginning of the list.

Note The IOS does not provide a simple method to edit ACL rules. To modify an ACL, you must delete the existing ACL and recreate it using the new rules. The easiest way to do this is to copy the rules from your terminal emulation software and paste them into a text file. Then reorder the rules in the text file and paste them back into the terminal. If your terminal application does not support cut and paste, you will need to retype your ACL rules.

As an example, take a look at the following show command and its output:

```
lab-r1> enable
lab-r1# sh ip access list IPFW-ACL
```

Notice from the following output that the rules in the ACL have been reordered to reflect the most often matched rules. In this case, the SMTP rule has been moved near the top of the list. Also, a new rule has been added to the top of the list, the *established* rule permit tcp any host 66.238.25.254 established.

```
ip access-list extended IPFW-ACL
permit tcp any host 66.238.25.254 established
permit tcp any host 66.238.25.254 eq smtp
permit tcp any host 66.238.25.254 eq www
permit tcp any host 66.238.25.254 eq 443
permit icmp any any administratively-prohibited
permit icmp any any echo-reply
permit icmp any any packet-too-big
permit icmp any any time-exceeded
permit icmp any any traceroute
deny ip any any log
```

Step 3: Apply the Established Rule

Established connections are connections that are *replying* to a host on your network. An example of this would be when an e-mail server on the Internet is delivering a message to your company e-mail server. When an e-mail server is sending a message, there is a TCP connection that is set up and used until the message is successfully delivered to the receiving server. The established rule (placed at the top) allows the already established session to take place without being processed by all the other rules in your ACL. The nice thing about this rule is that it applies to *any* established TCP connection regardless of the protocol. This means that it will be used for SMTP, SSH, WWW or any other connection-oriented protocol that has established a connection with your network.

Place the `permit tcp any host {ip-address} established` command at the top of your ACL (use the static IP address assigned to you by your ISP).

Before settling on an ACL order that most efficiently works for your network traffic, you may need to generate a new baseline by monitoring your network activity for a while. I like to check my router's traffic on a monthly basis. However, the established rule should always go at the top of the list.

Protecting Your Passwords

When you create passwords using the IOS, many of the passwords are stored in clear text by default in your router's configuration file. Cisco provides a single, global configuration mode command, `service password-encryption`, which will encrypt all the passwords, ensuring that they do not display when issuing the `show run` command. After issuing this command, your passwords display as a string of characters (e.g., 80471A1597547C515837590F) in your configuration when you issue the `show run` command. It's a good idea to run the global configuration mode command `service password-encryption` shown in Listing 5-17 on all your routers. This ensures that your

passwords are not compromised in the event that someone has access to your configuration file.

Listing 5-17. *How to Enable Automatic Password Encryption*

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# service password-encryption
lab-r1(config)# exit
lab-r1#
```

Disabling Show and Tell

The *Cisco Discovery Protocol* (CDP) is a Layer 2 protocol that allows Cisco routers and switches to share information with each other about their hardware platforms and configurations. Because CDP operates at the data link layer, Layer 2 of the *Open Systems Interconnection* (OSI) reference model, it can communicate this information without the need for TCP/IP. However, CDP does share your router's IP address and you may not want your router sharing that or any other information with its neighbors. Therefore, it's a good idea from a security standpoint to always disable CDP on your router.

Listing 5-18 shows how to use the global configuration mode command `no cdp run` to globally disable the Cisco Discovery Protocol on your router. This will prevent your router from sending or receiving CDP updates on all interfaces.

Listing 5-18. *How to Disable the CDP*

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# no cdp run
lab-r1(config)# exit
lab-r1#
```

Safeguarding Your E-mail Server

The Cisco IOS firewall can be configured to inspect POP3, SMTP, and IMAP traffic *inbound* on your WAN interface. Cisco refers to this as the *Email Inspection Engine* (EIE).

Because Cisco's firewall operates at the application layer, it knows which commands are valid for the application being inspected. Take the e-mail protocol SMTP, for example. You may not realize it, but when e-mail is delivered to your e-mail server, there's a conversation that takes place between the sender and your server. The dialog uses commands, such as HELO, DATA, SEND, VRFY, and QUIT. If you create a firewall that inspects SMTP traffic and a system cracker tries to send invalid commands to your SMTP server (usually

in an attempt to crash it), the IOS firewall will recognize those commands as invalid and will drop the packets and clear the connection.

The IOS firewall also knows about authentication. If you're running an IMAP or POP3 server, the firewall will inspect the authentication commands and drop a connection if it sees any funny business. In Listing 5-19 the EIE is being configured to inspect SMTP, POP3, and IMAP:

Listing 5-19. *How to Configure the EIE*

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# ip inspect name EMAIL-FW smtp
lab-r1(config)# ip inspect name EMAIL-FW smtp max-data 10000000

lab-r1(config)# ip inspect name EMAIL-FW imap
lab-r1(config)# ip inspect name EMAIL-FW imaps

lab-r1(config)# ip inspect name EMAIL-FW pop3
lab-r1(config)# ip inspect name EMAIL-FW pop3s

lab-r1(config)# int fa4
lab-r1(config-if)# ip inspect EMAIL-FW in
lab-r1(config-if)# exit
lab-r1(config)# exit
lab-r1#
```

Step 1: Name the EIE Firewall

The first step is to define a name for the firewall using the global configuration command `ip inspect name`. Since this firewall is different from the firewall named “IPFW” introduced in Chapter 2, you need to give it a different name. I’ve chosen to name this firewall “EMAIL-FW”, but you can use any name that makes sense to you.

Step 2: Define the Protocols

For the Cisco EIE to work, you need to properly configure the protocols that will be inspected. As you can see from the example in Listing 5-19, SMTP has an optional parameter, `max-data`, that defines the maximum size of an e-mail message that is allowed to be delivered to your server. The default is 20 MB, but in this example, it has been reduced to 10 MB using the `max-data` parameter. As of this writing, you can use SMTP, ESMTP, POP3, POP3S, IMAP, IMAP2, IMAP3, and IMAPS protocols with the Cisco EIE.

Step 3: Apply the Firewall

The last thing you need to do is apply your firewall *inbound* on the WAN interface of your router. You can only have one inbound and one outbound inspection firewall per interface. In Listing 5-19 the firewall named EMAIL-FW has been applied inbound to the WAN interface fa4 with the `ip inspect EMAIL-FW in` command.

Configuring a Logging Host for Intrusion Detection

Cisco routers are equipped with a *System Logger* (syslog client). Syslog is a client/server-based protocol used to send logging messages over a TCP/IP network to a server application that records messages for review. There are eight different levels of messages that can be recorded. These levels are often referred to as *traps*. By recording different types of messages and their frequency, you can create an *Intrusion Detection System* (IDS) that will help make you aware of when your router is under attack by a system cracker.

This is how it works: using the syslog protocol, your router sends logged information over your network to a host running a syslog server application as a service (a service is sometimes referred to as a daemon). The syslog application will record the events and allow you to set *event thresholds*. After a threshold has been met, the syslog application will then e-mail you an alert. At that point, it's your responsibility to look through the logs to find out what's happening. It could be that someone is attempting to gain access to your router directly, to gain access to a server, to crash a server on your network, or some other malicious activity.

If you have a Linux or a UNIX box, you can use its built-in syslog daemon called a *local logging facility* to capture logs from your router. By default, Cisco uses the "Local7" facility for logging. Personally, I'm a big fan of the Kiwi Syslog Daemon and feel that it warrants some explanation.

The Kiwi Syslog Daemon comes in two versions: a free version and a commercially licensed version. The commercial version has some nice features, but the free version also does a nice job and works well for intrusion detection. Unfortunately, it only runs on a Windows box, but I like it so much I can easily overlook that. What I like about Kiwi is that it runs as a service, logs any trap level you define, e-mails you daily reports with a running average of events, and allows you to set a threshold for any funny business going on with your router. If that threshold is reached, Kiwi alerts you via e-mail.

Step 1: Perform Basic KIWI Set Up

The first part to setting up Kiwi is to configure a new rule that will work with your Cisco Router. Figure 5-1 displays the first setup screen for Kiwi.

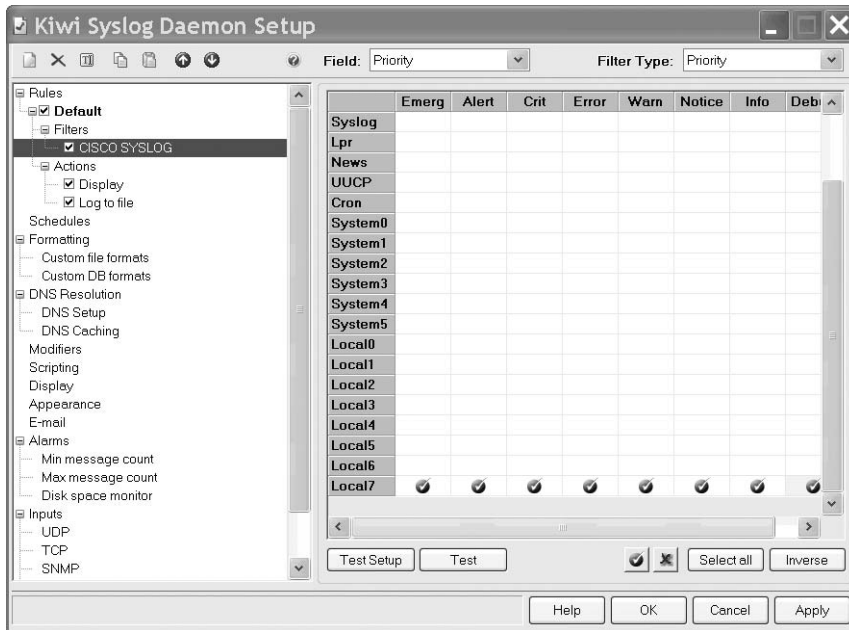


Figure 5-1. *Kiwi setup screen (Syslog Local7 for Cisco routers)*

Right-click “Filters” and select “New Filter”. I have named my filter “CISCO SYSLOG”. The next step is to tell Kiwi what to filter on. Use the pull-down menu to select the field and filter type, and choose the “Priority” option.

Scroll down to the “Local7” priority and use your mouse to place a green **check mark** in each of the logging fields. When you’re done, click “Apply” and “OK”.

Next, you will want to set up an e-mail address to be used for daily logs and IDS alarms.

Step 2: Configure E-mail Alarms

Under “Rules” in Figure 5-2, scroll down and click “E-mail”. Use your network e-mail settings to configure the SMTP mail server and e-mail addresses that Kiwi will use to send logs and IDS alarms to and from. This will typically be the e-mail server on your network.

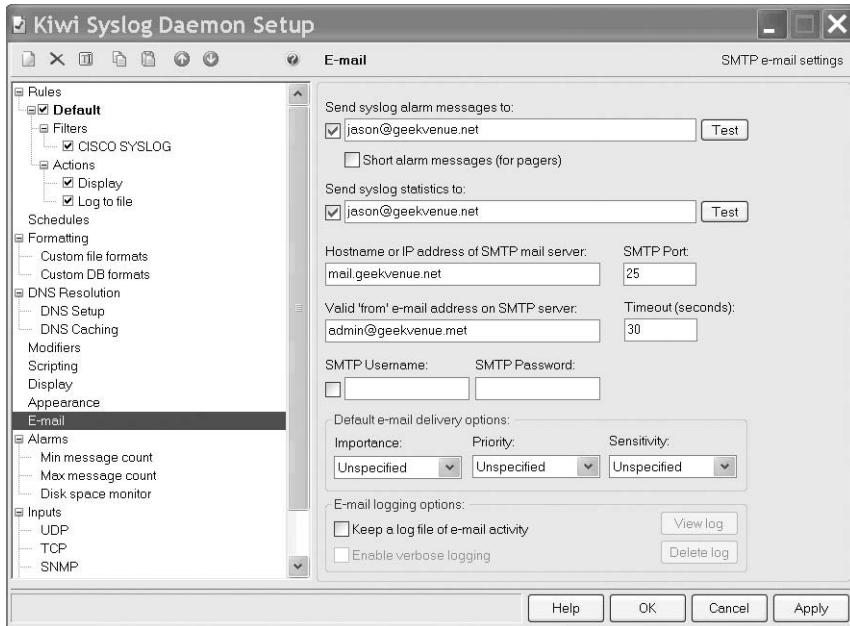


Figure 5-2. *Kiwi setup (e-mail options)*

Finally, you will need to set the alarm threshold that is used for intrusion detection. This is the number of times your firewall ACL “IPFW-ACL” logs the denial of TCP/IP activity. A sudden increase in logged denials indicates that someone may be port scanning or hacking your router.

Step 3: Set the Message Threshold

The goal is for you to come up with a baseline (an average) of the number of times per hour that your “IPFW-ACL” logs a deny. After you have established the baseline, you set the threshold above that number so you can tell if there’s any unusual activity on your router. To set the threshold, Click “Alarms” then “Max Message Count” to assign a threshold number. In Figure 5-3, I have set the threshold number to *100 messages per hour*. If more than 100 messages are logged per hour, I want to be alerted to the situation. Be sure to check the “Notify by e-mail” field.

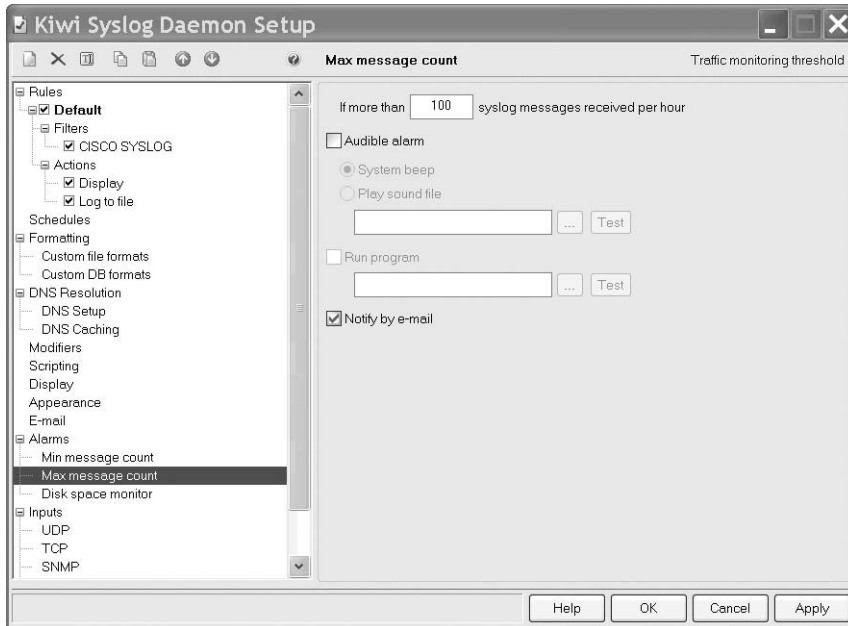


Figure 5-3. *Set the message threshold*

I chose 100 because it is twice as high as my baseline. Every day, Kiwi will send an e-mail message with logging statistics. You use those figures to come up with your baseline. The next output is an excerpt from the daily e-mail message sent by the Kiwi Syslog Daemon.

```

///      Kiwi Syslog Daemon Statistics      ///
-----
24 hour period ending on: Fri, 25 Jul 2008 00:00:03
Syslog Daemon started on: Thu, 10 Jul 2008 16:15:47
Syslog Daemon uptime:    14 days, 7 hours, 43 minutes
-----

+ Messages received - Total:                15599
+ Messages received - Last 24 hours:        1062
+ Messages received - Since Midnight:       1035
+ Messages received - Last hour:            43
+ Messages received - This hour:            23
+ Messages per hour - Average:              43

+ Messages forwarded:                       0
+ Messages logged to disk:                  1036

```

As you can see from this output, the logger has been running for 14 days, and the average number of messages per hour is 43. Therefore, 43 is our baseline for IDS. In Figure 5-3, I set the threshold to 100. If more than 100 messages are logged in one hour, I'll receive an e-mail alert.

Note You can set your threshold as loose or as tight as you like. I've seen systems with an average of 50 messages per hour suddenly spike, including one that logged over 1000 messages in an hour. In that particular case, the site was being probed by a port scan. Although it's difficult to track those guys down, I feel it's important to know when someone is rattling all the windows and doors on my router, looking for a way in. It forces me to check my security to ensure that they weren't successful.

Configuring Logging on Your Router

Before your router can begin logging events to your syslog server, there are a few configuration features that must be enabled on the router. Listing 5-20 shows the commands to enable the Cisco syslog client.

Listing 5-20. *How to Configure Logging on Your Router*

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# logging trap 7
lab-r1(config)# service timestamps log datetime localtime
lab-r1(config)# logging 192.168.1.2
```

Step 1: View Your Trap Levels

Use the global configuration command `logging trap ?` to display the trap options available on your router. By default, your router will log level 6 (informational) messages.

```
lab-r1# config t
lab-r1(config)# logging trap ?
```

The following output displays all the logging levels available on your Cisco router:

<0-7>	Logging severity level	
alerts	Immediate action needed	(severity=1)
critical	Critical conditions	(severity=2)
debugging	Debugging messages	(severity=7)
emergencies	System is unusable	(severity=0)
errors	Error conditions	(severity=3)
informational	Informational messages	(severity=6)
notifications	Normal but significant conditions	(severity=5)
warnings	Warning conditions	(severity=4)
<cr>		

Step 2: Change the Log Level

If you would like to change the trap level, use the global configuration mode command `logging trap {0-7}`. In the following example, the trap level has been set to 7 (debugging):

```
lab-r1(config)# logging trap 7
```

I've found that the default level (6) works well for IDS when used in conjunction with the last rule of the "IPFW-ACL" configured in this book. That rule denies and logs unwanted TCP/IP activity.

Note To use IDS, your WAN ACL must be configured to log all denied traffic. Therefore, you must place the rule `deny ip any any log` at the end of the ACL that is applied to inbound traffic on your WAN interface. In this book, that is the ACL named "IPFW-ACL".

Step 3: Timestamp Your Logs

To have the correct date and time information in your logs, you need to configure the IOS *timestamp* option. The global configuration command `service timestamps` indicates that a timestamp should be applied to system logging messages. The following is the complete command to configure timestamp logging:

```
lab-r1(config)# service timestamps log datetime localtime
```

Step 4: Define Your Logging Host

After you have set up your logging options, you need to configure your router with the TCP/IP address of your logging host. In Listing 5-20, I have used the global configuration command `logging 192.168.1.2`, which specifies the IP address of the host where the logger application is installed.

Defining a Login Banner

The last item I would like to touch upon in this chapter is the *login banner*. A banner is nothing more than a message that displays when you connect or log in to a router. There are several kinds of banners, but the `login banner` presented in Listing 5-21 is the one upon which I would like to focus. The command requires a delimiter character that used to identify the beginning and the end of the message. In Listing 5-21, the delimiter character is the “%” symbol.

A typical login banner informs users whether they are allowed access to your system and what their legal rights are while on that system. It typically informs users that they may be monitored when using the system or that they need to disconnect if they’re not authorized to access the system.

In reality, a legal warning won’t deter someone who is trying to break into your router. A system cracker already knows that what they’re doing is illegal. However, I still put these messages on my client’s routers to alleviate any ambiguity about who can access the router and what their rights are when they do.

Listing 5-21. How to Configure a Login Banner

```
lab-r1> enable
lab-r1# config t
lab-r1(config)#banner login %
Enter TEXT message. End with the character '%'.

```

```
This system is the property of COMPANY XYZ, LLC.
Disconnect IMMEDIATELY if you are not an authorized user!
```

```
THIS IS A PRIVATE NETWORK SYSTEM. It is for the exclusive
use of authorized personnel only. All users have no explicit or
implicit expectation of privacy while using this system.
```

```
All use of this system may be monitored or recorded and
disclosed to authorized personnel or law enforcement officials.
By continuing to use this system, you agree to such monitoring
and recording.
```


Unauthorized or improper use of this system could result in disciplinary action from *COMPANY XYZ, LLC* administration or civil and criminal penalties. By continuing to use this system, you indicate your awareness of these terms and consent to all terms and conditions stated.

If you do not agree to the terms stated, disconnect from this system IMMEDIATELY.

```
% {press ENTER}
```

```
lab-r1(config)#
```

Summary

This chapter primarily dealt with security, performance, and password recovery. The importance of this subject can't be overstated; as a network administrator, you have an obligation to your clients or employer to ensure that their equipment is secure and runs smoothly.

Create a Local User on the Router

The following commands will create a user named "CiscoAdmin" with a password of "P@\$word!". The user has a privilege level of 15.

```
lab-r1> ena
lab-r1# config t
lab-r1(config)# username CiscoAdmin privilege 15 secret P@$word!
lab-r1(config)# line vty 0 4
lab-r1(config-line)# login local
lab-r1(config-line)# exit
lab-r1(config)#
```

Configure Secure Shell (SSH)

Before generating your RSA keys, which are used by SSH, you must set your router's domain name. Your router's fully qualified domain name will be part of your SSH keys. SSHv2 requires a minimum key size of 1024 bits. For example:

```
lab-r1> ena
lab-r1# config t
lab-r1(config)# ip domain-name geekvenue.local
lab-r1(config)# crypto key generate rsa
```

Recover a Lost Password

Recovering a lost password is a multipart process that involves bypassing the startup configuration, rebooting the router without a configuration, copying the startup configuration into main memory, changing the lost password, resaving the startup configuration, and, lastly, modifying the configuration register so that the IOS will again load the startup configuration file when the router is restarted.

To begin, send a Ctrl+Break at startup to enter rommon mode, then use the confreg command confreg 0x2142 to bypass loading the startup configuration.

```
rommon 1> confreg 0x2142
rommon 2> reset
```

After you type **reset**, you will be prompted with a system configuration dialog. This occurs because you have bypassed your startup configuration. Answer “no” to the question “Would you like to enter the initial configuration dialog?”

Use the user EXEC mode command **enable** to go into privileged EXEC mode, then use the command **copy start run** to copy your startup configuration into main memory for editing.

```
Router> ena
Router# copy start run
Destination filename [running-config]? {press ENTER}
```

```
lab-r1# config t
lab-r1(config)# enable secret {YourNewPassword}
lab-r1(config)# config-register 0x2102
lab-r1(config)# exit
lab-r1# copy run start
lab-r1#
```

Back Up the IOS

To back up the IOS, you copy the image file from your router’s flash or webflash to an FTP or a TFTP server. First, display the contents of your router’s flash memory. This will tell you the name and size of your existing IOS image, then use the command **copy flash:{image-name} tftp** to copy your image file to an TFTP server.

```
lab-r1> enable
lab-r1# copy flash:/c850-advsecurityk9-mz.124-4.T7.bin tftp
Address or name of remote host []? 192.168.1.101
Destination filename [c850-advsecurityk9-mz.124-4.T7.bin]? {press ENTER}
```

Upgrade the IOS

To upgrade the IOS, you should first make a backup copy of the existing IOS image to a server. Next, delete the existing IOS from flash memory. Finally, copy your new IOS to flash and restart your router.

```
lab-r1> enable
lab-r1# delete flash:/c850-advsecurityk9-mz.124-4.T7.bin
Delete filename [c850-advsecurityk9-mz.124-4.T7]? Y
```

These are the IOS commands used to copy a new IOS image file from a TFTP server at address 192.168.1.101 to your router's flash memory. Once you have copied the new image, you power cycle the router.

```
lab-r1> enable
lab-r1# copy tftp flash
Address or name of remote host []? 192.168.1.101
Source filename []? c850-advsecurityk9-mz.124-4.T9.bin
Destination filename [c850-advsecurityk9-mz.124-4.T9.bin]? {press ENTER}
```

Back Up Your Configuration

There are many ways to back up the configuration of a Cisco router. In this chapter, I introduced three simple methods (flash, TFTP, and FTP).

The following are the commands to copy your startup-config to flash:

```
lab-r1> enable
lab-r1# copy run flash
Destination filename [running-config]? {Press ENTER}

lab-r1# sh flash
Directory of flash:/
...
 9  -rwx      10545  May 26 2008 20:12:29 -09:00  running-config
```

The following are the commands to back up your running configuration to a TFTP server at address 192.168.1.101:

```
lab-r1> enable
lab-r1# copy run tftp
Address or name of remote host []? 192.168.1.101
Destination filename [lab-r1-config]? {press ENTER}
```

Tune Your ACLs for Performance

Performance tuning involves changing the order of your ACLs so the rules that are most often matched are placed at the top of your ACL. You also place an established rule at the top of your ACL. Established connections are connections that are replying to a host on your network. This rule applies to any established connection, regardless of the protocol. Use the `show ip access-list` command as shown in the following example to display the ACL rules and their number of matches:

```
lab-r1> enable
lab-r1# sh ip access-list IPFW-ACL
```

```
Extended IP access list IPFW_ACL
 10 permit tcp any host 66.238.25.254 eq www (724 matches)
 20 permit tcp any host 66.238.25.254 eq smtp (9629 matches)
 30 permit tcp any host 66.238.25.254 eq 443 (32 matches)
 40 permit icmp any any administratively-prohibited (321 matches)
 ...
```

Remember to place the established rule at the top of your access list to ensure that it will be matched first. This will relieve your router's other ACL rules from processing already established connections. Use the user EXEC mode command `enable` to go into privileged EXEC mode then use the command `sh run` to display your ACL rules.

```
lab-r1> enable
lab-r1# sh run
...
ip access-list extended IPFW-ACL
 permit tcp any host 66.238.25.254 established
 permit tcp any host 66.238.25.254 eq smtp
 permit tcp any host 66.238.25.254 eq www
 permit tcp any host 66.238.25.254 eq 443
```

Protect Your Passwords

When you create passwords on a Cisco router, they are stored in clear text by default. Cisco provides a command that will encrypt all the passwords in your configuration file to ensure their integrity. Use the service `password-encryption` command, as follows:

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# service password-encryption
lab-r1(config)# exit
lab-r1#
```

Disable Show and Tell

Cisco routers and switches share information with each other regarding their hardware, software, and IP addresses using CDP. It's a good idea to *disable* CDP for better security using the following:

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# no cdp run
lab-r1(config)# exit
lab-r1#
```

Safeguard Your E-mail Server

The Cisco IOS firewall can be applied to inbound traffic on your WAN interface for inspecting e-mail protocols. Because the IOS firewall is an application layer firewall it will look for, and terminate, connections that attempt to send invalid commands. If you inspect SMTP you can define the maximum size for an e-mail message using the `max-data` command. The default is 20 MB (20000000 bytes). Following is an example showing common e-mail protocols that can be guarded by the EIE:

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# ip inspect name EMAIL-FW smtp max-data 10000000

lab-r1(config)# ip inspect name EMAIL-FW imap
lab-r1(config)# ip inspect name EMAIL-FW imaps

lab-r1(config)# ip inspect name EMAIL-FW pop3
lab-r1(config)# ip inspect name EMAIL-FW pop3s

lab-r1(config)# int fa4
lab-r1(config-if)# ip inspect EMAIL-FW in
lab-r1(config-if)# exit
lab-r1(config)# exit
lab-r1#
```

Set Up an Intrusion Detection System

Syslog is a protocol that uses TCP/IP to log messages to a remote server. In addition, it can be used to create an IDS. The Kiwi Syslog Daemon is an application for Windows that can be used for an IDS. Kiwi will generate and e-mail alerts once a pre-established syslog message threshold has been met.

Define a Login Banner

Login banners inform users of their legal rights. A typical login banner informs users whether they are allowed access to your system and what their legal rights are while on your system. Here's an example showing how to define such a banner:

```
lab-r1> enable
lab-r1# config t
lab-r1(config)#banner login %
Enter TEXT message. End with the character '%'.

```

```
This system is the property of COMPANY XYZ, LLC.
Disconnect IMMEDIATELY if you are not an authorized user!

```

```
% {press ENTER}

```

```
lab-r1(config)# exit
lab-r1#

```



Understanding Binary and Subnetting

In general, small businesses are comprised of a single network that uses a single network address and mask (e.g., the network address 192.168.1.0 255.255.255.0 used throughout the examples in this book). However, large companies or companies that have multiple departments often use multiple networks or break up their network into smaller *network segments* called *subnetworks* (subnets). Either networking method provides each department its own LAN within the same company network. Cisco refers to this network design as an “internetwork,” hence the reason the IOS is called an *Internetworking Operating System*. Using routers and routing protocols, a company can control the flow of data between departments within the company internetwork (simply called a “network” by most IT professionals).

Businesses do this for several reasons. One reason is to isolate different departments within the company for higher security. An example of this would be isolating a Sales department LAN from a Finance department LAN. Another reason would be to isolate network traffic to avoid network congestion. An example of this would be an Engineering department that works with large amounts of data, perhaps large CAD drawings, image renderings, or simulation data. In this scenario, a company would want to keep the engineering data separate from the other departments to prevent it from impacting the other department’s network performance.

There are two ways to achieve these goals: Create multiple networks, known as *classful* networks, or use subnetting to divide a single network into smaller subnets. Subnetting requires a little math, but is fun to learn and it’s a great conversation starter at your next party!

Note This chapter will discuss the principals and concepts of TCP/IP addresses, which are necessary for understanding routing with routing protocols (discussed in Chapter 7) or passing the Cisco Certified Network Associate (CCNA) exam.

If you have ever wondered why a TCP/IP address, such as the class C address **192.168.1.0 255.255.255.0**, can also be written as **192.168.1.0/24**, then read on and I will explain it to you! In this chapter I will try to dispel the mystery of binary (otherwise known as Base 2) and explain why it is important in subnetting. Learning this is not very complicated and it's critical to understanding networking with routers in an internetwork.

Decimal—Base 10

The first thing you must know is that the common number system used worldwide is the decimal system otherwise known as *Base 10*. What makes the decimal system a Base 10 system is that it is based on *grouping numbers by tens*. Conventional wisdom is the system evolved because we have ten fingers and ten toes, which we have used for counting over the years. I use mine all the time! We name the ten digits zero, one, two, three, four, five, six, seven, eight, and nine.

The decimal system has a 1's place, a 10's place, a 100's place, a 1000's place, and so on. We say the number places are grouped by tens because *multiplying each number place by "10" gives you the next number place*. Therefore, $1 \times 10 = 10$ (the 10's place), $10 \times 10 = 100$ (the 100's place), $100 \times 10 = 1000$ (the 1000's place), etc.

Let's look at the decimal number **103** by number place, reading the number from right to left.

We have a **3** in the **1's place**

We have a **0** in the **10's place**

We have a **1** in the **100's place**

Thus, $100 + 0 + 3 = 103$

By now you probably feel like you have attended grade school for the second time in your life. I'm sorry about that, but it is very important that you understand the concept of what a number system is, and what it is based on, before we look at binary.

It may seem odd, but any known number can be the basis of a number system. Another number system used in the computer world is the hexadecimal system (hex). *Hex* is a *Base 16* number system based on grouping numbers by 16. However, the number system you need to know before you can understand TCP/IP subnetting is the *Base 2* number system known as binary.

Binary—Base 2

Binary is a Base 2 number system and, thus, it groups numbers by twos instead of tens like the decimal system. We name the two digits zero and one. The binary system has a 1's place, a 2's place, a 4's place, an 8's place, a 16's place, and so on. We say the number places are grouped by twos because *multiplying each number place by two gives you the next number place*. Therefore, $1 \times 2 = 2$ (the 2's place), $2 \times 2 = 4$ (the 4's place), $4 \times 2 = 8$ (the 8's place), $8 \times 2 = 16$ (the 16's place), etc.

Let's look at the decimal number 103 in binary format. In binary, we have 01100111. Again, reading from right to left

We have a 1 in the 1's place

We have a 1 in the 2's place

We have a 1 in the 4's place

We have a 0 in the 8's place

We have a 0 in the 16's place

We have a 1 in the 32's place

We have a 1 in the 64's place

We have a 0 in the 128's place

Thus, $0 + 64 + 32 + 0 + 0 + 4 + 2 + 1 = 103$

BINARY CONVERSION QUIZ

Okay, now test your skills. Here is a list of binary numbers. Try converting them to decimal and check your answers at the end of this chapter.

10000000

11000000

11100000

01000000

10000011

10010001

11111111

If you could convert these numbers to decimal, then congratulations! You're ready to move on to the next section.

Subnet Masks

Now that you understand what binary is, let's have a look at our two subnet masks from the beginning of this chapter:

192.168.1.0 255.255.255.0

192.168.1.0/24

The concept of a subnet mask is simple. You have a network and you have hosts on the network (anything with an IP address is a host). **The subnet mask determines what portion of the TCP/IP address represents your network and what portion can be used for your hosts.** Because I am a simple person, I think of it like this: the network number represents the street I live on and the host portion is used for the numbers of all the houses on my street.

A subnet mask of **255.255.255.0** means that the first *three* octets of the address will be used for the network address. Thus, your network number is **192.168.1**, meaning you can have **254** computers on this network. You know this because the fourth octet in the subnet mask is **0** (255.255.255.0), indicating that it is not being used by the network portion of the address.

Note The reason your maximum number of hosts is 254 and not 255 is that 255.255.255.255 is a special broadcast address.

Each of the number sections is called an *octet* because when displayed in binary there are eight possible bits in each section. Eight bits is an octet. **11111111** in binary is 255 in decimal. So your decimal subnet mask 255.255.255.0 displayed in binary is

11111111.11111111.11111111.00000000

If you count all the ones, you will find that there are **24** of them. Now, look at the subnet mask examples again.

192.168.1.0 255.255.255.0

192.168.1.0/24

Do you see why both subnet masks are the same? The number **24** is the number of *bits* used in the network portion of the address and is shorthand for writing the address/subnet mask combination. It becomes important to understand this when you start dividing your network into multiple subnetworks.

Dividing Your Network

Before reading this section, you should have a good understanding of what a subnet mask is, and how binary bits represent the subnet mask.

There are two methods for dividing your network into multiple networks. One is to simply create more than one network. This method gives you two or more networks using more than one network address and mask. The other method is to divide your network into smaller *subnetworks*. This method allows you to divide your network using only a single network and subnet mask combination.

Simply put, subnetting is *dividing your network into multiple subnetworks*. To go back to my silly example about houses and streets, subnetting gives you multiple streets in your neighborhood.

Method 1: Keeping the Same Subnet Mask

Divide a network into two or more networks by changing the network address portion of the address, such as *192.168.1* and *192.168.2*, and keeping the same subnet mask. The network mask (255.255.255.0) indicates that the first three octets are used to define the network. These are different network addresses because of the change in the third octet (1 and 2, respectively), but you could just as easily change the address to something else, like the network address 200.100.10. What's important to understand is that these addresses are not subnets, they are completely separate networks. Here's an example of two network addresses and their masks:

192.168.1.0 255.255.255.0

192.168.2.0 255.255.255.0

Doing this would give you **two separate networks** with **254** hosts per network. This is a very common method of dealing with multiple networks. However, back in the good old days, you had to pay for every IP address you used and, if you had 25 computers on your network, you probably would not want to pay for 254 addresses! The answer to the problem is...subnetting.

Method 2: Subnetting a Network

Subnetting is when you use bits from the host portion of your address as part of your network number. This allows you to subdivide your network at the cost of host addresses, which is great if you're paying for every host IP address. It will save you money because you pay for fewer TCP/IP addresses. Confused? Here is where understanding binary is important.

Let's look at a new subnet mask.

255.255.255.224

As you can see from the number in the *fourth octet*, some of the host portion of this subnet mask is being used for part of the network address. A zero would indicate that it is being used exclusively for hosts; however, a number indicates you are *using some of the binary bits in the fourth octet for our network numbers*. This gives you fewer hosts than your old mask (which had 254), but more networks. Hence, the name, subnetting.

Determining How the Bits Are Used

To tell how many networks and how many hosts per network this new subnet mask will give you, use some of your newly acquired binary skills. The first task is to find out *how many bits in the fourth octet are being used for network addresses*. The decimal number is 224. What is the decimal number 224, as represented in binary?

In binary the decimal number 224 is 11100000. Reading from right to left

We have a 0 in the 1's place

We have a 0 in the 2's place

We have a 0 in the 4's place

We have a 0 in the 8's place

We have a 0 in the 16's place

We have a 1 in the 32's place

We have a 1 in the 64's place

We have a 1 in the 128's place

Thus, $128+64+32+0+0+0+0+0=224$. And our complete subnet mask in binary is

1111111.11111111.11111111.11100000

The three bits from the fourth octet are used for network addresses.

Determining the Number of Subnets Available

Determining how many subnetworks you're going to have requires a little math. The standard formula is 2^n , where n is the number of bits being used from the host portion of our subnet mask (3 in this case). For example, the standard formula for *three bits* is: $2^3=8$.

Note Math geeks use the term “two to the third power” to describe this formula.

In simpler terms:

$$2 \times 2 \times 2 = 8$$

So our network is divided into *eight subnetworks*.

Determining the Network Numbers and Number of Hosts

Next, we want to know what the network numbers are and how many hosts we can have on each of the eight subnetworks. To determine the range of the subnets, look at the bits in your *fourth octet* again. The bit that gives you the answer is the “1” *closest to the first zero*; in this example, it is the third bit from the left.

11100000

The third bit is in the 32’s place (remember to think of it in binary) thus the block size for each of your subnets is 32. To determine what the subnet numbers are, add the value 32 to itself eight times, starting at zero, to get the eight subnet numbers.

A block of addresses includes a network address, a host range, and a broadcast address. The available host addresses fall between the network address and the broadcast address on each subnet.

Tip A quicker way to find your network block size is to subtract your subnet mask from 256.
(256-224=32)

Here are your subnets:

0

32

64

96

128

160

192

224

A better way to display our subnet addresses is

192.168.1.0

192.168.1.32

192.168.1.64

192.168.1.96

192.168.1.128

192.168.1.160

192.168.1.192

192.168.1.224

The host addresses will fall between the network numbers, so you will have 30 hosts per network. You're probably wondering why it's not 31. The answer is that the last address of each subnet is used as the broadcast address for that subnet. Let's look at the second subnet 192.168.1.32.

Subnet: 192.168.1.32 / 255.255.255.224

Address range: 192.168.1.33 through 192.168.1.62 (30 hosts)

Subnet broadcast address: 192.168.1.63

Note A broadcast address cannot be assigned to a host. Broadcast addresses are used to send messages to all hosts for a given subnet.

If you were paying for your TCP/IP addresses, you would only pay for one network and host combination, thus paying for 30 hosts and *not* 254. It could mean some real savings. It also frees up the remaining addresses for other organizations to use.

SUBNETTING QUIZ

Test your subnetting skills—write the address range and broadcast address for the following subnet. You will find the answer near the end of this chapter.

Subnet: 192.168.1.**128** 255.255.255.**224**

Address range?

Subnet broadcast address?

More Examples

Let's look at another subnet mask:

255.255.255.**240**

To determine how many bits are used from the host portion of the subnet mask, you have to know how the decimal number 240 is represented in binary. The answer to that question is

11110000

Four bits are taken from the host portion of our mask. We do the same math as before.

$2^4=16$

In simpler terms

$2 \times 2 \times 2 \times 2 = 16$

We will have **16 subnetworks**. And what then will the network numbers be? Look at the *fourth bit*; it's in the 16's place:

11110000

Note A quicker way to find the network block size is to subtract the value of your subnet mask's right-most, non-zero octet from 256. In this case: $256-240=16$.

Begin adding 16 to itself, starting at zero, 16 times to get all 16 subnetworks.

0

16

32

48

64

80

96

112

128

144

160

176

192

208

224

240

A better way to display our subnets is

192.168.1.0

192.168.1.16

192.168.1.32

192.168.1.48

192.168.1.64

192.168.1.80
192.168.1.96
192.168.1.112
192.168.1.128
192.168.1.144
192.168.1.160
192.168.1.176
192.168.1.192
192.168.1.208
192.168.1.224
192.168.1.240

The host addresses fall between the network numbers. So you will have 14 host addresses on each of your 16 subnetworks. If you had a small company with ten hosts and needed to have a static IP address for all of your hosts, you would be assigned a network subnet mask and a valid IP address range from your ISP.

The following is an example of what that might look like:

Network: 205.112.10.16 255.255.255.240

Address range: 205.112.10.17 through 205.112.10.30 (14 hosts)

Subnet broadcast address: 205.112.10.31

The network address is 205.112.10.16 which means your first host for the subnet starts at the next address, which is 205.112.10.17, and the last host is at address 205.112.10.30. The next and last address for this subnet is the broadcast address at 205.112.10.31.

Since there is no network address and subnet mask that exactly fits your ten hosts, you have to use a combination that comes close. A subnet address that supports 14 hosts is as close as you can come to the ten host addresses you require without wasting a lot of addresses. In this scenario, you would have four addresses available for future growth.

NETWORK CLASSES

In TCP/IP networking, there are three main network classes. They are referred to as “Class A,” “Class B,” and “Class C.” There are also classes D and E, but they’re special classes and not used for creating valid network addresses. Each class is governed by a set of rules that defines the network portion and the host portion for a given network address. In particular, the rules determine if a network address is valid for a particular class, and how many hosts will be supported by the address. The rules themselves are governed by binary but I won’t go into that now. Here’s a brief overview of the rules that make up a class:

“Class A” networks use the first octet for the network address, and the last three octets for hosts. Class A networks allow you to have 16,777,214 hosts per network. The following is an example of a “Class A” network and mask:

1.0.0.0 255.0.0.0

“Class B” networks use the first two octets for the network address, and the last two octets for the hosts. Class B networks allow you to have 65,534 hosts per network. The following is an example of a “Class B” network address and mask:

128.0.0.0 255.255.0.0

“Class C” networks use the first three octets for the network address, and the last octet for hosts. Class C networks allow you to have 254 hosts per network. In this book, I have stuck to using Class C networks. The following is an example of a “Class C” network address and mask:

192.0.0.0 255.255.255.0

Not all addresses are routable on the Internet. Some are reserved for private network use only. The *Internet Assigned Numbers Authority* (IANA) has reserved the following network addresses for use on private networks:

10.0.0.0 through 10.255.255.255
172.16.0.0 through 172.31.255.255
192.168.0.0 through 192.168.255.255

Summary

This chapter covered the theory of TCP/IP addressing and subnetting. You learned why a company might want to create multiple networks or breakup their network address into multiple subnetwork segments, and the different addressing techniques that are used to divide a network.

Businesses create multiple networks for security and to isolate network traffic by a group of users or departments within a company. You can create multiple networks by

using multiple disparate network addresses, or you can subnet your network using a single network address and subnet mask.

You learned what a number system is and that the most commonly used number system worldwide is the decimal system. You also learned about binary and how it relates to a subnet mask.

Decimal—Base 10

The decimal system is a Base 10 number system because it groups numbers by tens. In the decimal system, there is a 1's place, 10's place, a 100's place, a 1000's place, and so on. Multiplying each number place by "10" gives the next number place.

Binary—Base 2

Binary is a Base 2 number system because it groups numbers by twos. In binary there is a 1's place, a 2's place, a 4's place, an 8's place, and so on. Multiplying each number place by "2" gives the next number place.

Subnet Mask

A subnet mask determines which portion of a TCP/IP address is the network and which portion is used for the hosts on that network. In the address 192.168.1.0 255.255.255.0, the entire last octet can be used for hosts, which allows for 254.

Dividing Your Network

A network can be divided two ways: You can create multiple networks using more than one network address and mask, or you can subnet your network using only one network address and mask. Subnetting allows you to create multiple networks at the expense of hosts. When you subnet a network, you will have fewer hosts per segment than an unsubnetted network with the same address.

Determining the Number of Subnets

Use the formula 2^n , where n is the number of bits being used from the host portion of our subnet mask. If three bits were being used, you would solve the problem like this:

$$2 \times 2 \times 2 = 8$$

Determining the Network Number and Number of Hosts

The easiest way to determine what your network numbers are is to find the network block size. Subtract the right-most octet (that's not a zero) of your subnet mask, from the number 256. For the mask 255.255.255.**224**, use this math: $256 - 224 = 32$ (the block size is 32).

Next, starting at 0, add the block size to itself until you reach the number of subnets.

For eight subnets, the networks are **0, 32, 64, 96, 128, 196, 192, and 224**.

The last address of each subnet is a broadcast address. All your host addresses fall between the network address and the broadcast address. For your first subnet, the network address is 0, so your hosts are **1-62** and 63 (being the last address is the broadcast address for the subnet).

Quiz Answers

Here are the answers to the exercises found in this chapter.

Binary Quiz Answers

10000000 = 128

11000000 = 192

11100000 = 224

01000000 = 64

10000011 = 131

10010001 = 145

11111111 = 255

Subnetting Quiz Answers

Subnet: 192.168.1.**128** 255.255.255.224

Address range: 192.168.1.**129** through 192.168.1.**158** (30 hosts)

Subnet broadcast address: 192.168.1.**159**



Routing—What Routers Do Best

Nearly everyone has a router these days. We use them at home, work, and at our favorite caffeine hangouts. But just what the heck is routing! In this chapter, I'm going to introduce you to some routing protocols and show you how to configure the *Routing Information Protocol* (RIP) using two routers to set up a true DMZ on your private network.

Routing Defined

The purpose of routing, as defined by Cisco, is *path determination and packet switching*. Basically, it's about routers sharing information about routes to remote networks, and using that information to choose the best path by which to get there. In large installations, routers are often interconnected to one another to provide route redundancy. This allows connectivity to continue in the event of a downed connection or a router failure.

Routing requires that a *routing protocol* be configured on your routers. Many different routing protocols exist, but the most commonly used on small networks are RIP, *Enhanced Interior Gateway Routing Protocols* (EIGRP), and *Open Shortest Path First* (OSPF). Cisco's 800 series routers only support RIP, so that's what I'll be covering in this chapter.

Routing protocols use different methods of advertising routes, and choosing routes from one network to another. Some protocols like RIP use a *hop count*, while others use a *cost* method. A hop count measures how many networks a packet must pass through (hop) to travel from point A to point B; it doesn't look at the actual bandwidth (speed) of the paths. On the other hand, routing based on cost measures bandwidth and uses that information to select the fastest route from point A to point B. Simply put, routing is about routers in an internetwork, advertising their paths to a destination network and choosing the best path to get there.

Routing vs. Routed Protocols

You were just introduced to a few *routing* protocols, so what are *routed* protocols? Well, they're protocols that can be routed, of course! These days there's really only one routed protocol in use and it's TCP/IP. In the past, networks used other protocols, such as Novell's IPX/SPX and Apple's AppleTalk protocols. If you want routing on your private network, you will need to use a routing protocol that supports those other network protocols if they are used on your network. RIP only supports TCP/IP.

Routing Information Protocol (RIP)

I like to think of RIP as the poor man's routing protocol. Many vendors, including Cisco, support it and private networks that have multiple routers commonly use it because it's easy to set up, extremely dependable, and works well for the needs of a small business. RIP only routes IP packets but that's just fine for most of today's small business networks.

RIP Basics

When RIP is configured on a router, you need to configure which of its network addresses you would like it to advertise to its neighbors (the other routers on your private network). All the routers using RIP will then share with each other information about their connected networks and their neighbor networks. This method of sharing information is often referred to as *routing by rumor*. It allows your routers to build a network map known as a *routing table*, which determines where to send data packets to get them from one network to another.

It's called routing by rumor because RIP-configured routers don't learn about all the routes firsthand. Let's assume you have three routers called "R1," "R2," and "R3." R1 is connected to R2 and R2 is connected to R3. Using RIP, router R1 tells R2 about its configured networks, R2 then tells R3 about its configured networks, as well as R1's networks. Therefore, R3 learns about the location of R1's networks from R2. It's called a rumor because the information did not come directly from the source.

Configuring RIP on a Router

Cisco 800 Series routers support RIPv1 and RIPv2. The main difference between the two versions is that RIPv1 does not share subnet mask information with its neighbors. This means that RIPv1 does not support subnetted networks (if you don't know about subnets, refer to Chapter 6). Therefore, it will only advertise classful network addresses and not subnetworks. Unlike version 1, RIPv2 does propagate the subnet mask information in its route advertisements.

Note Subnetted networks are referred to as “classless” networks.

Let’s have a look at a network that uses two routers named “*lab-r1*” and “*lab-r2*”. In Figure 7-1, the private network is comprised of two separate networks. They are 192.168.1.0 and 192.168.2.0. Network 192.168.1.0 is our private *internal* network, and 192.168.2.0 is our private dedicated *DMZ* network (I’ll discuss the DMZ later in this chapter). It’s important to remember that these are network addresses. The IP addresses assigned to individual router interfaces and hosts will replace the zero in the network number with a unique number from 1 through 254 that represent the host or interface address (e.g., 192.168.1.1 and 192.168.2.1).

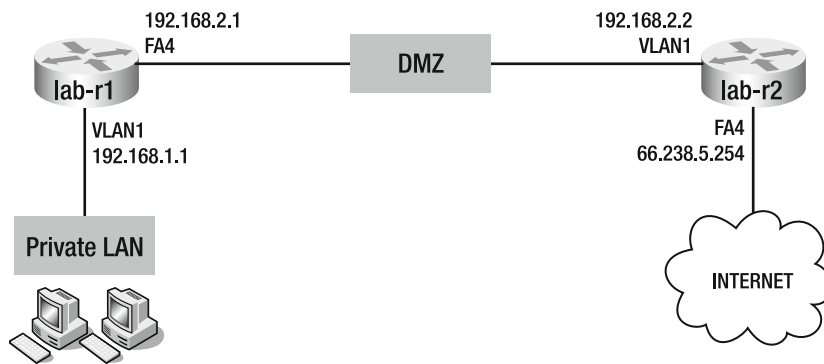


Figure 7-1. *RIP network diagram*

RIP provides a way for the two routers to inform each other about their directly connected networks and any remote networks that they know about, so that routing between all the networks can occur. Without a routing protocol like RIP, the routers would only know about their directly connected networks and none of the remote networks. Therefore, router “*lab-r1*” knows about both directly connected private networks 192.168.1.0 and 192.168.2.0, but router “*lab-r2*” only knows about its directly connected private network 192.168.2.0. Thus, router “*lab-r2*” would not know what interface to use when sending packets that are destined for network 192.168.1.0. Remember, routing is all about path determination and packet switching.

The way to solve this problem is to configure RIP on each of the routers. Listing 7-1 shows how to configure RIP on router “*lab-r1*” to advertise both of its private networks 192.168.1.0 and 192.168.2.0.

Listing 7-1. *How to Configure RIP on a Router*

```
lab-r1> enable
lab-r1# config t
lab-r1(config)# router rip
lab-r1(config-router)# version 2
lab-r1(config-router)# network 192.168.1.0
lab-r1(config-router)# network 192.168.2.0
lab-r1(config-router)# exit
lab-r1(config)#
```

Step 1: Enable RIP

Begin by enabling RIP on your internal router “lab-r1”. Use the user EXEC mode command `enable`, to go into privileged EXEC mode. Then use the command `config t` to start global configuration mode. Use the command `router rip` to enable the protocol. Notice that your prompt changes to `(config-router)#`, which indicates that you’re in routing configuration mode. RIPv1 is enabled by default; to enable the more current version RIPv2, use the router configuration command `version 2`.

Step 2: Advertise Your Networks

Once you’re in routing configuration mode, use the routing configuration command `network {network-address}` to allow RIP to advertise a network to its neighbor routers that are also running RIP. In Listing 7-1, “lab-r1” will advertise both networks 192.168.1.0 and 192.168.2.0.

Configuring RIP on a Neighbor Router

Now that RIP is enabled on one router, it will need a neighbor router with which it can share routing information before routing can occur. The router “lab-r2” in Listing 7-1 is our neighbor router, but it’s also connected to an ISP making it the gateway router to the Internet.

This is important because you configure RIP differently on a gateway router than you do an interior router. This is because you don’t want RIP to advertise your private networks on the WAN interface to the Internet. It’s generally considered bad form from a security standpoint, and it tends to tick off your ISP.

Listing 7-2 displays all the commands to configure RIP on the gateway router “lab-r2”. The gateway router only advertises the private DMZ network 192.168.2.0 and is configured not to send RIP updates on the WAN interface (fa4).

Listing 7-2. *How to Configure RIP on a Gateway Router*

```
lab-r2> enable
lab-r2# config t
lab-r2(config)# router rip
lab-r2(config-router)# version 2
lab-r2(config-router)# network 192.168.2.0
lab-r2(config-router)# passive-interface fa4
lab-r2(config-router)# exit
lab-r2(config)#
```

Step 1: Enable RIP

Use the user EXEC mode command `enable`, to enter privileged EXEC mode. Next, use the command `config t` to start global configuration mode. Finally, use the command `router rip` to enable RIP routing. This will place you in routing configuration mode.

Step 2: Advertise Your Networks

Because “lab-r2” is an Internet gateway router, you only need to advertise your private network, 192.168.2.0, using the routing configuration command `network 192.168.2.0`.

Step 3: Configure a Passive Interface

Use the routing configuration command `passive-interface {interface-name}` to prevent a particular interface from sending out RIP updates. In this case, you don’t want “lab-r2” to send updates via the WAN interface `fa4` because it’s connected to the Internet. Issue the router configuration command `passive-interface fa4` to discontinue all RIP updates on the interface.

Verifying RIP Routing

After you have configured your routers with RIP, you can verify that they are receiving routing updates using the IOS. Two commonly used commands are `show ip protocols` and `show ip route`.

Use Show IP Protocols

Use the user EXEC mode command `enable` to enter privileged EXEC mode and issue the command `show ip protocols` to display information about your routing protocols and advertised networks.

```
lab-r1> enable
lab-r1# show ip protocols
```

In the following output, router “lab-r1” is configured to use RIP version 2 and advertises networks 192.168.1.0 and 192.168.2.0 to its neighbors:

Routing Protocol is "rip"

```
Sending updates every 30 seconds, next due in 14 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Redistributing: rip
Default version control: send version 2, receive version 2
  Interface          Send  Recv  Triggered RIP  Key-chain
  VLAN1              1     1  2
  FastEthernet4      1     1  2
Automatic network summarization is in effect
Maximum path: 4
Routing for networks:
  192.168.1.0
  192.168.2.0
Routing information sources:
  Gateway          Distance    Last Update
  192.168.2.2      120        00:00:16
Distance: <default is 120>
```

Use Show IP Route

Issue the user EXEC mode command `enable` to start privileged EXEC mode then issue the command `show ip route` to display information about all the networks your router knows about, and which interface will be used to reach those networks.

```
lab-r2> enable
lab-r2# show ip route
```

The resulting output provides us with a wealth of information.

```
.  
.  
Gateway of last resort is 66.238.5.1 to network 0.0.0.0
```

```
    66.238.5.0/24 is subnetted, 1 subnets  
C    66.238.5.0 is directly connected, FastEthernet4  
R    192.168.1.0 [120/1] via 192.168.2.1, 00:00:19, VLAN1  
C    192.168.2.0/24 is directly connected, VLAN1  
S*   0.0.0.0 [1/0] via 66.238.5.1
```

Take a moment to dissect this information. First, you have two physically connected networks configured on the router. The C stands for *connected*. They are networks 66.238.5.0 on interface FastEthernet4 (fa4) and 192.168.2.0 on interface VLAN1.

Next, you have received a RIP *routing-update* to network 192.168.1.0, as indicated by the R, and that it was advertised to you through your VLAN1 interface from address 192.168.2.1. This means that RIP on router “lab-r2” will use interface vlan1 to forward packets to network address 192.168.1.0, which is exactly what you want.

Finally, you have a *static route* 0.0.0.0, represented by the S*, which informs you that you have configured a static route to the default gateway of your ISP at address 66.238.5.1.

At this point you should be able to use the ping utility from “lab-r1” to ping all the IP addresses on “lab-r2” and vice versa. Doing so will help to ensure that RIP routing is functional. If it is, then congratulations, you can move on to the next section!

Setting Up a True DMZ

In earlier chapters, I explained how to set up a simple DMZ using a host on your private network by redirecting specific traffic through your router’s firewall to that host. Although this is a common practice, especially on small networks, it’s not the most secure method of setting up a DMZ. A *true* DMZ sits on its own network, completely isolated from your other hosts and private servers. This type of DMZ is often referred to as a *perimeter LAN*. If a server (for instance, an e-mail server) is placed on a true DMZ and an attempt is made to hack into it, the hacker will be stuck in the DMZ’s perimeter network and will need to launch a second attack to try to gain access to the hosts on your internal private network. A true DMZ is just another layer of defense. A host located in the perimeter LAN is often referred to as a *bastion host*.

This type of DMZ is common on larger network installations, but it is rarely used on smaller ones, mainly because people don't know they exist, think they're expensive, or are just too complicated to set up. In reality, they're not expensive and they're easy to set up. You just need a little knowledge.

The Bastion Host

A bastion host is a fancy term for a server that sits on a DMZ network. As a rule, this host should be configured for a single purpose, such as receiving e-mail for your network or serving up web pages. All other services should be disabled on the bastion host. The term used for this is *hardening* your server. In Figure 7-2, there's an e-mail server sitting on the DMZ network at address 192.168.2.3. This server is the bastion host.

What you may not have noticed is that there is a *second* e-mail server that is located on the internal private network, at address 192.168.1.2. The idea is that the exposed server (bastion host) will receive e-mail for your domain. The e-mail will then be relayed to the other e-mail server on the private network. The private e-mail server is the actual mail store for all the users. They will access their e-mail from this server.

Note Often the bastion host is a Linux or FreeBSD server running sendmail or some other SMTP server application, and the private e-mail server is a Microsoft Exchange Server.

There are several benefits to this scenario.

- The bastion host is a *security-hardened* server and it is only configured to receive e-mail messages and then relay them to our private e-mail server. This type of server is often called an *e-mail relay server*. To help make this server less vulnerable to system crackers, you need to disable all nonessential protocols and services.
- In the event that a system cracker successfully compromises the server, there's no accessible data on that server because your bastion host *only relays user e-mail* and doesn't store it there.
- There are *two firewalls* on the network (one on each router) that provides double coverage. The firewall on "lab-r2", the outside router that's exposed to the Internet, is configured with the standard firewall ACLs introduced in Chapter 2. However, your interior router's firewall ACL denies all traffic except for relayed e-mail from your bastion host.

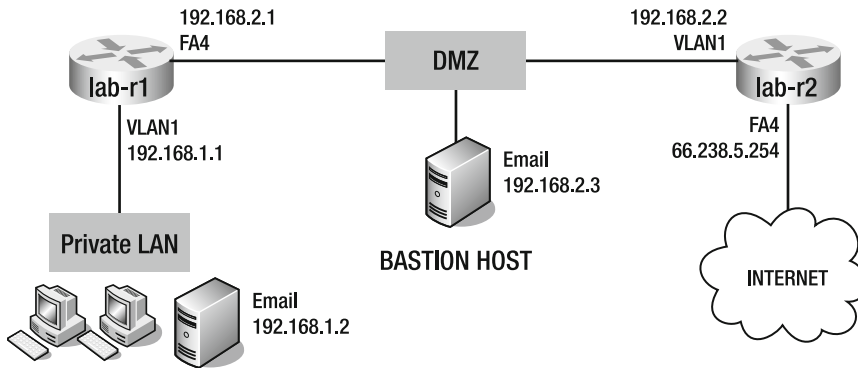


Figure 7-2. True DMZ network and bastion host

Configuring Your Gateway Router

You configure the outside gateway router “lab-r2” in almost the same way described in Chapter 2, with NAT and port forwarding. You just need to remember to forward e-mail to your bastion host and not your private e-mail server. Also, the ACL used with NAT named “NAT-ACL” needs to permit traffic from your *inside private network* address 192.168.1.0 and not your DMZ. This will ensure that your inside private hosts can be processed by NAT and have access to the Internet through your gateway router.

Step 1: Configure NAT

Listing 7-3 shows all the commands to configure NAT on your gateway router.

Listing 7-3. How to Configure NAT

```
lab-r2> enable
lab-r2# config t
lab-r2(config)# ip access-list extended NAT-ACL
lab-r2(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 any
lab-r2(config-ext-nacl)# exit

lab-r2(config)# ip nat inside source list NAT-ACL interface fa4 overload

lab-r2(config)# int vlan1
lab-r2(config-if)# ip nat inside

lab-r2(config-if)# int fa4
lab-r2(config-if)# ip nat outside
```

As I stated earlier, the “NAT-ACL” on your gateway router needs to be configured to translate the addresses from your inside private network and not the DMZ network. The inside private network is configured on router “lab-r1” at address 192.168.1.0.

Step 2: Configure Port Forwarding

Next, configure port forwarding on your gateway router. Use NAT to forward SMTP e-mail to your bastion host. In Listing 7-4, SMTP (port 25) is being forwarded to our e-mail server located in the DMZ, at address 192.168.2.3.

Listing 7-4. *How to Configure Port Forwarding with NAT*

```
lab-r2>enable
lab-r2# config t
lab-r2(config)# ip nat inside source static tcp 192.168.2.3 25 interface fa4 25
lab-r2(config)# exit
```

That’s all there is to setting up your Internet gateway router. Next, you’ll need to set up your internal router. The private network on the internal router is where you connect your servers and workstations.

Configuring Your Interior Router

An internal router is configured differently than a gateway router. You do not configure NAT on this router and you create a very limited firewall ACL with no port forwarding.

Step 1: Configure a Firewall ACL to Permit a Bastion Host

Listing 7-5 illustrates you how to configure an ACL named “IPFW-ACL” that will not only allow outbound ping and traceroute (like the other router’s firewall ACL), but will also allow inbound SMTP from the bastion host. Because the internal router is not configured with NAT, do not port forward SMTP to the private e-mail server. Instead, using an ACL rule, allow it to be routed directly to the private e-mail server by IP address.

Listing 7-5. *How to Configure a Firewall ACL to Permit a Bastion Host*

```
lab-r1(config)# ip access-list extended IPFW-ACL
lab-r1(config-ext-nacl)# permit tcp host 192.168.2.3 host 192.168.1.2 eq smtp
lab-r1(config-ext-nacl)# permit icmp any any administratively-prohibited
lab-r1(config-ext-nacl)# permit icmp any any echo-reply
lab-r1(config-ext-nacl)# permit icmp any any packet-too-big
lab-r1(config-ext-nacl)# permit icmp any any time-exceeded
lab-r1(config-ext-nacl)# permit icmp any any traceroute
lab-r1(config-ext-nacl)# permit gre any any
lab-r1(config-ext-nacl)# deny ip any any log
lab-r1(config-ext-nacl)# exit
```

```
lab-r1(config)# int fa4
lab-r1(config-if)# ip access-group IPFW-ACL in
lab-r1(config-if)# exit
```

What's important to understand is that you don't want to allow all data from your bastion host to enter your internal private network. Instead, you want your DMZ perimeter to be a *choke point* that strictly limits inward traffic. You need to create an ACL rule that only permits SMTP e-mail to be relayed to your internal mail server. In Listing 7-5 that rule is

```
lab-r1(config-ext-nacl)# permit tcp host 192.168.2.3 host 192.168.1.2 eq smtp
```

Step 2: Configure the Internal IOS Firewall

In addition to the ACL configured on the internal router, you will need to configure an IOS firewall (see Chapter 2 to learn more about the Cisco CBAC Firewall). Be sure to also configure the e-mail inspection engine. If someone compromises your bastion host, they may attempt to send invalid SMTP commands to your internal e-mail server in an attempt to compromise it, too. The commands shown in Listing 7-6 configure a basic IOS firewall and e-mail inspection engine.

Listing 7-6. How to Configure the Internal IOS Firewall

```
lab-r1(config)# ip inspect name IPFW tcp
lab-r1(config)# ip inspect name IPFW udp
lab-r1(config)# ip inspect name EMAIL-FW smtp

lab-r1(config)# int fa4
lab-r1(config-if)# ip inspect IPFW out
lab-r1(config-if)# ip inspect EMAIL-FW in
lab-r1(config-if)# exit
lab-r1(config)# exit
```

A Note on VPNs and DMZs

One last subject I would like to discuss is IPSec VPN tunnels. What you need to know is that if you're using a true DMZ and have an IPSec VPN between two sites, you'll configure the VPN on your gateway router, which is "lab-r2" in this example. However, you configure all your VPN-related ACLs using the network address of your internal private network (not your DMZ). In the example network for this chapter, that address is 192.168.1.0.

The reason for this is that your VPN tunnel is for providing your private network hosts access to your remote office and not your DMZ hosts. You would not want a situation where someone compromises your bastion host and then has unrestricted access to the remote hosts at the other end of your VPN tunnel! You only want your internal private hosts to have access to the VPN tunnel. The two ACLs that need be considered are the VPN friendly "NAT-ACL" and the "VPN-ACL" that have been used throughout this book.

Listing 7-7 presents the commands to configure the NAT and VPN ACLs on the gateway router.

Listing 7-7. *How to Configure VPN-friendly ACLs on the Gateway Router*

! VPN-friendly NAT-ACL

```
lab-r2(config)# ip access-list extended NAT-ACL
```

```
lab-r2(config-ext-nacl)#deny ip 192.168.1.0 0.0.0.255 10.10.10.0 0.0.0.255
```

```
lab-r2(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 any
```

```
lab-r2(config-ext-nacl)# exit
```

```
lab-r2(config)# route-map NO-NAT permit 10
```

```
lab-r2(config-route-map)# match ip address NAT-ACL
```

```
lab-r2(config)# ip nat inside source route-map NO-NAT int fa4 overload
```

! VPN-ACL

```
lab-r2(config)# ip access-list extended VPN-ACL
```

```
lab-r2(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 10.10.10.0 0.0.0.255
```

```
lab-r2(config-ext-nacl)# exit
```

```
lab-r2(config)#
```

Notice that the VPN-related ACLs have been configured on the gateway router “lab-r2” and that those ACLs are configured to control traffic from “lab-r1’s” internal private network address 192.168.1.0 to our remote office 10.10.10.0.

Now that you have some knowledge of routing protocols and routing, you may begin to think differently about the Internet. The entire Internet is made up of all sorts of independently configured networks that are interconnected using a wide range of routing protocols. Although RIP is not routed on the Internet, it is commonly used on private networks like the one introduced in this chapter.

Summary

In this chapter you learned about a few routing protocols, one in particular—RIP—you learned how to configure on a pair of Cisco routers. One router is used on an internal private network and the other is a gateway router connected to the Internet. You learned how to configure a true DMZ, which exists between your private network and your gateway. You also learned what a bastion host is and how it is used to create a more secure network.

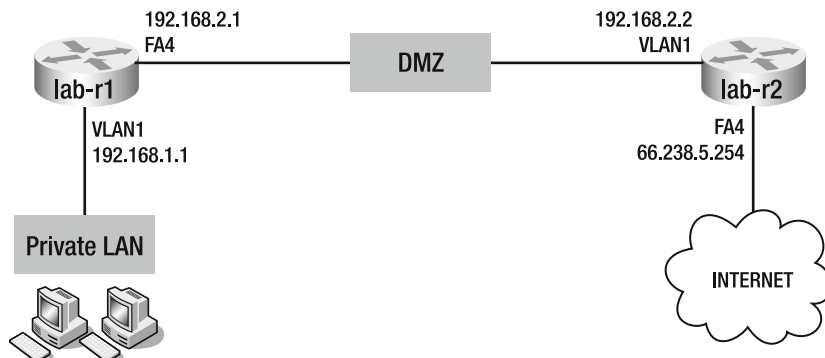


Figure 7-3. RIP network diagram

Configure RIP

Issue the user EXEC mode command `enable`, to enter privileged EXEC mode. Next, use the command `config t` to start global configuration mode. Use the configuration command `router rip` to start router configuration mode. Finally, use the `network {network-address}` command to advertise your router's networks to other RIP-enabled routers on your network. By default Cisco routers use RIPv1, which does not use subnet mask information and, thus, does not support subnetting. RIPv2 does support subnetting and supplies subnet mask information in its route advertisements. Use the command `version 2` to enable RIPv2 on your routers. For routing to occur correctly, all your routers must use the same version of RIP.

Following are the commands to enable RIPv2 on your router:

```

lab-r1> enable
lab-r1# config t
lab-r1(config)# router rip                ! enable rip
lab-r1(config-router)# version 2          ! enable rip version 2
lab-r1(config-router)# network 192.168.1.0 ! advertised network
lab-r1(config-router)# network 192.168.2.0 ! advertised network
lab-r1(config-router)# exit
lab-r1(config)#

```

Configure RIP on a Neighbor Router

For RIP to work properly, it needs a neighbor router with which to share its routes. If the neighbor router is also the gateway router (the router connected to the Internet), you use the `passive-interface` command to prevent sending or receiving RIP advertisements on your WAN interface.

Following are the commands to configure RIP routing on a gateway router that prevents RIP on interface fa4 (the WAN interface):

```
lab-r2> enable
lab-r2# config t
lab-r2(config)# router rip                ! enable rip
lab-r2(config-router)# version 2          ! enable rip version 2
lab-r2(config-router)# network 192.168.2.0 ! advertised network
lab-r2(config-router)# passive-interface fa4 ! No RIP updates on fa4
lab-r2(config-router)# exit
lab-r2(config)#
```

Verify RIP Routing

There are two commands that are commonly used to verify RIP routing. The first is `show ip protocols` and the other is `show ip route`.

Use `show ip protocols` to display information about what routing protocols are configured on your router. For example, the commands

```
lab-r1> enable
lab-r1# show ip protocols
```

On our example network, the command provides the following output:

```
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 14 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface          Send Recv Triggered RIP Key-chain
    VLAN1               1     1  2
    FastEthernet4       1     1  2
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for networks:
    192.168.1.0
    192.168.2.0
  Routing information sources:
    Gateway      Distance    Last Update
    192.168.2.2    120        00:00:16
  Distance: <default is 120>
```

Use `show ip route` to display information about what networks exist and which interface will be used on your router to get there. The following is an example of the command being issued:

```
lab-r2> enable
lab-r2# show ip route
```

The `show ip route` command displays information about your directly connected networks, RIP advertised networks, and your statically assigned networks, as shown in the following output:

```
.
.
Gateway of last resort is 66.238.5.1 to network 0.0.0.0
```

```
66.238.5.0/24 is subnetted, 1 subnets
```

```
 C   66.238.5.0 is directly connected, FastEthernet4
 R   192.168.1.0 [120/1] via 192.168.2.1, 00:00:19, VLAN1
 C   192.168.2.0/24 is directly connected, VLAN1
 S*   0.0.0.0 [1/0] via 66.238.5.1
```

Set Up a True DMZ

A true DMZ is a network that sits between two routers, known as an internal router and a gateway router. Publicly accessed servers, such as e-mail and web servers, are placed in the DMZ network. Servers in a DMZ are often called bastion hosts. These servers should be security hardened by only running the publicly accessible services that are needed to get the job done. All other services should be removed. Each of your routers is configured with its own firewall. The two firewalls work together to create a much more secure network.

VPN Configuration

When configuring an IPSec VPN using multiple routers on an endpoint network site, the actual VPN is configured on the Internet gateway routers but the NAT-ACL and VPN-ACL will use the private network address from the internal private network.



Understanding Variable Length Subnet Mask Networking

This chapter is intended for anyone who is interested in becoming Cisco certified. If you're studying for the Cisco CCNA exam, then you need to have a solid understanding of *Variable Length Subnet Mask* (VLSM) networking. I have found that most books do a poor job explaining how it works even though it's not really all that complicated. Hopefully I can do a better job!

You must understand standard subnetting before you can begin to understand VLSM networking. If you need a quick refresher on this, please refer to Chapter 6.

VLSM networking is a method of subnetting a network address using multiple subnet masks. What this means is that you use a single network address but apply a different subnet mask, per subnet, based on your need for hosts.

The reason you do this is that you can use your IP addresses more efficiently, which in turn, means you can establish more or fewer subnets per network, and more or fewer hosts per subnet than with standard subnetting. You're simply using your available address space more efficiently. This sounds far more complicated than it really is, so stick with me and I'll keep it simple.

Getting Started

Let's start by looking at a network with a standard subnet mask combination of 192.168.1.0 255.255.255.224. This network and subnet mask combination gives you eight subnets with 30 hosts per subnet. This is great if you indeed have 30 hosts on each subnet, but you may have fewer. You may have a WAN link with two routers connected to one another. In other words, you only have two hosts on a subnet that supports 30. The problem is that you're wasting 28 host addresses!

Let's look at the standard subnets for 192.168.1.0 255.255.255.224. Table 8-1 lists them, along with the range of host addresses supported by each.

Table 8-1. *Standard Subnets*

Subnet Addresses (8)		
RBlock Size 32	Hosts Range (30 hosts)	Broadcast Address
192.168.1.0	192.168.1.1-30	192.168.1.31
192.168.1.32	192.168.1.33-62	192.168.1.63
192.168.1.64	192.168.1.65-94	192.168.1.95
192.168.1.96	192.168.1.97-126	192.168.1.127
192.168.1.128	192.168.1.129-158	192.168.1.159
192.168.1.160	192.168.1.161-190	192.168.1.191
192.168.1.192	192.168.1.193-222	192.168.1.223
192.168.1.224	192.168.1.225-254	192.168.1.255

With standard subnetting, the address space falls between 0 and 256 and it has merely been divided into eight even chunks (or blocks) of 32 addresses. This is important to remember. Here’s another way to look at your 8 subnets in blocks of 32:

0			
4			
8			
12	Subnet Address	Host Range	Broadcast
16	192.168.1.0	192.168.1.1-30	192.168.1.31
20			
24			
28			
32			
36			
40			
44			
48	Subnet Address	Host Range	Broadcast
52	192.168.1.32	192.168.1.33-62	192.168.1.63
56			
60			
64			
68			
72			
76			
80	Subnet Address	Host Range	Broadcast
84	192.168.1.64	192.168.1.65-94	192.168.1.95
88			
92			
96			
100			
104			
108			
112	Subnet Address	Host Range	Broadcast
116	92.168.1.96	192.168.1.97-126	192.168.1.127
120			
124			
128			

132			
136			
140	Subnet Address	Host Range	Broadcast
144	192.168.1.128	192.168.1.129-158	192.168.1.159
248			
152			
156			
160			
164			
168			
172	Subnet Address	Host Range	Broadcast
176	192.168.1.160	192.168.1.161-19	192.168.1.191
180			
184			
188			
192			
196			
200			
204	Subnet Address	Host Range	Broadcast
208	192.168.1.192	192.168.1.193-222	192.168.1.223
212			
116			
220			
224			
228			
232			
236	Subnet Address	Host Range	Broadcast
240	192.168.1.224	192.168.1.225-254	192.168.1.255
244			
248			
252			
256			

Notice how the address space is displayed in increments of 4 (0, 4, 8, 12, 16, etc.). This is important because a block of four addresses is the smallest size into which you can divide your address space and it is the building block of VLSM networking. A block size of four allows for two hosts, one network address, and one broadcast address. Add them up and you've got the magic number four. This is the block size you would use on a WAN interface.

Planning a VLSM Network

Figure 8-1 shows a network with six subnets. Net 1 has 30 hosts and Net 2 has 14 hosts. Nets 3, 4, 5, and 6 are all WAN links so they have only two hosts per subnet (one for each router on the end of the WAN links).

The important information to note here is that you have six subnets with a total of 52 hosts. If you used your standard network/subnet mask combination of 255.255.255.224, you would waste a whole lot of address space—especially on your WAN links.

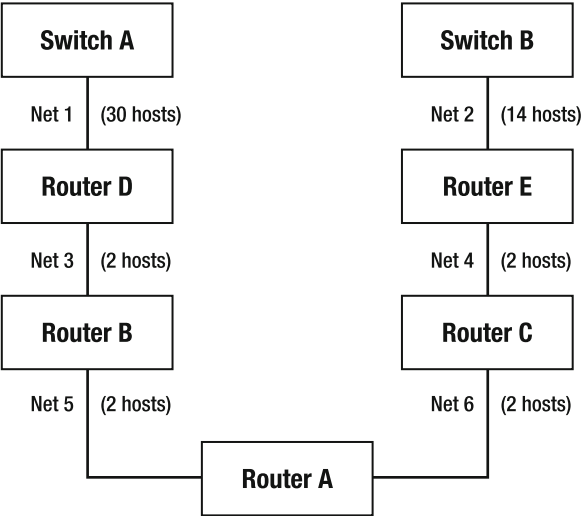


Figure 8-1. Network diagram

Now, let’s subnet the network using VLSM networking. To be efficient, layout your network in *contiguous blocks* so as not to waste a single IP address.

Note Address planning and layout are particularly important in routing. Together they allow you to summarize multiple blocks of addresses into one route entry; thus, keeping your routing tables smaller and more efficient. It’s called “supernetting” and I’ll discuss this in more detail in the next section, so keep it in the back of your mind.

The key to understanding the following subnetting scheme lies in the subnet masks—they’re different for each subnet based on that subnet’s need for hosts. Notice how each subnet mask accurately supports the correct number of hosts for that subnet:

0	
4	
8	
12	
16	Net 1 192.168.1.0 255.255.255.224 (30 hosts)
20	
24	
28	
32	

36					
40	Net 2	192.168.1.32	255.255.255.240	(14 hosts)	
44					
48					
52	Net 3	192.168.1.48	255.255.255.252	(2 hosts)	
56	Net 4	192.168.1.52	255.255.255.252	(2 hosts)	
60	Net 5	192.168.1.56	255.255.255.252	(2 hosts)	
64	Net 6	192.168.1.60	255.255.255.252	(2 hosts)	
		<i>Available address Space</i>			
256					

Following are the three subnet masks you used for subnetting:

A mask of 255.255.255.224 uses 27 bits and allows 30 hosts on a subnet.

A mask of 255.255.255.240 uses 28 bits and allows 14 hosts on a subnet.

A mask of 255.255.255.252 uses 30 bits and allows 2 hosts on a subnet.

As you can see, using three different subnet masks efficiently configures your network. What's really great about this is that you're only using one quarter of your available address space, even though you have six subnets. You have plenty of room to grow and you're not wasting IP addresses! You could still create another subnet and have 210 hosts, or more subnets with fewer hosts. The limit is the **available address space**. The key to understanding this is to know that your total address space falls between 0 and 256 and you're only using address space 0 through 63, which means the address space from 64 through 256 is still available to create more subnets.

Table 8-2 shows a more traditional look at our new subnets.

Table 8-2. *A Network Subnetted Using VLSM*

Block	Size(s)	Subnet Address	Host Range	Broadcast
Net 1	32	192.168.1.0	192.168.1.1-30	192.168.1.31
Net 2	16	192.168.1.32	192.168.1.33-46	192.168.1.47
Net 3	4	192.168.1.48	192.168.1.49-50	192.168.1.51
Net 4	4	192.168.1.52	192.168.1.53-54	192.168.1.55
Net 5	4	192.168.1.56	192.168.1.57-58	192.168.1.59
Net 6	4	192.168.1.60	192.168.1.61-62	192.168.1.63

That's all there is to VLSM networking. Again, the purpose of VLSM is to more efficiently use your available address space through the use of different subnet masks on a single network.

Route Summarization (Supernetting)

The first time I read about *route summarization*, often referred to as *supernetting*, I found it very confusing. I was confused because I didn't have a clear understanding of VLSM networking. If you understand VLSM then summarization is very straight forward to understand.

Route summarization is a way to sum up a number of subnets into a single route entry. You do this by entering a summary route that includes an IP address and subnet mask combination into your router. The summary route address covers the entire block of addresses on all your subnets. This helps keep your routing tables smaller and, especially on larger networks where you may have a lot of routing table entries, helps make your routers more efficient. Without summarization, you would have to enter each of your six subnet addresses into your router, instead of just the one.

Take another look at your VLSM network. Notice in Figure 8-2 that I've added a third interface, "interface (F0/0)" to Router A. This is the interface where you sum up all of your other subnets using a single network address and subnet mask combination.

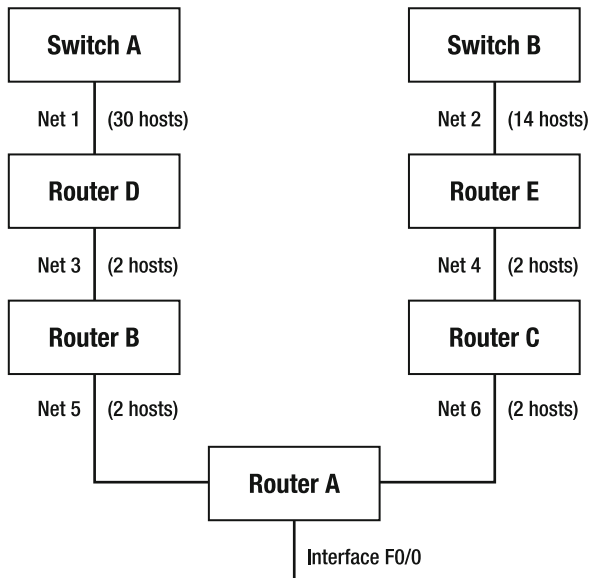


Figure 8-2. Network diagram for VLSM summarization

Remember I stated that, “address planning and layout are particularly important in routing.” What I meant was that you want your network subnets to be laid out neatly into contiguous blocks within your address space. You then configure the interfaces of your network routers using IP addresses that fall into the range of your VLSM subnets.

Take another look at your VLSM network layout, which you can see in the following example. Notice your entire network with all six subnets fits neatly into the first 64 blocks of your address space. That’s good planning.

0				
4				
8				
12	Net 1	192.168.1.0	255.255.255.224	(30 hosts)
16				
20				
24				
28				
32				
36				
40	Net 2	192.168.1.32	255.255.255.240	(14 hosts)
44				
48				
52	Net 3	192.168.1.48	255.255.255.252	(2 hosts)
56	Net 4	192.168.1.52	255.255.255.252	(2 hosts)
60	Net 5	192.168.1.56	255.255.255.252	(2 hosts)
64	Net 6	192.168.1.60	255.255.255.252	(2 hosts)

To summarize your network, you need a subnet mask combination that covers a block size of 64. Look at the network address and mask combination that covers the range

192.168.1.0 255.255.255.192

A 26-bit (255.255.255.192) subnet mask has a block size of 64. Bingo! This combination covers all the address space of your six subnets and is the summary address you would use on interface F0/0 of Router A. Table 8-3 shows what the network and host range looks like:

Table 8-3. Route Summarization Range for a Block Size of 64

Block Size	Subnet Address	Host Range	Broadcast
64	192.168.1.0	192.168.1.1-62	192.168.1.63

That’s all there is to it! Now that you have the proper mask, I’ll show you all the Cisco IOS summarization commands you need to add to Router A. You have to use a routing protocol that supports *classless routing* (subnets) and in this example it’s RIP. It could also be Cisco’s EIGRP. Listing 8-1 shows the IOS command steps for manually summarizing your 64-block address space of subnets.

Listing 8-1. *Route Summarization (Supernetting)*

```
RouterA# config t
RouterA(config)# router rip
RouterA(config-router)# version 2
RouterA(config-router)# network 192.168.1.0
RouterA(config-router)# no auto-summary
RouterA(config-router)# exit
RouterA(config)# int fa0/0
RouterA(config-if)# ip summary-address rip 192.168.1.0 255.255.255.192
```

Note I've chosen to configure the routers using RIP. In this scenario RIP would be installed on the routers that are summarized on interface fa0/0 of "Router A." EIGRP would be a better routing protocol for a large enterprise network and is what you need to know for the CCNA exam. If you wanted to summarize the network using EIGRP, use the command `ip summary-address eigrp 10 192.168.1.0 255.255.255.192`. The number "10" is the EIGRP *Autonomous System* (AS) number.

The first step is to configure RIP routing protocol for your network. The second step is to disable auto-summarization with the `no auto-summary` command. Finally, configure interface fa0/0 to summarize your networks into one entry with the command `ip summary-address rip 192.168.1.0 255.255.255.192`.

You're done. Your network has been subnetted using VLSM networking and you've trimmed Router A's RIP routing table from six entries down to one.

One thing you have to always keep in mind is that VLSM and supernetting are classless. Therefore, you have to use a routing protocol that supports *classless* networking. In a nutshell, your routing protocol has to provide subnet mask information along with its network addresses to other routers. EIGRP and RIP Version 2 are routing protocols that do this quite nicely.

Summary

This chapter covered the concepts and fundamentals of Variable Length Subnet Mask (VLSM) networking. The choice of each subnet mask is based on your need for a particular number of hosts on each subnet.

Planning a VLSM Network

Planning and layout are critical in designing a VLSM network. Blocking is a method of laying out your internetwork using blocks of four addresses. Four is the smallest block size

that can be used on a TCP/IP network because you must have one network addresses, two host addresses, and a broadcast address.

Route Summarization

A network that has been properly planned and organized using VLSM can be summarized using a single route entry on a router. This is often referred to as supernetting. Supernetting improves a router's performance by decreasing the size of its routing table. For routing tables to be properly updated using VLSM, you must use a routing protocol that supports classless routing. Classless routing protocols are used to propagate the network address information and subnet mask. Using a Cisco router, these protocols include RIP Version 2 and EIGRP. You must disable auto-summarization on your routers before you can use VLSM and supernetting to propagate routing information between the routers on your network. The following is an example using the routing protocol RIP:

```
RouterA# config t
RouterA(config)# router rip
RouterA(config-router)# version 2
RouterA(config-router)# network 192.168.1.0
RouterA(config-router)# no auto-summary
RouterA(config-router)# exit
RouterA(config)# int e1
RouterA(config-if)# ip summary-address rip 192.168.1.0 255.255.255.192
```



Sample Configuration for a Cable Modem

Appendix A displays all the Cisco IOS commands, keystroke for keystroke, that can be used to configure a Cisco 851 series router as an Internet gateway and firewall using a static IP address that has been provided by an ISP.

As an IT professional or system administrator, you should be able to quickly and easily adapt this information to meet your small business needs, regardless of whether you're using an 850 series or some other model of Cisco router.

In this example, the router will use NAT to forward SMTP e-mail messages using port 25, secure and nonsecure web server requests using ports 80 and 443, as well as Microsoft Remote Web Workplace requests using port 4125 to a Microsoft server on your private network. This configuration also supports Microsoft PPTP connections to that server using port 1723 in conjunction with Cisco's Generic Routing Encapsulation (GRE) protocol. This is a basic Cisco router configuration that will support a network with a Windows Small Business Server.

Table A-1 lists the basic configuration details for the two interfaces. The addresses and other values in the table are used in this chapter's examples. Replace them with addresses specific to your environment.

Table A-1. *IP Addresses in this Appendix's Examples*

Interface	Item	IP Address	Network Mask
fa4	WAN IP	66.238.5.254	255.255.255.0
	ISP Gateway	66.238.5.1	n/a
	ISP DNS	66.238.5.2 and 66.238.5.3	n/a
vlan1	LAN Address	192.168.1.0	255.255.255.0
	Cisco Router LAN IP	192.168.1.1	n/a
	Windows SBS Server IP	192.168.1.2	n/a

Note You can replace the information displayed in *italic* with information specific to your network to configure a Cisco router for your environment.

Standard Setup

The standard setup contains the commands used to disable logging to your console, disable the Cisco Discovery Protocol (CDP), and set your router's hostname and your company's domain name. Implement the standard setup as follows:

```
Router> ena
Router# config t
Router(config)# no logging console           ! Suppress console messages
Router(config)# no cdp run                   ! Disable Cisco discovery protocol
Router(config)# hostname lab-r1
lab-r1(config)# ip domain-name geekvenue.net ! Your domain name
```

LAN Interface

Use the following commands to set a static IP address on your LAN interface and to secure your router by disabling some Cisco default features:

```
lab-r1(config)# int vlan1
lab-r1(config-if)# descr LAN switch ports on inside interface
lab-r1(config-if)# ip address 192.168.1.1 255.255.255.0 ! Private IP address
lab-r1(config-if)# no ip unreachable
lab-r1(config-if)# no ip redirects
lab-r1(config-if)# no ip proxy-arp
lab-r1(config-if)# ip virtual-reassembly
lab-r1(config-if)# no shut
```

WAN Interface

The following example supplies all of the commands to configure your router's WAN interface using a static IP address. It shows how to set the default gateway to your ISP and configure your router to use your ISP's domain name servers to resolve Internet names to addresses.

```
lab-r1(config-if)# int fa4
lab-r1(config-if)# descr WAN interface to ISP using a STATIC IP
lab-r1(config-if)# ip address 66.238.5.254 255.25.255.0 ! Static IP from your ISP
lab-r1(config-if)# duplex auto
```

```

lab-r1(config-if)# speed auto
lab-r1(config-if)# no ip unreachable
lab-r1(config-if)# no ip redirects
lab-r1(config-if)# no ip proxy-arp
lab-r1(config-if)# ip virtual-reassembly
lab-r1(config-if)# no shut
lab-r1(config-if)# exit
lab-r1(config)# ip route 0.0.0.0 0.0.0.0 66.238.5.1 ! Provided by ISP (Gateway)
lab-r1(config)# ip name-server 66.238.5.2 66.238.5.3 ! Provided by ISP

```

Router Passwords

Use the following example to set the Cisco passwords for privileged EXEC mode and a local user on the router named “Admin” who is granted privileged EXEC mode upon logging in.

```

lab-r1(config)# enable secret MyPass ! Privileged EXEC mode password
lab-r1(config)# user Admin privilege 15 secret MyPass ! Telnet user name and pass
lab-r1(config)# line vty 0 4 ! Telnet virtual terminal
lab-r1(config-line)# login local ! Use local user accounts for telnet/SSH
lab-r1(config-line)# exit

```

NAT Setup

The following commands show how to configure an ACL for NAT that permits all private hosts access to the Internet. It also configures port forwarding for SMTP (25), HTTP (80), HTTPS (443), Remote Web Workplace (4125), and PPTP (1723).

```

lab-r1(config)# ip access-list extended NAT-ACL
lab-r1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 any ! All local hosts
lab-r1(config-ext-nacl)# exit
lab-r1(config)# ip nat inside source list NAT-ACL interface fa4 overload
lab-r1(config)# ip nat inside source static tcp 192.168.1.2 25 int fa4 25
lab-r1(config)# ip nat inside source static tcp 192.168.1.2 80 int fa4 80
lab-r1(config)# ip nat inside source static tcp 192.168.1.2 443 int fa4 443
lab-r1(config)# ip nat inside source static tcp 192.168.1.2 4125 int fa4 4125
lab-r1(config)# ip nat inside source static tcp 192.168.1.2 1723 int fa4 1723

lab-r1(config)# int vlan1 ! Your LAN switch port
lab-r1(config-if)# ip nat inside
lab-r1(config-if)# int fa4 ! Your WAN port
lab-r1(config-if)# ip nat outside
lab-r1(config-if)# exit

```

CBAC Firewall

Use the following commands to configure the Cisco CBAC firewall and the Cisco EIE.

```
lab-r1(config)# ip inspect name IPFW tcp
lab-r1(config)# ip inspect name IPFW udp
lab-r1(config)# ip inspect name IPFW cuseeme
lab-r1(config)# ip inspect name IPFW ftp
lab-r1(config)# ip inspect name IPFW tftp
lab-r1(config)# ip inspect name IPFW rcmd
lab-r1(config)# ip inspect name IPFW realaudio
lab-r1(config)# ip inspect name IPFW smtp
lab-r1(config)# ip inspect name IPFW h323
lab-r1(config)# ip inspect name IPFW sqlnet
lab-r1(config)# ip inspect name IPFW streamworks
lab-r1(config)# ip inspect name IPFW vdolive

lab-r1(config)# ip inspect name MAIL-FW smtp

lab-r1(config)# int fa4                                ! Your WAN interface
lab-r1(config-if)# ip inspect IPFW out                  ! Apply CBAC to WAN interface out
lab-r1(config-if)# ip inspect MAIL-FW in                ! Apply CBAC EIE to WAN interface in
lab-r1(config-if)# exit
```

DHCP Server

If you do not have a DHCP server running on your network, you can use the following commands to configure one on your router:

Note DHCP is not actually needed in a Windows SBS environment because SBS has its own DHCP server. However, this information is provided as an example of how to set up DHCP on a Cisco router in case you ever need to.

```
lab-r1(config)# ip dhcp pool MYNET
lab-r1(dhcp-config)# network 192.168.1.0 255.255.255.0
lab-r1(dhcp-config)# domain-name geekvenue.net          ! Your domain name
lab-r1(dhcp-config)# default-router 192.168.1.1         ! This router's address
lab-r1(dhcp-config)# dns-server 192.168.1.2            ! Your local DNS or ISPs
lab-r1(dhcp-config)# netbios-name-server 192.168.1.2    ! Your WINS (optional)
lab-r1(dhcp-config)# lease 0 2                          ! 0 days & 2 hours
lab-r1(dhcp-config)# exit
lab-r1(config)# ip dhcp excluded-address 192.168.1.1 192.168.1.25 !Your static IPs
```


IPFW Access List

Use the commands in the following example to set up an ACL that permits NAT to forward SMTP (25), HTTP (80), HTTPS (443), Remote Web Workplace (4125), and PPTP. This section also contains the rules to allow ping and traceroute replies.

```
lab-r1(config)# ip access-list extended IPFW-ACL           ! Create a named ACL

!Allow Windows Small Business Server Services
lab-r1(config-ext-nacl)# permit tcp any host 66.238.5.254 eq smtp    ! smtp
lab-r1(config-ext-nacl)# permit tcp any host 66.238.5.254 eq www      ! http
lab-r1(config-ext-nacl)# permit tcp any host 66.238.5.254 eq 443     ! https
lab-r1(config-ext-nacl)# permit tcp any host 66.238.5.254 eq 4125    ! MS RWW
lab-r1(config-ext-nacl)# permit tcp any host 66.238.5.254 eq 1723    ! MS PPTP

! Allow Outbound Ping and Traceroute
lab-r1(config-ext-nacl)# permit icmp any any administratively-prohibited
lab-r1(config-ext-nacl)# permit icmp any any echo-reply
lab-r1(config-ext-nacl)# permit icmp any any packet-too-big
lab-r1(config-ext-nacl)# permit icmp any any time-exceeded
lab-r1(config-ext-nacl)# permit icmp any any traceroute
lab-r1(config-ext-nacl)# permit gre any any
lab-r1(config-ext-nacl)# deny ip any any log                  ! Deny and log ALL traffic
lab-r1(config-ext-nacl)# exit

lab-r1(config)# int fa4                                       ! WAN interface
lab-r1(config-if)# ip access-group IPFW-ACL in               ! ACL on WAN interface
lab-r1(config-if)# exit
```

VTY Access List

Following are the commands to secure your VTY lines with an ACL that only allows local hosts access to your router via the VTY line.

```
lab-r1(config)# ip access-list standard VTY-ACL
lab-r1(config-std-nacl)# permit 192.168.1.0 0.0.0.255      ! All local hosts
lab-r1(config-std-nacl)# exit
lab-r1(config)# line vty 0 4
lab-r1(config-line)# access-class VTY-ACL in
lab-r1(config-line)# login local
lab-r1(config-line)# transport input telnet ssh
lab-r1(config-line)# exit
```

Configure SSH (Version 2)

Following are the commands to secure your VTY lines with an ACL that only allows local hosts access to your router via the VTY line. This ACL is used to permit telnet and SSH connections to your router.

```
lab-r1(config)# crypto key generate rsa
How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
lab-r1(config)#
```

Encrypt All Router Passwords

Issue the `service password-encryption` command to ensure that no clear-text passwords are stored in your router's configuration file. For example:

```
lab-r1(config)# service password-encryption
lab-r1(config)# exit
```

Save the Configuration

After configuring your router, use the following commands to save your configuration:

```
lab-r1# copy run start
Destination filename [startup-config]? {press ENTER}
```



Sample Configuration for DSL and PPPoE

Appendix B displays all the Cisco IOS commands, keystroke for keystroke, that can be used to configure a Cisco 851 series router as an Internet gateway and firewall using a static IP address that has been provided by an ISP. This appendix covers using PPPoE authentication to connect to your ISP using DSL technology such as ADSL or CDSL.

As an IT professional or system administrator, you should be able to quickly and easily adapt this information to meet your small business needs, regardless of whether you're using an 850 series or some other model of Cisco router.

In this example, the router will use NAT to forward SMTP e-mail messages using port 25, secure and nonsecure Web server requests using ports 80 and 443, as well as Microsoft Remote Web Workplace requests using port 4125 to a Microsoft server on your private network. This configuration also supports Microsoft PPTP connections to that server using port 1723 in conjunction with Cisco's GRE protocol. This is a basic Cisco router configuration that will support a network with a Windows Small Business Server.

Examples in this appendix use the following DSL username and password:

DSL USERNAME: *jason@acmeisp.com*

DSL PASSWORD: *MyISPPass*

Following is the configuration information for the dialer 1 interface:

WAN IP: *66.238.5.254 255.255.255.0*

ISP GATEWAY: *66.238.5.1*

ISP DNS: *66.238.5.2 66.238.5.3*

MTU SIZE: *1492*

And following are the details for the `vlan1` interface:

LAN ADDRESS: *192.168.1.0 255.255.255.0*

Cisco Router LAN IP: *192.168.1.1*

Windows SBS Server IP: *192.168.1.2*

Note You can (and should!) replace the information displayed in *italic* in the preceding lists with information specific to your network to configure a Cisco router for your environment.

Standard Setup

Following are the standard setup commands to disable logging on to your console, disable the Cisco Discovery Protocol (CDP), and set your router's hostname and your company's domain name.

```
Router> ena
Router# config t
Router(config)# no logging console      ! Suppress console messages
Router(config)# no cdp run              ! Disable Cisco discovery protocol
Router(config)# hostname lab-r1
lab-r1(config)# ip domain-name geekvenue.net      ! Your domain name
```

LAN Interface

Issue the following commands to set a static IP address on your LAN interface and to secure your router by disabling some Cisco default features.

```
lab-r1(config)# int vlan1
lab-r1(config-if)# descr LAN switch ports on inside interface
lab-r1(config-if)# ip address 192.168.1.1 255.255.255.0      ! Private IP address
lab-r1(config-if)# ip tcp adjust-mss 1412                    ! 1400-1500 range
lab-r1(config-if)# no ip unreachable
lab-r1(config-if)# no ip redirects
lab-r1(config-if)# no ip proxy-arp
lab-r1(config-if)# ip virtual-reassembly
lab-r1(config-if)# no shut
lab-r1(config-if)# exit
```

ENABLE PPPoE

Execute the following commands to enable PPPoE. They are only necessary if you are running an IOS version on your router prior to IOS 12.4. However, adding them will not have a negative affect on your ability to connect to your ISP using PPPoE.

```
lab-r1(config)# vpdn enable
lab-r1(config)# vpdn-group PPPOE          ! You can use any group name
lab-r1(config-vpdn)# request-dialin
lab-r1(config-vpdn-req-in)# protocol pppoe
lab-r1(config-vpdn-req-in)# exit
lab-r1(config-vpdn)# exit
```

WAN Interface (Physical)

Following are the commands to configure the physical WAN interface. When configuring DSL using PPPoE you have a physical WAN interface and virtual WAN interface known as a dialer interface. The physical interface is where you configure the PPPoE protocol. You do not assign an IP address to the physical interface.

```
lab-r1(config)# interface fa4
lab-r1(config-if)# descr Physical WAN interface used for DSL
lab-r1(config-if)# no ip address
lab-r1(config-if)# pppoe enable group global
lab-r1(config-if)# pppoe-client dial-pool-number 1
lab-r1(config-if)# duplex auto
lab-r1(config-if)# speed auto
lab-r1(config-if)# no ip unreachable
lab-r1(config-if)# no ip redirects
lab-r1(config-if)# no ip proxy-arp
lab-r1(config-if)# no shut
lab-r1(config-if)# exit
```

WAN Interface (Virtual Dialer)

Issue the following commands to configure the virtual WAN interface. Your ISP should provide you with the MTU size setting and the authentication credentials used to bring up the DSL circuit.

```
lab-r1(config)# interface dialer 1
lab-r1(config-if)# descr Logical WAN interface used for DSL
lab-r1(config-if)# ip mtu 1492          ! Provided by ISP
```

```

lab-r1(config-if)# dialer pool 1
lab-r1(config-if)# dialer-group 1
lab-r1(config-if)# no cdp enable                ! Disable CDP
lab-r1(config-if)# ip address negotiated
lab-r1(config-if)# encapsulation ppp
lab-r1(config-if)# ppp authentication chap pap callin
lab-r1(config-if)# ppp chap hostname jason@acmeisp.com    ! DSL user name
lab-r1(config-if)# ppp chap password MyISPPass           ! DSL password
lab-r1(config-if)# ppp pap sent-user jason@acmeisp.com    pass MyISPPass

lab-r1(config-if)# no ip unreachable
lab-r1(config-if)# no ip redirects
lab-r1(config-if)# no ip proxy-arp
lab-r1(config-if)# ip virtual-reassembly
lab-r1(config-if)# no shut
lab-r1(config-if)# exit
lab-r1(config)# ip route 0.0.0.0 0.0.0.0 dialer 1

```

Router Passwords

Use the following commands to set the Cisco passwords for privileged EXEC mode and a local user on the router named “Admin” who is granted privileged EXEC mode upon logging in.

```

lab-r1(config)# enable secret MyPass            ! Privileged EXEC mode password
lab-r1(config)# user Admin privilege 15 secret MyPass ! Telnet username/password
lab-r1(config)# line vty 0 4                    ! Telnet virtual terminal
lab-r1(config-line)# login local                 ! Use local user account for telnet/ssh
lab-r1(config-line)# exit

```

NAT Setup

Issue the following commands to configure an ACL for NAT that permits all private hosts access to the Internet. The commands also configure port forwarding for SMTP (25), HTTP (80), HTTPS (443), Remote Web Workplace (4125), and PPTP (1723).

```

lab-r1(config)# ip access-list extended NAT-ACL
lab-r1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 any    ! All local hosts
lab-r1(config-ext-nacl)# exit
lab-r1(config)# ip nat inside source list NAT-ACL interface dialer 1 overload
lab-r1(config)# ip nat inside source static tcp 192.168.1.2 25 int dial 1 25
lab-r1(config)# ip nat inside source static tcp 192.168.1.2 80 int dial 1 80
lab-r1(config)# ip nat inside source static tcp 192.168.1.2 443 int dial 1 443

```

```

lab-r1(config)# ip nat inside source static tcp 192.168.1.2 4125 int dial 1 4125
lab-r1(config)# ip nat inside source static tcp 192.168.1.2 1723 int dial 1 1723

lab-r1(config)# int vlan1                                ! Your LAN switch port
lab-r1(config-if)# ip nat inside
lab-r1(config-if)# int dialer 1                          ! Your virtual WAN port
lab-r1(config-if)# ip nat outside
lab-r1(config-if)# exit

```

CBAC Firewall

Execute the following commands to configure the Cisco CBAC firewall and the Cisco EIE.

```

lab-r1(config)# ip inspect name IPFW tcp
lab-r1(config)# ip inspect name IPFW udp
lab-r1(config)# ip inspect name IPFW cuseeme
lab-r1(config)# ip inspect name IPFW ftp
lab-r1(config)# ip inspect name IPFW tftp
lab-r1(config)# ip inspect name IPFW rcmd
lab-r1(config)# ip inspect name IPFW realaudio
lab-r1(config)# ip inspect name IPFW smtp
lab-r1(config)# ip inspect name IPFW h323
lab-r1(config)# ip inspect name IPFW sqlnet
lab-r1(config)# ip inspect name IPFW streamworks
lab-r1(config)# ip inspect name IPFW vdolive

lab-r1(config)# ip inspect name MAIL-FW smtp

lab-r1(config)# int dialer 1                                ! Your virtual WAN interface
lab-r1(config-if)# ip inspect IPFW out                    ! Apply CBAC to WAN interface out
lab-r1(config-if)# ip inspect MAIL-FW in                  ! Apply CBAC EIE to WAN interface in
lab-r1(config-if)# exit

```

DHCP Server

If you do not have a DHCP server running on your network, you can issue the following commands to configure one on your router:

Note DHCP is not actually needed in a Windows SBS environment because SBS has its own DHCP server. However, this information is provided as an example of how to set up DHCP on a Cisco router in case you ever need to.

```

lab-r1(config)# ip dhcp pool MYNET
lab-r1(dhcp-config)# network 192.168.1.0 255.255.255.0
lab-r1(dhcp-config)# domain-name geekvenue.net          ! Your domain name
lab-r1(dhcp-config)# default-router 192.168.1.1         ! This router's address
lab-r1(dhcp-config)# dns-server 192.168.1.2            ! Your local DNS or ISPs
lab-r1(dhcp-config)# netbios-name-server 192.168.1.2    ! Your WINS (optional)
lab-r1(dhcp-config)# lease 0 2                         ! 0 days & 2 hours
lab-r1(dhcp-config)# exit
lab-r1(config)# ip dhcp excluded-address 192.168.1.1 192.168.1.25 !Your static IPs

```

IPFW Access List

Following are the commands to set up an ACL that permits NAT to forward SMTP (25), HTTP (80), HTTPS (443), Remote Web Workplace (4125), and PPTP. The commands also create the rules to allow ping and traceroute replies.

```

lab-r1(config)# ip access-list extended IPFW-ACL          ! Create a named ACL

! Allow Windows Small Business Server Services
lab-r1(config-ext-nacl)# permit tcp any host 66.238.5.254 eq smtp    ! smtp
lab-r1(config-ext-nacl)# permit tcp any host 66.238.5.254 eq www      ! http
lab-r1(config-ext-nacl)# permit tcp any host 66.238.5.254 eq 443     ! https
lab-r1(config-ext-nacl)# permit tcp any host 66.238.5.254 eq 4125    ! MS RWW
lab-r1(config-ext-nacl)# permit tcp any host 66.238.5.254 eq 1723    ! PPTP

! Allow outbound Ping and Traceroute
lab-r1(config-ext-nacl)# permit icmp any any administratively-prohibited
lab-r1(config-ext-nacl)# permit icmp any any echo-reply
lab-r1(config-ext-nacl)# permit icmp any any packet-too-big
lab-r1(config-ext-nacl)# permit icmp any any time-exceeded
lab-r1(config-ext-nacl)# permit icmp any any traceroute
lab-r1(config-ext-nacl)# permit gre any any
lab-r1(config-ext-nacl)# deny ip any any log                  ! Deny and log ALL traffic
lab-r1(config-ext-nacl)# exit

lab-r1(config)# int dialer 1                                ! Virtual WAN interface
lab-r1(config-if)# ip access-group IPFW-ACL in              ! ACL on WAN interface
lab-r1(config-if)# exit

```

VTY Access List

Following are the commands to secure your VTY lines with an ACL that only allows local hosts access to your router via the VTY line. This ACL is used to permit telnet and SSH connections to your router.


```
lab-r1(config)# ip access-list standard VTY-ACL
lab-r1(config-std-nacl)# permit 192.168.1.0 0.0.0.255      ! All local hosts
lab-r1(config-std-nacl)# exit
lab-r1(config)# line vty 0 4
lab-r1(config-line)# access-class VTY-ACL in
lab-r1(config-line)# login local
lab-r1(config-line)# transport input telnet ssh
lab-r1(config-line)# exit
```

Configure SSH (Version 2)

Issue the following commands to configure your router with an SSH server so that you can securely configure your router from an SSH client.

```
lab-r1(config)# crypto key generate rsa
How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
lab-r1(config)#
```

Encrypt All Router Passwords

Issue the service password-encryption command to ensure that no clear-text passwords are stored in your router's configuration file. For example:

```
lab-r1(config)# service password-encryption
lab-r1(config)# exit
```

Save the Configuration

After configuring your router, use the following commands to save your configuration:

```
lab-r1# copy run start
Destination filename [startup-config]? {press ENTER}
```



Sample Configuration IPSec VPN Over DSL

Appendix C displays all the Cisco IOS commands, keystroke for keystroke, that can be used to configure a Cisco 851 series router as an Internet gateway and firewall using a static IP address that has been provided by an ISP. In this example, you will be configuring a site-to-site IPSec VPN from your Branch office to a Corporate office. The configuration in this appendix is for the router located at the Branch office.

As an IT professional or system administrator, you should be able to quickly and easily adapt this information to meet your small business needs, regardless of whether you're using an 850 series or some other model of Cisco router.

In this example, the router will use NAT to forward SMTP e-mail messages using port 25, secure and nonsecure Web server requests using ports 80 and 443, as well as Microsoft Remote Web Workplace requests using port 4125 to a Microsoft server on your private network. This configuration also supports Microsoft PPTP connections to that server using port 1723 in conjunction with Cisco's GRE protocol. This is a basic Cisco router configuration that will support a network with a Windows Small Business Server.

Following is the branch office configuration:

DSL USERNAME: *jason@acmeisp.com*

DSL PASSWORD: *MyISPass*

WAN IP: *66.238.5.254 255.255.255.0* **! Interface dialer 1**

ISP GATEWAY: *66.238.5.1*

ISP DNS: *66.238.5.2 66.238.5.3*

MTU SIZE: *1492*

LAN ADDRESS: *192.168.1.0 255.255.255.0* ! Interface vlan1
Cisco Router LAN IP: *192.168.1.1*
Windows SBS Server IP: *192.168.1.2*
 And following is the corporate office configuration:
REMOTE WAN ADDRESS: *24.237.8.112*
REMOTE LAN ADDRESS: *10.10.10.0*
VPN Name: *CORP-VPN*
VPN Pre Shared Key: *Fre@kSh0w!*

Note You can replace the information displayed in *italic* with information specific to your network to configure a Cisco router for your environment.

Standard Setup

Issue the following, standard setup commands to disable logging on to your console, disable the Cisco Discovery Protocol (CDP), and set your router's hostname and your company's domain name.

```

Router> ena
Router# config t
Router(config)# no logging console          ! Suppress console messages
Router(config)# no cdp run                  ! Disable Cisco discovery protocol
Router(config)# hostname lab-r1
lab-r1(config)# ip domain-name geekvenue.net ! Your domain name
  
```

LAN Interface

Use the following commands to set a static IP address on your LAN interface and to secure your router by disabling some Cisco default features.

```

lab-r1(config)# int vlan1
lab-r1(config-if)# descr LAN switch ports on inside interface
lab-r1(config-if)# ip address 192.168.1.1 255.255.255.0 ! Private IP address
lab-r1(config-if)# ip tcp adjust-mss 1412                ! 1400-1500 range
lab-r1(config-if)# no ip unreachable
lab-r1(config-if)# no ip redirects
lab-r1(config-if)# no ip proxy-arp
  
```

```
lab-r1(config-if)# ip virtual-reassembly
lab-r1(config-if)# no shut
lab-r1(config-if)# exit
```

Enable PPPoE

Issue the following commands to enable PPPoE. They are only necessary if you are running an IOS version on your router prior to IOS 12.4. However, adding them will not have a negative affect on your ability to connect to your ISP using PPPoE.

```
lab-r1(config)# vpdn enable
lab-r1(config)# vpdn-group PPPOE ! You can use any group name
lab-r1(config-vpdn)# request-dialin
lab-r1(config-vpdn-req-in)# protocol pppoe
lab-r1(config-vpdn-req-in)# exit
lab-r1(config-vpdn)# exit
```

WAN Interface (Physical)

Following are the commands to configure a physical WAN interface. When configuring DSL using PPPoE you have a physical WAN interface and virtual WAN interface known as a dialer interface. The physical interface is where you configure the PPPoE protocol. You do not assign an IP address to the physical interface.

```
lab-r1(config)# interface fa4
lab-r1(config-if)# descr Physical WAN interface used for DSL
lab-r1(config-if)# no ip address
lab-r1(config-if)# pppoe enable group global
lab-r1(config-if)# pppoe-client dial-pool-number 1
lab-r1(config-if)# duplex auto
lab-r1(config-if)# speed auto
lab-r1(config-if)# no ip unreachable
lab-r1(config-if)# no ip redirects
lab-r1(config-if)# no ip proxy-arp
lab-r1(config-if)# no shut
lab-r1(config-if)# exit
```

WAN Interface (Virtual Dialer)

Execute the following commands to configure the virtual WAN interface. Your ISP should provide you with the MTU size setting and the authentication credentials used to bring up the DSL circuit.

```

lab-r1(config)# interface dialer 1
lab-r1(config-if)# descr Logical WAN interface used for DSL
lab-r1(config-if)# ip mtu 1492                                ! Provided by ISP
lab-r1(config-if)# dialer pool 1
lab-r1(config-if)# dialer-group 1
lab-r1(config-if)# no cdp enable                              ! Disable CDP
lab-r1(config-if)# ip address negotiated
lab-r1(config-if)# encapsulation ppp
lab-r1(config-if)# ppp authentication chap pap callin
lab-r1(config-if)# ppp chap hostname jason@acmeisp.com        ! DSL user name
lab-r1(config-if)# ppp chap password MyISPPass               ! DSL Password
lab-r1(config-if)# ppp pap sent-user jason@acmeisp.com pass MyISPPass
lab-r1(config-if)# no ip unreachable
lab-r1(config-if)# no ip redirects
lab-r1(config-if)# no ip proxy-arp
lab-r1(config-if)# ip virtual-reassembly
lab-r1(config-if)# no shut
lab-r1(config-if)# exit
lab-r1(config)# ip route 0.0.0.0 0.0.0.0 dialer 1

```

Router Passwords

Use the following commands to set the Cisco passwords for privileged EXEC mode and a local user on the router named “Admin” who is granted privileged EXEC mode upon logging in.

```

lab-r1(config)# enable secret MyPass                        ! Privileged EXEC mode password
lab-r1(config)#user Admin privilege 15 secret MyPass        ! Telnet username/password
lab-r1(config)# line vty 0 4                                ! Telnet virtual terminal
lab-r1(config-line)# login local                            ! Use local user account for telnet log ins
lab-r1(config-line)# exit

```

NAT Setup

Execute the following commands to configure an ACL for NAT that permits all private hosts access to the Internet. It also configures port forwarding for SMTP (25), HTTP (80), HTTPS (443), Remote Web Workplace (4125), and PPTP (1723).

```

lab-r1(config)# ip access-list extended NAT-ACL

lab-r1(config-ext-nacl)# deny ip 192.168.1.0 0.0.0.255 10.10.10.0 0.0.0.255
lab-r1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 any
lab-r1(config-ext-nacl)# exit

```

```

lab-r1(config)# route-map NO-NAT permit 10                                ! VPN routing policy
lab-r1(config-route-map)# match ip address NAT-ACL
lab-r1(config-route-map)# exit
lab-r1(config)# ip nat inside source route-map NO-NAT int dial 1 overload

lab-r1(config)# int loopback0
lab-r1(config-if)# ip address 1.1.1.1 255.255.255.252
lab-r1(config-if)# exit

lab-r1(config)# ip access-list ext NONAT-LAN-ACL
lab-r1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 10.10.10.0 0.0.0.255
lab-r1(config-ext-nacl)# exit

lab-r1(config)# route-map NONAT-LAN
lab-r1(config-route-map)# match ip address NONAT-LAN-ACL
lab-r1(config-route-map)# set interface loopback0
lab-r1(config-route-map)# exit

lab-r1(config)# int vlan1
lab-r1(config-if)# ip policy route-map NONAT-LAN
lab-r1(config-if)# exit

lab-r1(config)# ip nat inside source static tcp 192.168.1.2 25 int dial 1 25
lab-r1(config)# ip nat inside source static tcp 192.168.1.2 80 int dial 1 80
lab-r1(config)# ip nat inside source static tcp 192.168.1.2 443 int dial 1 443
lab-r1(config)# ip nat inside source static tcp 192.168.1.2 4125 int dial 1 4125
lab-r1(config)# ip nat inside source static tcp 192.168.1.2 1723 int dial 1 1723

lab-r1(config)# int vlan1                                              ! Your LAN switch port
lab-r1(config-if)# ip nat inside
lab-r1(config-if)# int dialer 1                                       ! Your virtual WAN port
lab-r1(config-if)# ip nat outside
lab-r1(config-if)# exit

```

CBAC Firewall

The following commands configure the Cisco CBAC firewall and the Cisco EIE.

```

lab-r1(config)# ip inspect name IPFW tcp
lab-r1(config)# ip inspect name IPFW udp
lab-r1(config)# ip inspect name IPFW cuseeme
lab-r1(config)# ip inspect name IPFW ftp
lab-r1(config)# ip inspect name IPFW tftp
lab-r1(config)# ip inspect name IPFW rcmd
lab-r1(config)# ip inspect name IPFW realaudio

```

```

lab-r1(config)# ip inspect name IPFW smtp
lab-r1(config)# ip inspect name IPFW h323
lab-r1(config)# ip inspect name IPFW sqlnet
lab-r1(config)# ip inspect name IPFW streamworks
lab-r1(config)# ip inspect name IPFW vdolive

lab-r1(config)# ip inspect name MAIL-FW smtp

lab-r1(config)# int dialer 1                ! Your virtual WAN interface
lab-r1(config-if)# ip inspect IPFW out      ! Apply CBAC to WAN interface out
lab-r1(config-if)# ip inspect MAIL-FW in    ! Apply CBAC EIE to WAN interface in
lab-r1(config-if)# exit

```

VPN Cryptographic Settings

The following commands implement the phase 1 and phase 2 cryptographic settings used to set up an IPsec VPN. You will use a preshared key and AES 256-bit encryption to authenticate and encrypt data between sites.

```

lab-r1(config)# crypto isakmp policy 1                ! Phase 1 (IKE)
lab-r1(config-isakmp)# encryption aes 256
lab-r1(config-isakmp)# hash sha
lab-r1(config-isakmp)# authentication pre-share
lab-r1(config-isakmp)# group 2
lab-r1(config-isakmp)# exit
lab-r1(config)# crypto isakmp key Fre@kSh0w! Address 24.237.8.112 ! Corp IP
lab-r1(config)# crypto ipsec security-association lifetime seconds 28800

lab-r1(config)# ip access-list extended VPN-ACL
lab-r1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 10.10.10.0 0.0.0.255
lab-r11(config-ext-nacl)# exit

lab-r1(config)# crypto ipsec transform-set SET1 esp-aes 256 esp-sha-hmac

lab-r1(config)# crypto map CORP-VPN 10 ipsec-isakmp                ! Phase 2 (IPSec)
lab-r1(config-crypto-map)# set peer 24.237.8.112
lab-r1(config-crypto-map)# set transform-set SET1
lab-r1(config-crypto-map)# set pfs group2
lab-r1(config-crypto-map)# match address VPN-ACL
lab-r1(config-crypto-map)# exit

```

DHCP Server

If you do not have a DHCP server running on your network, you can use the following commands to configure one on your router:

Note DHCP is not actually needed in a Windows SBS environment because SBS has its own DHCP server. This information is provided as an example of how to set up DHCP on a Cisco router in case you ever need to.

```
lab-r1(config)# ip dhcp pool MYNET
lab-r1(dhcp-config)# network 192.168.1.0 255.255.255.0
lab-r1(dhcp-config)# domain-name geekvenue.net           ! Your domain name
lab-r1(dhcp-config)# default-router 192.168.1.1          ! This router's address
lab-r1(dhcp-config)# dns-server 192.168.1.2              ! Your local DNS or ISPs
lab-r1(dhcp-config)# netbios-name-server 192.168.1.2     ! Your WINS (optional)
lab-r1(dhcp-config)# lease 0 2                           ! 0 days & 2 hours
lab-r1(dhcp-config)# exit
lab-r1(config)# ip dhcp excluded-address 192.168.1.1 192.168.1.25 !Your static IPs
```

IPFW Access List

Issue the following commands to set up an ACL that permits NAT to forward SMTP (25), HTTP (80), HTTPS (443), Remote Web Workplace (4125), and PPTP. The commands also implement the rules to allow ping and traceroute replies.

```
lab-r1(config)# ip access-list extended IPFW-ACL          ! Create a named ACL

! Allow VPN Setup
lab-r1(config-ext-nacl)# permit udp any any eq isakmp
lab-r1(config-ext-nacl)# permit udp any eq isakmp any
lab-r1(config-ext-nacl)# permit esp any any

! Allow Windows Small Business Server Services
lab-r1(config-ext-nacl)# permit tcp any host 66.238.5.254 eq smtp ! smtp
lab-r1(config-ext-nacl)# permit tcp any host 66.238.5.254 eq www   ! http
lab-r1(config-ext-nacl)# permit tcp any host 66.238.5.254 eq 443   ! https
lab-r1(config-ext-nacl)# permit tcp any host 66.238.5.254 eq 4125  ! MS RWW
lab-r1(config-ext-nacl)# permit tcp any host 66.238.5.254 eq 1723  ! PPTP
```



```

! Allow outbound Ping and Traceroute
lab-r1(config-ext-nacl)# permit icmp any any administratively-prohibited
lab-r1(config-ext-nacl)# permit icmp any any echo-reply
lab-r1(config-ext-nacl)# permit icmp any any packet-too-big
lab-r1(config-ext-nacl)# permit icmp any any time-exceeded
lab-r1(config-ext-nacl)# permit icmp any any traceroute
lab-r1(config-ext-nacl)# permit gre any any
lab-r1(config-ext-nacl)# deny ip any any log           ! Deny and log ALL traffic
lab-r1(config-ext-nacl)# exit

lab-r1(config)# int dialer 1                           ! Virtual WAN interface
lab-r1(config-if)# ip access-group IPFW-ACL in         ! ACL on WAN interface
lab-r1(config-if)# crypto map CORP-VPN                ! Apply VPN to WAN interface
lab-r1(config-if)# exit

```

VTY Access List

Here are the commands to secure your VTY lines with an ACL that only allows local hosts access to your router via the VTY line. This ACL is used to permit telnet and SSH connections to your router.

```

lab-r1(config)# ip access-list standard VTY-ACL
lab-r1(config-std-nacl)# permit 192.168.1.0 0.0.0.255 ! All local hosts
lab-r1(config-std-nacl)# exit
lab-r1(config)# line vty 0 4
lab-r1(config-line)# access-class VTY-ACL in
lab-r1(config-line)# login local
lab-r1(config-line)# transport input telnet ssh
lab-r1(config-line)# exit

```

Configure SSH (Version 2)

The following commands configure your router with an SSH server so that you can securely configure your router of IP with an SSH client:

```

lab-r1(config)# crypto key generate rsa
How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
lab-r1(config)#

```

Encrypt All Router Passwords

The `service password-encryption` command will ensure that no clear-text passwords are stored in your router's configuration file. For example:

```
lab-r1(config)# service password-encryption
lab-r1(config)# exit
```

Save the Configuration

After configuring your router, use the following commands to save your configuration:

```
lab-r1# copy run start
Destination filename [startup-config]? {press ENTER}
```



CCNA CLI Command Reference

This appendix is a command reference designed for anyone preparing for the *Cisco Certified Network Associate* (CCNA) exam. It describes the commands necessary to complete networking tasks on Cisco routers and switches. This information is not intended to provide in-depth details, but instead should be used as a checklist of the concepts and commands specific to the exam.

Cisco Router Commands

The commands are grouped by the following topics:

Table D-1. *Router Command Topics*

Access Control List (ACL)	Network Address Translation (NAT)
Backup and Restore the IOS	Password—Encrypting
Cisco Discovery Protocol (CDP)	Password—Setting
Command History	PPP Configuration
Configuration Register Commands	Routing—Default Routes
Password Recovery	Routing—EIGRP
Console Messages	Routing—IGRP
Date and Time	Routing—OSPF
DHCP Configuration	Routing—RIP
DNS Lookup	Routing—Static Routes
Frame-Relay	Secure Shell (SSH)
Hostname and Message of the Day (MOTD)	Startup-Config and Running-Config Files
Interface—Configuration	Telnet
Interface—Verifying TCP/IP Configurations	VTY ACL for Telnet and SSH

Access Control List (ACL)

! There is an implicit **deny** at the end of every access list
! Standard access lists <1-99> and <1300-1399> (expanded)

! Note: Always place a standard ACL as close to the destination as possible

! Denying a Single Host (two ways)

```
lab-r1# config t
lab-r1(config)# access-list 10 deny host 192.168.10.1
lab-r1(config)# access-list 10 deny 192.168.10.1 0.0.0.0
```

! Denying an Entire Network or Subnet

```
lab-r1# config t
lab-r1(config)# access-list 10 deny 192.168.10.0 0.0.0.255
```

! Denying a host range on a network/subnet

! For an ACL range, think in terms of block sizes 4, 8, 16, 32, 64
! and subtract 1 from the wildcard mask (because you start at 0)

! In the next example, you deny

! hosts 172.16.0.1 through 172.16.15.255 (a block size of 16)

```
lab-r1# config t
lab-r1(config)# access-list 10 deny 172.16.0.0 0.0.15.255
```

! Denying a host range on a network/subnet. In this example, you deny

! a host range of 192.168.10.1 through 192.168.10.7 (a block size of 8)

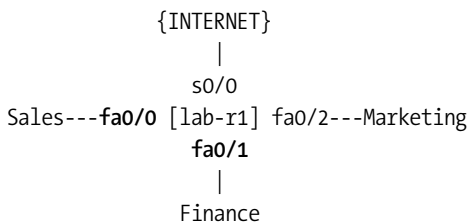
```
lab-r1# config t
lab-r1(config)# access-list 10 deny 192.168.10.0 0.0.0.7
```

! Access list Examples

! Deny Sales access to Finance LAN, but allow all others

! Sales network is 192.168.10.0/24 on fa0/0

! Finance is on Interface Fa0/1



```
lab-r1# config t
lab-r1(config)# access-list 10 deny 192.168.10.0 0.0.0.255
lab-r1(config)# access-list 10 permit 192.168.0.0 0.0.255.255
lab-r1(config)# int fa0/1
lab-r1(config-if)# ip access-group 10 out
```

! Extended access lists <100-199> <2000-2699> (expanded)

Note Always place a standard ACL as close to the source as possible.

! Deny SMTP Access (port 25) to a Single Host

```
lab-r1# config t
lab-r1(config)# access-list 110 deny tcp any host 192.168.10.1 eq 25
lab-r1(config)# access-list 110 permit ip any any
```

! Deny SMTP Access (port 25) to an Entire Network

```
lab-r1# config t
lab-r1(config)# access-list 110 deny tcp any 192.168.10.0 0.0.0.255 eq smtp
lab-r1(config)# access-list 110 permit ip any any
! Applying an extended access-list to an Interface
lab-r1# config t
lab-r1(config)# int fa0/1
lab-r1(config-if)# ip access-group 110 out (or in)
```

! Standard named access-list

! This ACL will deny an entire network/subnet but allow others

```
lab-r1# config t
lab-r1(config)# ip access-list standard SalesDeny
lab-r1(config-std-nacl)# deny 192.168.1.0 0.0.0.255
lab-r1(config-std-nacl)# permit any
lab-r1(config-std-nacl)# exit
```

! Applying a standard named access-list to an interface

```
lab-r1# config t
lab-r1(config)# int f0/0
lab-r1(config-if)# ip access-group SalesDeny out
```

! Removing standard named access-lists

```
lab-r1# config t
lab-r1(config)# int f0/0
lab-r1(config-if)# no ip access-group SalesDeny out
lab-r1(config-if)# exit
lab-r1(config)# no ip access-list standard SalesDeny
```

Backup and Restore the IOS

! Verify the Information in Flash

lab-r1# **sh flash**

! Displays IOS Image int flash

File	Length	Name/Status
1	8122000	c2500-js-l-112-18.bin

! Backing up the Cisco IOS

lab-r1# **copy flash tftp**

Address or name of remote host [255.255.255.255]? *192.168.10.2*

Source filename? *C2500-js-l-112-18.bin*

Destination filename [c2500-js-l-112-18.bin]?

! Restoring the Cisco IOS (using a tftp server)

lab-r1# **copy tftp flash**

Proceed[confirm]? {ENTER}

Source filename? *C2500-js-l-112-18.bin*

Destination filename [c2500-js-l-112-18.bin]?

! Backing up the config file to a tftp server (either run or start)

lab-r1# **copy run tftp**

Address or name of remote host? *192.168.10.2*

! tftp server's IP

Destination filename[router-config]? *my-startup-config*

! Restoring the config file from a tftp server

lab-r1# **copy tftp run**

Address or name of remote host? *192.168.10.2*

! tftp server's IP

Source filename[]? *my-startup-config*

Destination filename[router-config]? [enter]

! Erasing the startup-config

lab-r1# **erase start**

Cisco Discovery Protocol (CDP)

! Gathering CDP neighbor information

lab-r1# **sh cdp neighbor**

lab-r1# **sh cdp neighbor detail**

lab-r1# **sh cdp entry ***

! Same as 'sh cdp neighbor detail'

```
! Gathering CDP traffic information
! Displays number of packets sent/received and any errors
lab-r1# sh cdp traffic

! Gathering CDP port and interface information
lab-r1# sh cdp interface

! Display timer and holdtime
lab-r1# sh cdp

    Sending CDP packets every 60 seconds
    Sending a holdtime value of 180 second

! Changing CDP timer and holdtime
lab-r1# cdp timer 120
lab-r1# cdp holdtime 360
lab-r1# no cdp enable                                ! Turns CDP off for the router

! Turning off CDP on an Interface
lab-r1# config t
lab-r1(config)# int s0/0
lab-r1(config-if)# no cdp enable
```

Command History

```
lab-r1# show history                                ! Display last 10 commands
lab-r1# show terminal                                ! Used to verify history size
lab-r1# terminal history size {size}                 ! Sets history buffer size
lab-r1# show version                                 ! Displays hw/sw ver+cfg files
```

Configuration Register Commands

```
! Checking the configuration register
lab-r1# sh version                                ! Displays active configuration register

Configuration register is 0x2102
! A configuration register of 0x2102 loads the startup-config
! A configuration register of 0x2142 bypasses the startup-config
```

Password Recovery

! Send ctrl-break at startup & change config-reg to 0x2142
 ! Reload the router's configuration
 ! Enter privileged mode & copy startup-config to running-config
 ! Change the password, change config-register back to 0x2102
 ! Save the config and reload the router

! Recovering a Lost Password on a 2500 Series Router

[ctrl-break] at startup

>

> o/r 0x2142 ! Bypass the startup-config

> I ! Initialize (save the new config)

! The Router will restart and ask for setup mode- answer NO

! to get a prompt

```
router> enable
```

```
router# config t
```

```
router(config)# copy start run
```

```
lab-r1(config)# enable secret {new password}
```

```
lab-r1(config)# copy run start
```

```
lab-r1(config)# config-register 0x2102 ! Enables startup-config
```

```
lab-r1(config)# ^Z
```

```
lab-r1# reload
```

! Recovering a lost Password on a 2600 series router

[ctrl-break] at startup

```
rommon 1> confreg 0x2142 ! Bypass the startup-config
```

```
rommon 2> reset
```

! The router will restart and ask for setup mode- answer NO

! to get a prompt

```
router> enable
```

```
router# config t
```

```
router(config)# copy start run
```

```
lab-r1(config)# enable secret {new password}
```

```
lab-r1(config)# copy run start
```

```
lab-r1(config)# config-register 0x2102 ! Enables startup-config
```

```
lab-r1(config)# ^Z
```

```
lab-r1# reload
```

! END Password Recovery Example

Console Messages

! Stop the Console from displaying messages
lab-r1(config)# **no logging console**

Date and Time

! Setting the Date and Time
lab-r1# **clock set 15:40:22 10 AUG 2007**

! Displaying Date and Time
lab-r1# **sh clock**

DHCP Configuration

! Setting up a DHCP Server
lab-r1(config)#**ip dhcp pool DHCP_POOL1**
lab-r1(dhcp-config)# **import all**
lab-r1(dhcp-config)# **network 10.10.10.0 255.255.255.0**
lab-r1(dhcp-config)# **domain-name geekvenue.net**
lab-r1(dhcp-config)# **dns-server 192.168.10.2 192.168.10.3**
lab-r1(dhcp-config)# **lease 0 2** ! 0 days 2 hours
lab-r1(dhcp-config)# **exit**
lab-r1(config)# **ip dhcp excluded-address 10.10.10.1**

DNS Lookup

! Enabling DNS lookup
lab-r1# **config t**
lab-r1(config)# **ip domain-lookup**
lab-r1(config)# **ip name-server {IP address(es) used for DNS}**

! Using DNS to Resolve Hostnames
lab-r1# **config t**
lab-r1(config)# **ip domain-lookup**
lab-r1(config)# **ip name-server 192.168.10.2**
lab-r1(config)# **ip domain-name geekvenue.net**

! Checking Network Connectivity
lab-r1# **ping {IP address}**
lab-r1# **traceroute {IP address}**

Frame-Relay

! Configuring Frame-Relay on a Single Interface
! If connecting two routers directly together, use the same dlci # on both

```
lab-r1# config t
lab-r1(config)# int s0/0
lab-r1(config-if)# no shutdown
lab-r1(config-if)# encapsulation frame-relay
lab-r1(config-if)# frame-relay lmi-type ansi    ! cisco, ansi or q933a
lab-r1(config-if)# ip address 192.168.1.1 255.255.255.0
lab-r1(config-if)# frame-relay interface-dlci 100
lab-r1(config-if)# ^Z
```

! Configuring Frame-Relay on a Sub-interface

! Set Encapsulation to frame-relay
! Set the sub-interface to point-to-point
! Give the sub-interface an IP address
! Set frame-relay interface-dlci numbers

```
lab-r1# config t
lab-r1(config)# int s0/0
lab-r1(config-if)# no ip address
lab-r1(config-if)# no shutdown
lab-r1(config-if)# encapsulation frame-relay
lab-r1(config-if)# int s0/0.18 point-to-point    !sub-if # same as dlci #
lab-r1(config-subif)# ip address 192.16.10.1 255.255.255.0
lab-r1(config-subif)# frame-relay interface-dlci 18
```

Hostname and Message of the Day (MOTD)

```
router> enable
router# config terminal
router(config)# hostname lab-r1
lab-r1(config)# banner motd # Some message... #
```

Interface—Configuration

! Set Interface Descriptions
lab-r1(config)# int e0
lab-r1(config-if)# description {some description}

```
! Viewing Interface Descriptions
lab-r1# sh run                ! Displays all interfaces and their descriptions
lab-r1# sh int e0             ! Displays info for specific interface (e0,s0 etc)

! Router Interfaces
lab-r1(config)# int s5        ! Configure Serial Interface no. 5
lab-r1(config-if)#

lab-r1(config)# int e1        ! Config Ethernet Interface no. 1
lab-r1(config-if)#

lab-r1(config)# int s0/0/0    ! Slot/Port and Interface no.
lab-r1(config-if)#

lab-r1(config)# int fast 0/1  ! Config Ether Int Slot 0/Port 1 (2600r)
lab-r1(config-if)#

! Bringing up an Interface
lab-r1(config)# int e1
lab-r1(config-if)# no shut    ! Brings up the Interface

! Configuring IP on an Interface
lab-r1(config)# int e1
lab-r1(config-if)# ip address 192.168.10.1 255.255.255.0
lab-r1(config-if)# no shut

! Configuring a Serial Interface
lab-r1(config)# int s0
lab-r1(config-if)# clock rate 64000    ! Bits per second
lab-r1(config-if)# bandwidth 64        ! Used by IGRP, EIGRP & OSPF
```

Note Clocking is usually provided by a CSU/DSU (use this command in a lab).

Interface—Verifying TCP/IP Configurations

```
! Verifying IP Connectivity
lab-r1# ping {ip address}
lab-r1# traceroute {ip address}
lab-r1# telnet {ip address}
```

!Verify Interface Configuration

lab-r1# **sh int fa0/0** ! Display info for FastEther Slot 0/Port 0

lab-r1# **sh int e0** ! Display info for Ethernet 0

lab-r1# **sh int s0** ! Display info for Serial Interface 0

"serial 0 is administratively down, *line protocol* is down"

Note "Administratively down" means the interface has been shutdown. The `no shut` command will bring it up. "Line protocol down" means there's a framing or clocking problem. Set clock rate 64000 or check the serial cable to correct this.

! Verify Interfaces with 'sh ip interface'

lab-r1# **sh ip int brief** ! Displays a quick overview whether
! up or down and the IP address

lab-r1# **sh ip protocols** ! Quick status of layers 1&2 on each int.

lab-r1# **sh controllers s 0** ! Display DCE/DTE cable information

Network Address Translation (NAT)

! NAT/PAT Configuration Example 1

! Static NAT (multiple ISP addresses)

! A 1-to-1 mapping of "inside local" addresses to "static IP" addresses

192.168.10.0 **e0**–[lab-r1]–**e1** 209.112.160.1-----{ISP}

! Map "inside local" addr 192.168.10.1 to "static IP" addr 209.112.160.1

lab-r1# **config t**

lab-r1(config)#**ip nat inside source static 192.168.10.1 209.112.160.1**

! Define inside and outside NAT interfaces

lab-r1(config)# **int e0**

lab-r1(config-if)# **ip nat inside**

lab-r1(config-if)# **ip int e1**

lab-r1(config-if)# **ip nat outside**

! END Example 1

! NAT/PAT Configuration Example 2

! Dynamic NAT (multiple ISP addresses)

! Maps "inside local" addresses to a "static IP" address pool

```
192.168.10.0 e0-[lab-r1]-e1 209.112.160.1-209.112.160.10---{ISP}
```

! Create a pool of "static" ISP assigned addresses (10 in this example)

! The name 'ISP-POOL' could be any name you want

```
lab-r1# config t
```

```
lab-r1(config)#ip nat pool ISP-POOL 209.112.160.1 209.112.160.10 netmask
255.255.255.0
```

! Create an access-list that allows "inside local" addresses outside

! access via NAT

```
lab-r1(config)# access-list 1 permit 192.168.10.0 0.0.0.255
```

! Translate addresses that match "access-list 1" to addresses

! in the pool

```
lab-r1(config)# ip nat inside source list 1 pool ISP-POOL
```

! Define inside and outside nat interfaces

```
lab-r1(config)# int e0
```

```
lab-r1(config-if)# ip nat inside
```

```
lab-r1(config-if)# int e1
```

```
lab-r1(config-if)# ip nat outside
```

! END Example 2

! NAT/PAT Configuration Example 3

! PAT - Port Address Translation- 'overloading' (1 ISP address)

! Works exactly like Dynamic NAT, but you only use 1 ISP address in

! the pool's range, and you use the "overload" statement

```
192.168.10.0 e0-[lab-r1]-e1 209.112.160.1-----{ISP}
```

! Create a "static ip" address pool with only 1 ISP address in the range

```
lab-r1(config)# ip nat pool ISP-POOL 209.112.160.1 209.112.160.1 netmask
255.255.255.0
```

! Create an access-list that allows "inside local" addresses outside via NAT

```
lab-r1# config t
```

```
lab-r1(config)# access-list 1 permit 192.168.10.0 0.0.0.255
```

! Translate addresses that match "access-list 1" to addresses in the pool

! Note the "overload" statement

```
lab-r1(config)# ip nat inside source list 1 pool ISP-POOL overload
```

! You can configure NAT without using a NAT pool

! Instead use this statement

```
lab-r1(config)# ip nat inside source list 1 int e1 overload
```

! Define Inside and Outside NAT Interfaces

```
lab-r1(config)# int e0
```

```
lab-r1(config-if)# ip nat inside
```

```
lab-r1(config-if)# int e1
```

```
lab-r1(config-if)# ip nat outside
```

! END Example 3

! Verifying NAT

! Displays Basic IP Address Translation Information

```
lab-r1# sh ip nat translation
```

```
lab-r1# sh ip nat status
```

! Shows sending address(es), the translation and the destination

! address(es)

```
lab-r1# debug ip nat
```

! Port Forwarding with NAT/PAT

! Example of SMTP, SSH and HTTP being forwarded to inside IP address 10.10.10.2

! Redirects traffic coming in on outside interface fa4

```
lab-r1(config)# ip nat inside source static tcp 10.10.10.2 25 int fa4 25
```

```
lab-r1(config)# ip nat inside source static tcp 10.10.10.2 22 int fa4 22
```

```
lab-r1(config)# ip nat inside source static tcp 10.10.10.2 80 int fa4 80
```

Password—Encryption

```
! Manually Encrypting Passwords
lab-r1(config)# service password-encryption
```

Password—Setting

```
! Set the Privileged EXEC mode password (enable password)
lab-r1# config t
lab-r1(config)# enable secret {password}
```

```
! Set Aux Port Password
lab-r1# config t
lab-r1(config)# line aux 0
lab-r1(config-line)# password {password}
lab-r1(config-line)# login
```

```
! Set Console Password
lab-r1# config t
lab-r1(config)# line console 0
lab-r1(config-line)# password {password}
lab-r1(config-line)# login
```

```
! Set VTY Password (telnet)
lab-r1# config t
lab-r1(config)# line vty 0 4
lab-r1(config-line)# password {password}
lab-r1(config-line)# login
```

PPP Configuration

```
! Configuring PPP encapsulation on a serial interface
lab-r1(config)# int s0/1
lab-r1(config-if)# encapsulation ppp
lab-r1(config-if)# ^Z
```

```
! Configuring PPP Authentication
! The other router's hostname becomes the username for
! authentication (lab-r2 in this example)
```

```

router# config t
router(config)# hostname lab-r1          ! First, set hostname on routers
lab-r1(config)# username lab-r2 password cisco
lab-r1(config)# int s0/1
lab-r1(config-if)# encapsulation ppp
lab-r1(config-if)# ppp authentication pap chap
lab-r1(config-if)# ^Z

! Verifying PPP Encapsulation
lab-r1# sh int s0/1

.
Serial 0/0 is up, line protocol is up
Encapsulation PPP                      ! Shows encapsulation on the interface
LCP Open                               ! Shows that PPP has negotiated a session

```

Note If the line protocol is “down,” you may have mismatched encapsulation (PPP on one end and HDLC on the other).

! To check for mismatched IP addresses use the 'cdp' command
! Addresses can be wrong and ppp encap will still look good but you will
! have no connectivity between the link (encapsulation is "layer 2")

```

lab-r1# sh cdp neighbor detail

.
.
Device ID:   lab-r2
Entry Address(es)
  IP Address: 192.168.110.1

```

Routing—Default Routes

! There are 3 ways to set a Default Route (gateway)

```

lab-r1# config t
lab-r1(config)# ip route 0.0.0.0 0.0.0.0 fa0/1          ! Use an interface
lab-r1(config)# ip route 0.0.0.0 0.0.0.0 192.168.20.1  ! Use IP
lab-r1(config)# ip default-network 192.168.20.1        ! Use IP

```


Routing—EIGRP

```
! Configuring EIGRP (classless - uses an autonomous system number)
lab-r1# config t
lab-r1(config)# router eigrp 10          ! All routers must have same AS
lab-r1(config-router)# network 192.168.10.0 ! Advertised network

! Verify EIGRP Protocol
lab-r1# debug ip eigrp events            ! Displays summary of routing
lab-r1# debug ip eigrp tran             ! Displays requests from neighbors
lab-r1# show ip eigrp neighbors         ! Display eigrp neighbors

! EIGRP maximum-paths- used for load balancing (4 links by default)
lab-r1(config-router)# maximum-paths {1-6}

! EIGRP maximum-hops- metric maximum-hops (100 by default)
lab-r1(config-router)# metric maximum-hops {1-255}
```

Routing—IGRP

```
! Configuring IGRP (uses Autonomous System number)
lab-r1# config t
lab-r1(config)# router igrp 10          ! All routers must use the same AS
lab-r1(config-router)# network 192.168.10.0
```

Routing—OSPF

```
! Configuring OSPF (OSPF must always have backbone- area 0)
! 10 is the Process ID, masks (wildcards) are reversed just like ACLs
! A 0 means an exact match

lab-r1# config t
lab-r1(config)# router ospf 10          ! 10 is the AS No/Process ID
lab-r1(config-router)# network 192.168.10.0 0.0.0.0 area 0

! The following mask will match the 1st two octets exactly and allow
! any in the last two octets

lab-r1(config)# network 192.168.10.1 0.0.255.255 area 0
```

! Configuring the Loopback Interface
 ! The loopback will become the RID (Router ID) for OSPF if set

```
lab-r1# config t
lab-r1(config)# int lo0
lab-r1(config-if)# ip address 192.168.10.1 255.255.255.255
lab-r1(config-if)# no shut
lab-r1(config-if)# ^z
```

```
! Verifying OSPF
lab-r1# sh ip route           ! Basic info about all running routing protocols
lab-r1# sh ip ospf           ! RID, area info, SPF stats, LSA timers
lab-r1# sh ip ospf database   ! number of links and neighbor RIDs
lab-r1# sh ip ospf interface  ! All interface-related OSPF info.
lab-r1# sh ip ospf neighbor   ! Summarizes all OSPF info.
lab-r1# sh ip ospf neighbor detail
lab-r1# sh ip protocols       ! Overview of running protocols
                                ! (ospf, rip igrp etc.)
```

Routing—RIP

```
! Configuring RIP
lab-r1# config t
lab-r1(config)# router rip
lab-r1(config-router)# network 192.168.10.0
lab-r1(config-router)# network 172.16.0.0
```

```
! Verifying RIP
lab-r1# sh ip route

C 192.168.10.0 is directly to connected fa0/0
C 192.168.20.0 is directly to connected fa0/1
R 192.168.30.0 [120/1] via 192.168.20.2 00:00:24 fa0/1
```

! RIP Ver. 2 (entered as 'classful' but will propagate subnet mask info.)

```
lab-r1# config t
lab-r1(config)# router rip
lab-r1(config-router)# network 192.168.10.0
lab-r1(config-router)# version 2
```

! Stop RIP routing updates out a specific interface (s0/1)

```
lab-r1# config t
lab-r1(config)# router rip
lab-r1(config-router)# passive-interface serial 0/1
```

```
! Verifying routing protocols (RIP/IGRP)
lab-r1# sh ip protocols          ! Display "routing" protocols (RIP IGRP)
lab-r1# sh protocols            ! Display "routed" protocols (IP,IPX etc.)
lab-r1# sh ip int brief         ! Display IP address and int. status

lab-r1# debug ip rip            ! Display real-time updates (every 30 secs)
```

Routing—Static Routes

```
! Configuring Static Routes
lab-r1# config t
lab-r1(config)# ip route {destination} {closest int. on next router}
lab-r1(config)# ip route 192.168.20.0 255.255.255.0 192.168.10.2
```



```

[lab-r1]                [lab-r2]                [lab-r3]
  |                        |                        |
192.168.10.1    192.168.10.2---192.168.20.1    192.168.20.2---192.168.30.1

```

Secure Shell (SSH)

```
! You must configure a host and domain name before configuring SSH
lab-r1# config t
lab-r1(config)# ip domain-name geekvenue.local
lab-r1(config)# crypto key generate rsa
How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
lab-r1(config)#
```

Startup-Config and Running-Config Files

```
! Viewing, Saving & Erasing Configurations
lab-r1# copy run start          ! Copy the running config (DRAM)
                                ! to startup (NVRAM)

lab-r1# sh run                 ! Shows running configuration
lab-r1# sh start               ! Shows startup configuration
lab-r1# erase start            ! Erase startup config in NVRAM
```

Telnet

Note You must create a VTY password on a router before you can telnet to that router.

```
lab-r1# telnet 192.168.10.100
lab-r1# 192.168.10.100                ! In this case Telnet is implied

! Checking Telnet Connections and Users
lab-r1# sh sessions
lab-r1# sh users

! Closing Telnet Connections
lab-r1# exit
lab-r1# disconnect
lab-r1# disconnect 2                  ! Close session number 2

! Resolving Hostnames Statically (like a Unix hosts file)
lab-r1# config t
lab-r1(config)# ip host lab-r2 192.168.2.200
lab-r1(config)# ^z
lab-r1# sh hosts
```

host	flags	age	type	address
lab-r2	(perm/ok)	0	IP	192.168.2.200

VTY ACL for Telnet and SSH

```
! Permitting Telnet (from network 192.168.10.0) on 5 vty lines
lab-r1# config t
lab-r1(config)# access-list 10 permit 192.168.10.0 0.0.0.255
lab-r1(config)# line vty 0 4
lab-r1(config-line)# access-class 10 in
```


Passwords

! Setting Catalyst 1900 user mode/enable mode passwords

```
Switch1# config t
```

```
Switch1(config)# enable password level 1      ! User pass 1-8 char
```

```
Switch1(config)# enable password level 15     ! Priv pass 1-8 char
```

! Catalyst 2950 user mode/enable mode passwords

! Configured like a router

```
2950# config t
```

```
2950(config)# line vty 0 15
```

```
2950(config-line)# password {telnet password}
```

```
2950(config-line)# login
```

```
2950# config t
```

```
2950(config)# line console 0
```

```
2950(config-line)# password {console password}
```

```
2950(config-line)# login
```

```
2950(config-line)# exit
```

! Setting the 2950 enable /enable secret password

```
2950# config t
```

```
2950(config)# enable password {password}
```

```
2950(config)# enable secret {secret password}
```

Port Security

! Setting Port Security (allows a certain machine access to the switch port based on
! MAC address)

```
2950# config t
```

```
2950(config)# int f0/1
```

```
2950(config-if)# switchport port-security mac-address {mac-addr}
```

! Prevent multiple devices from being added to a switch port

```
switch1# config t
```

```
switch1(config)# int f0/1
```

```
switch1(config-if)# switchport port-security maximum 1
```

```
switch1(config-if)# switchport port-security violation shutdown
```

Saving and Deleting Configurations

! The 2950 has a running-config and a startup-config,

! the 1900 does NOT.

! Saving and Erasing the config on a 2950 Switch

```
2950# copy running-config startup-config
```

```
2950# erase startup-config
```

! The config on a 1900 switch is automatically saved

! To delete the config on a 1900 switch, delete nvram

```
1900# delete nvram
```

VLAN—Configuration

! Configuring VLANs on a 1900 Switch

```
# config t
```

```
(config)# hostname 1900
```

```
1900(config)# vlan 2 name MarketingDept
```

```
1900(config)# vlan 3 name SalesDept
```

```
1900(config)# exit
```

! Assigning ports to VLANs on a 1900 switch (in this example, port 2

! is assigned to vlan2 & port 3 is assigned to vlan3)

```
1900(config)# int e0/2
```

```
1900(config-if)# vlan-membership static 2
```

```
1900(config)# int e0/3
```

```
1900(config-if)# vlan-membership static 3
```

! Display VLANs on a 1900 Switch

```
1900# sh vlan
```

! Configuring VLANs on a 2950 Switch

```
# config t
```

```
switch(config)# hostname 2950
```

```
2950(config)# exit
```

```
2950# vlan database
```

```
2950(config-vlan)# vlan 2 name MarketingDept
```

```
VLAN 2 added:
```

```
Name: MarketingDep
```

```
2950(config-vlan)#
```

! Display VLANs on a 2950 Switch

```
2950# sh vlan brief
```

! Assigning ports to VLANs on a 2950 Switch (in this example
! Port 2 is assigned to vlan2 and port 3 is assigned to to vlan3)

```
2950(config-if)# int fa0/2
2950(config-if)# switchport access vlan 2
2950(config-if)# int fa0/3
2950(config-if)# switchport access vlan 3
```

! Configuring Trunk Ports on a 1900 switch (1900 will only use ISL)

```
1900# config t
1900(config)# int f0/26                ! FastEthernet
1900(config)# trunk on
```

! Configuring Trunk Ports on a 2950 switch (2950 will only use dot1q)

```
2950# config t
2950(config-if)# int f0/12
2950(config-if)# switchport mode trunk
```

! Configuring Trunk Ports on a 3550 switch (can use isl or dot1q)

```
3550# config t
3550(config-if)# int f0/12
3550(config-if)# switchport mode trunk
3550(config-if)# switchport trunk encapsulation {isl or dot1q}
```

! Configuring Inter-VLAN Routing

! 2600 Router

```
2600# config t
2600(config)# int f0/0.1                ! use a sub-interface
2500(config-subif)# encapsulation isl {vlan#}
```

! -- or ---

```
2500(config-subif)# encapsulation dot1q {vlan#}
```

! Assign an IP address to the sub-interfaces for Inter-VLAN Routing

```
2500(config-subif)# ip address 192.168.10.254 255.255.255.0
```

VLAN—Inter-VLAN Routing Example

e0/1 VLAN2 MarketingDept:192.168.1.0/24

e0/2 VLAN3 SalesDept: 192.168.2.0/24

[1900 Switch]--fa0/26-----fa0/0--[2600 Router]

! Switch Configuration

! Create the vlans

```
1900# config t
1900(config)# vlan 2 name MarketingDept
1900(config)# vlan 3 name SalesDept
```

! Assign ports to vlans

```
1900(config)# int e0/1
1900(config-if)# vlan-membership static 2
1900(config-if)# int e0/2
1900(config-if)# vlan-membership static 3
```

! Setup vlan trunking

```
1900(config-if)# int fa0/26
1900(config-if)# trunk on ! Only ISL is supported on a 1900 switch
```

! Router Configuration

! Prep the Router Interface

```
2600# config t
2600(config)# int fa0/0
2600(config-if)# no ip address
2600(config-if)# no half-duplex
2600(config-if)# no shut
2600(config-if)# exit
```

! Setup sub-interfaces for VLAN Routing using ISL

```
2600(config)# int fa0/0.2          ! It's good practice to use VLAN #'s
                                   ! for sub-interface #'s
```

```
2600(config-if)# encapsulation isl 2      ! 2 is for VLAN 2
2600(config-if)# ip address 192.168.1.1 255.255.255.0
2600(config-if)# exit
2600(config)# int fa0/0.3
2600(config-if)# encapsulation isl 3      ! 3 is for VLAN 3
2600(config-if)# ip address 192.168.2.1 255.255.255.0
```

! End of VLAN Routing Example.

VLAN—VTP Domain Configuration

! VTP domain servers propagate VLANs to other switches within
! the same VTP domain name

! VTP Domain Steps

! Create a VTP Server
! Create VTP Clients
! Create VLANs on Server

! One switch will be the server, this is where you create the VLANs

! Create the server for the VTP domain

```
2950A# config t
2950A(config)# vtp domain geekvenue
2950A(config)# vtp mode server
```

Note Server mode is the default and not necessary to set unless you are changing the mode from client!

! Create the VTP clients

```
2950B# config t
2950B(config)# vtp domain geekvenue
2950B(config)# vtp mode client
```

! create VLANs on the vtp 'server' switch

```
2950A# vlan database
2950A(config-vlan)# vlan 2 name MarketingDept
      VLAN 2 added:
      Name: Sales
```

! End of VTP domains – vtp client switch should now have
! the same VLAN info as the VTP server



ACL and Firewall Names Used in This Book

To help clarify information, I have consistently used the same variable names for the Access Control Lists (ACLs) and firewall names used throughout this book. This appendix contains a list and description of those ACLs and firewall names for your reference. You may wish to use the same variable names on your production routers. Maintaining a documented standard helps aid in troubleshooting and it allows someone less familiar than yourself to step in and work on your router.

ACL Names

IPFW-ACL: The name of the ACL that is used to allow or deny incoming traffic from the Internet to the router. This ACL is used to allow access to a host, such as an email server or web server, and is used in conjunction with port forwarding by Network Address Translation (NAT).

NAT-ACL: The name of the ACL that is used to allow or deny client computers to the Internet by way of NAT. In the examples in this book, it is configured to allow all hosts access to the Internet.

NONAT-LAN-ACL: The name of the ACL that is used in conjunction with the NONAT-LAN routing policy.

VTY-ACL: The name of the ACL that is used to allow or deny client computers the ability to log in to the router using Telnet or SSH via the TTY lines of the router. In the examples in this book, it is configured to allow all hosts access to the VTY lines.

VPN-ACL: The name of the ACL that is used to allow or deny client computers through a VPN tunnel. In the examples in this book, it is configured to allow all hosts on one end of the VPN tunnel access to all hosts on the other.

CBAC Firewall Names

EMAIL-FW: The name of the Email Inspection Engine (EIE) firewall that is used throughout this book. This firewall is always applied to incoming traffic on your WAN interface.

IPFW: The name of the Cisco Context Based Access Control (CBAC) firewall that is used throughout this book. This firewall is always applied to outgoing traffic on your WAN interface.

DHCP Pool Name

MYNET: The name of the DHCP pool that is used throughout this book.

Routing Policy Names

NO-NAT: The name of the routing policy that is used to prevent packets from being processed by NAT through a VPN tunnel.

NONAT-LAN: The name of the routing policy that is used to redirect packets returning from a remote endpoint back through the VPN tunnel.

Index

■ Numbers

3DES (Triple Data Encryption Standard), 86

■ A

Access Control Lists. *See* ACLs

access-class command, 29, 50

ACLs (Access Control Lists)

 applying to VTY, 29

 Cisco router commands, 208–209

 creating, 29–35, 39–40, 64

 naming, 29–35, 64, 231

 overview, 231

 rules

 applying established, 126

 creating, 35, 64

 displaying matches, 124–125

 reordering, 125

 securing VTY with, 50

 tuning for performance, 124–126, 139

 verifying setup, 47

active mode, 117

address resolution protocol (ARP), 37

administrator (admin), 105

ADSL (Asymmetric DSL), 57

Advanced Encryption Standard (AES), 86

advertising networks, 160–161

AES (Advanced Encryption Standard), 86

American National Standards Institute
 (ANSI) terminal emulation, 5

any statements, 91

application rules, 38–39

ARP (address resolution protocol), 37

Asymmetric DSL (ADSL), 57

authentication pre-share command, 85–86,
 99, 101

AUTH-NAK statement, 73

available address space, 177

■ B

backing up

 existing IOS image file, 115–117

 IOS, 121–123, 137–138

Base 2, 145–155

Base 10, 144, 155

bastion host, 163–164, 166–167

binary system, 145–155

bits, 146

boot system flash {image-name} command,
 121

booting new IOS image file, 120–121

branch office configuration, 197

buffered logging, 68–69

■ C

cable modem configuration

 CBAC firewall, 186

 DHCP server, 186

 IPFW access list, 187

 LAN interface, 184

 NAT setup, 185

 overview, 183–184

 router passwords, 185, 188

 saving, 188

 SSH (Version 2) connections, 188

 standard setup, 184

 VTY access list, 187

 WAN interface, 184

- Catalyst Switch commands
 - hostnames, 225
 - interface configuration, 225
 - passwords, 226
 - Port Security, 226
 - saving and deleting configurations, 226–227
 - VLAN-configuration, 227–228
 - VLAN-Inter-VLAN routing example, 229
 - VLAN-VTP domain configuration, 230
- CBAC (Context Based Access Control) firewalls
 - cable modem configuration, 186
 - configuring, 167
 - DSL and PPPoE configuration, 193
 - IPSec VPN over DSL configuration, 201–202
 - names, 232
 - overview, 37
- CCNA (Cisco Certified Network Associate), 143, 207
- CDP (Cisco Discovery Protocol)
 - commands, 210–211
 - disabling
 - cable modem configuration, 184
 - DSL and PPPoE configuration, 190
 - IPSec VPN over DSL configuration, 198
 - overview, 127, 140
- CDSL (Consumer grade DSL), 57
- Challenge-Handshake Authentication Protocol (CHAP), 58, 63
- choke point, 167
- Cisco Catalyst Switch commands. *See* Catalyst Switch commands
- Cisco Certified Network Associate (CCNA), 143, 207
- Cisco Context Based Access Control firewalls. *See* CBAC firewalls
- Cisco Discovery Protocol. *See* CDP
- Cisco IOS (Internetworking Operating System)
 - ACLs, tuning, 124–126
 - backing up, 121–123, 137–138, 210
 - CDP, disabling, 127, 140
 - debugger
 - buffered logging, 68–69
 - PPP, 71–74
 - PPPoE response, 69–71
 - stopping, 74–75
 - DHCP server status, checking, 26
 - e-mail servers, safeguarding, 127–129, 140
 - local users, creating, 105–107, 136
 - logging, configuring, 133–135
 - logging host, configuring, 129–133
 - logging host, configuring for intrusion detection, 140
 - login banner, defining, 135–141
 - overview, 1, 105
 - passwords
 - protecting, 126–127, 139
 - recovering lost, 109–113, 137
 - PPPoE, troubleshooting, 67–68
 - SSH, configuring, 107–109, 136
 - upgrading
 - backing up existing image file, 115–117
 - booting new image file, 120–121
 - deleting old image file, 117–118
 - flash memory, displaying contents of, 113–115
 - installing new image file, 118–120
 - overview, 113, 138
- Class A networks, 154
- Class B networks, 154
- Class C networks, 154
- classful networks, 143
- classless routing, 179
- clear log command, 74–75
- clearing logs, 74–75
- CLI (Command Line Interface)
 - first commands, 7–9
 - help, 10
 - keyboard shortcuts, 19

- overview, 1, 6, 19
- privileged EXEC mode, 9
- setting date and time, 9
- suppressing console messages, 19–20
- undoing command effects, 20
- clock set command, 9–10, 15
- COM port, 4
- command history commands, 211
- Command Line Interface. *See* CLI
- commands
 - ACL, 208–209
 - backing up and restoring IOS, 210
 - Catalyst Switch
 - hostnames, 225
 - interface configuration, 225
 - passwords, 226
 - Port Security, 226
 - saving and deleting configurations, 226–227
 - VLAN-configuration, 227–228
 - VLAN-Inter-VLAN routing example, 229
 - CDP, 210–211
 - command history, 211
 - configuration register, 211
 - console messages, 213
 - date and time, 213
 - DHCP configuration, 213
 - DNS lookup, 213
 - frame-relay, 214
 - hostname and MOTD, 214
 - interface
 - configuration, 214–215
 - verifying TCP/IP configurations, 215–216
 - NAT, 216–218
 - overview, 207
 - password
 - encryption, 219
 - recovery, 212
 - setting, 219
 - PPP configuration, 219–220
 - routing
 - default routes, 220
 - EIGRP, 221
 - IGRP, 221
 - OSPF, 221–222
 - RIP, 222–223
 - static routes, 223
 - SSH, 223
 - startup-config and running-config files, 223
 - Telnet, 224
 - undoing effects of, 20
 - VTY ACL for Telnet and SSH, 224
- CONFACK statement, 72
- config t command, 160–161, 165–166, 169–170
- (config)# prompt, 10
- config-register 0x2102 command, 112, 137
- configuration, router
 - CLI, 19–20
 - DHCP servers, 23–26, 49
 - DMZ, 40–43
 - erasing startup configuration, 17–19, 49
 - firewalls, 37–39, 53
 - LAN interface, 20–22, 49
 - NAT, 34–36, 52
 - overview, 17
 - printing copy of, 67
 - restoring default, 44, 54
 - saving, 43, 54
 - securing interfaces, 36–37, 52
 - telnet, 26–28, 50
 - verifying setup, 44–48, 55
 - VTY, 28–29, 50
 - WAN interface
 - creating ACLs for, 39–40
 - with dynamic IP, 30–32, 51
 - with static IP, 32–34, 51

- configuration commands, for saving and deleting, 226–227
- configuration register commands, 211
- configure terminal command, 10, 16, 21
- confreg 0x2142 command, 110, 137
- CONFREQ statement, 72
- console cables, 1, 3
- console logging, disabling, 69. *See also* logging
- console messages
 - commands, 213
 - suppressing, 19–20
- console port, 1–2, 14
- Consumer grade DSL (CDSL), 57
- Context Based Access Control firewalls. *See* CBAC firewalls
- contiguous blocks, 176
- copy flash command, 44, 116, 117, 137
- copy ftp flash command, 119
- copy run flash command, 121, 138
- copy run ftp command, 123
- copy run start command, 16, 91, 111–112, 137
- copy running-configuration
 - startup-configuration command, 13, 43, 54
- copy start run command, 111, 137
- copy startup-configuration
 - running-configuration command, 111
- corporate office configuration, 198
- cost method, 157
- crypto ipsec security-association lifetime
 - seconds command, 87–88, 99, 101
- crypto ipsec transform-set command, 89
- crypto isakmp key command, 88
- crypto isakmp policy 1 command, 85
- crypto key generate rsa command, 107, 136
- crypto map command, 88–90, 93–94, 99–100, 102
- crypto maps, 88–90
- cryptographic hash function, 86
- Ctrl key shortcuts, 19

D

- Data Encryption Standard (DES), 86
- date and time
 - commands, 213
 - setting, 9
- DB9 serial port, 3
- debug ppp authentication command, 73–74
- debug ppp negotiation command, 71
- debug pppoe events command, 69, 71
- decimal system, 144, 155
- default gateway
 - assigning, 33
 - defining, 24
 - setting, 65–78
- default routes commands, 220
- default to deny concept, 39
- default to permit concept, 39
- default-router command, 23–24, 49–50
- delete flash command, 117
- deleting old IOS image files, 117–118
- demilitarized zones. *See* DMZs
- denial of service (DOS), 36
- deny ip any any statement, 41, 124–125, 134
- DES (Data Encryption Standard), 86
- description command, 21–22, 33, 49, 61
- descriptions
 - adding to LAN interface, 21
 - adding to WAN interface, 30–33, 61
- DH (Diffie-Hellman) key, 86
- DHCP (Dynamic Host Control Protocol)
 - commands
 - cable modem configuration, 186
 - DSL and PPPoE configuration, 193
 - IPSec VPN over DSL configuration, 203
 - overview, 49, 213
 - configuration, 23–26
 - pools, 232
 - requesting addresses, 63
 - WAN interface, configuring to use, 31

- DHCP-excluded address ranges, 25
- dialer pool 1 command, 62, 77
- dialer-group 1 command, 62, 77
- dialers. *See also* WAN interface
 - defined, 58
 - interface, 58, 60, 62, 70, 189, 191
 - pool, setting, 62
- dictionary attacks, 105
- Diffie, Whitfield, 87
- Diffie-Hellman (DH) key, 86
- Digital Subscriber Line. *See* DSL
- Digital Subscriber Line Access Multiplexer (DSLAM), 61
- disabling
 - CDP, 127
 - console and monitor logging, 69
 - IP redirects, 37
 - IP unreachable messages, 36
 - proxy ARP, 37
- dividing networks
 - by keeping same subnet mask, 147
 - by subnetting
 - determining how bits are used, 148
 - determining network numbers and number of hosts, 149–156
 - determining number of subnets available, 148–155
 - examples, 151–154
 - overview, 147–148
- DMZs (demilitarized zones)
 - bastion host, 164
 - configuring, 54
 - gateway router, 165–166
 - inside source rule, applying, 42–43
 - interior router, 166–167
 - IPFW-ACL, 41
 - NAT, 41–42
 - overview, 40, 163–164
 - VPNs and, 167–168
- DNS (Domain Name System), 24
- DNS lookup commands, 213
- dns-server command, 23–24, 49–50
- domain lookup, enabling, 34
- Domain Name System (DNS), 24
- domain names, defining, 24
- domain-name command, 24
- DOS (denial of service), 36
- DSL (Digital Subscriber Line)
 - commands
 - CBAC firewall, 193
 - DHCP server, 193
 - IPFW access list, 194
 - LAN interface, 190
 - NAT setup, 192
 - overview, 189–190
 - password encryption, 195
 - passwords, 192
 - PPPoE, enabling, 191
 - saving configuration, 195
 - SSH (Version 2) connection, 195
 - standard setup, 190
 - VTY access list, 194
 - WAN interface, 191
 - configuring
 - debugger, 68–75
 - default gateway, setting, 65, 78
 - ISPs, 59, 75–76
 - MSS, adjusting on LAN interface, 65–66, 78
 - overview, 57–58
 - troubleshooting, 66–68, 78
 - VPDN, 59–60, 76
 - WAN interface, 60–65, 77
- DSLAM (Digital Subscriber Line Access Multiplexer), 61
- duplex auto command, 30–33, 51
- Dynamic Host Control Protocol. *See* DHCP
- dynamic IP, 30–32

E

EIE (E-mail Inspection Engine), 105, 127, 128–129, 232

EIGRP (Enhanced Interior Gateway Routing Protocols), 157, 221

e-mail alarms, configuring, 130–131

E-mail Inspection Engine (EIE), 105, 127, 128–129, 232

e-mail servers

- bastion host, 164
- safeguarding, 127–129, 140

EMAIL-FW firewall, 128, 232

enable command, 15, 18, 21, 28, 111, 137, 139, 160–162, 169

enable mode, 9

enable secret cisco command, 16

enable secret command, 27, 50

Encapsulating Security Payload (ESP) protocol, 89

encapsulation, 220

encapsulation ppp command, 62–63, 77

encryption type, defining, 86

Enhanced Interior Gateway Routing Protocols (EIGRP), 157, 221

ENTER key, 18

erase start command, 18, 49

erasing startup configuration, 17–19, 49

Esc key shortcuts, 19

ESP (Encapsulating Security Payload) protocol, 89

established rule, 139

Ethernet

- interfaces, 6
- ports, 3

event thresholds, 129

EXEC mode, 11, 15, 19, 200

exit command, 15, 22

F

FastEthernet4 (fa4), 163

File Transfer Protocol servers. *See* FTP servers

filter packets, 83

firewalls

- applying, 129
- CBAC, 232
- creating, 37–39, 53
- DHCP pools, 232
- naming, 128
- routing policies, 232
- verifying setup, 48

flash memory

- backing up IOS to, 121–122
- displaying contents of, 113–115

frame-relay commands, 214

FTP (File Transfer Protocol) servers

- backing up
 - existing image files to, 116–117
 - IOS to, 123
- copying new image files from, 119–120
- overview, 115

G

gateway routers, configuring, 165–166

general network settings, troubleshooting, 94–103

global configuration mode, 10–13, 16

GRE (Generic Routing Encapsulation) protocol, 39, 183

group command, 86

H

hardening servers, 164

hash function, 86

hash sha command, 85–86, 99, 101

Hellman, Martin, 86–87

help feature, 10

hexadecimal system (hex), 144

hop count, 157

hostame lab-r1 command, 16

hostname command, 21

hostnames, 11

- assigning to LAN interface, 21

- commands, 214, 225
- Hyper Terminal, 3–5, 18
-
- IANA (Internet Assigned Numbers Authority), 34, 154
- ICMP (Internet control message protocol), 36
- IDS (Intrusion Detection Systems), 129–133, 140
- IDSL (Internet DSL), 57
- IGRP (Interior Gateway Routing Protocols)
 - commands, 221
- IKE Phase 1 (Internet Key Exchange negotiation phase), 82, 85–88
- image files
 - backing up existing, 115–117
 - booting new, 120–121
 - deleting old, 117–118
 - installing new, 118–120
- in statement, 29, 40
- inside address translation, 35, 64
- inside private network addresses, 165
- installing new IOS image files, 118–120
- int e1 command, 84, 89–90, 98–100
- interface commands
 - configuration, 214–215
 - verifying TCP/IP configurations, 215–216
- interface configuration mode
 - commands, 225
 - LAN, 21–22
 - WAN, 30–33, 60–62
- interface dialer 1 command, 62, 77
- interface FastEthernet4 command, 33
- interface vlan1 command, 20–21
- Interior Gateway Routing Protocols (IGRP)
 - commands, 221
- interior routers, configuring, 166–167
- Internet Assigned Numbers Authority (IANA), 34, 154
- Internet control message protocol (ICMP), 36
- Internet DSL (IDSL), 57
- Internet Key Exchange negotiation phase (IKE Phase 1), 82, 85–88
- Internet port, 8
- Internet Security Association Key Management Protocol (ISAKMP)
 - policy, 85, 103
- Internet Service Providers. *See* ISPs
- Internetworking Operating System. *See* Cisco IOS
- Intrusion Detection Systems (IDS), 129–133, 140
- IOS. *See* Cisco IOS
- ip access-group command, 40
- ip access-group IPFW-ACL in command, 40, 53–54
- ip access-list extended command, 52
- ip access-list extended NAT-ACL command, 35, 64, 77
- ip access-list standard command, 50
- ip address command, 21, 33, 49, 51
- ip address dhcp client-id command, 51
- ip address dhcp client-id FastEthernet4 command, 30–31, 51
- ip address negotiated command, 62–63, 77
- IP addresses
 - assigning to LAN interface, 21, 49
 - assigning to WAN interface, 33
 - removing assigned, 61
- ip dhcp excluded-address command, 23, 25, 49–50
- ip dhcp pool command, 23, 49–50
- ip dhcp pool MYNET command, 23
- ip domain-lookup command, 30–31, 33–34, 51
- ip domain-name geekvenue.local command, 107, 136
- ip ftp password command, 123
- ip ftp username command, 123
- ip inspect EMAIL-FW in command, 129
- ip inspect IPFW out command, 38
- ip inspect name command, 37–38, 53, 128, 140

- ip mtu 1492 command, 62, 77
 - ip name-server command, 33–34, 51
 - ip nat inside command, 35–36, 42, 52, 54, 64–65, 68, 77–78
 - ip nat inside source list NAT-ACL int dialer 1 overload command, 64, 77
 - ip nat inside source list NAT-ACL interface fa4 overload command, 35
 - ip nat inside source list overload command, 52
 - ip nat inside source static command, 42, 54
 - ip nat inside source static tcp 192.168.1.3 25 command, 42, 54
 - ip nat outside command, 35–36, 52, 64–65, 68, 78
 - ip nat statement, 84
 - IP redirects, disabling, 37
 - ip route 0.0.0.0 0.0.0.0 dialer 1 command, 65, 78
 - IP Security. *See* IPSec
 - ip summary-address rip 192.168.1.0 255.255.255.192 command, 180
 - ip tcp adjust-mss command, 66
 - IP unreachable messages, disabling, 36
 - ipconfig command, 25
 - ipconfig/all command, 25
 - IPFW firewall
 - cable modem configuration, 187
 - DSL and PPPoE configuration, 194
 - IPSec VPN over DSL configuration, 203–204
 - overview, 37, 232
 - IPFW-ACL, 41, 90–91, 97, 231
 - IPSec (IP Security)
 - commands
 - CBAC firewall, 201–202
 - DHCP server, 203
 - enable PPPoE, 199
 - IPFW access list, 203–204
 - LAN interface, 198
 - NAT setup, 200–201
 - overview, 197–198
 - password encryption, 205
 - passwords, 200
 - saving configuration, 205
 - SSH (Version 2) connection, 204
 - standard setup, 198
 - VPN cryptographic settings, 202
 - VTY access list, 204
 - WAN interface, 199
 - configuring
 - applying VPN routing policy, 84
 - corporate office VPN configuration, 100–102
 - creating VPN-friendly ACL, 83
 - defining VPN routing policy, 83–84
 - IKE Phase 1, 85–88
 - modifying IPFW-ACL, 90–91
 - overview, 81
 - preparing sites, 81–82
 - troubleshooting, 102–103
 - verifying VPN connection, 92–93
 - Phase 2 data encryption, 88–90
 - ISAKMP (Internet Security Association Key Management Protocol) policy, 85, 103
 - ISPs (Internet Service Providers)
 - collecting information from, 59, 76
 - requesting VPDN tunnel to, 60
 - trouble with, 75–76
- K**
- key exchange policy, 85
 - key size, defining, 86–87
 - key type, defining, 86
 - keyboard shortcuts, 19
 - Kiwi, 130
- L**
- LAN (Local Area Network) interface
 - adjusting MSS on, 65–66, 78
 - commands

- cable modem configuration, 184
- DSL and PPPoE configuration, 190
- IPSec VPN over DSL configuration, 198
- configuring, 20–22
- configuring NAT to forward traffic to LAN host, 41–42
- IP address, assigning, 49
- LAN Ethernet ports, 2, 8, 14
- LCP (Link Control Protocol), 71
- LCP: State is Open statement, 71
- lease time, defining, 24–25
- line vty 0 4 command, 27–29, 50, 106–108, 136
- Link Control Protocol (LCP), 71
- Local Area Network interface. *See* LAN interface
- local logging facility, 129
- local users, creating, 105–107, 136
- log level, changing, 134
- logging
 - clearing logs, 74–75
 - configuring, 133–135
 - disabling, 69
- logging 192.168.1.2 command, 133, 135
- logging buffered 4096 debugging command, 68–69, 74–75
- logging console command, 20
- logging host
 - configuring, 129–133, 140
 - defining, 135
- logging trap command, 133–134
- login, setting to local, 107
- login banner command, 135
- login banner, defining, 135–141
- login command, 27
- login local command, 106–107, 136

M

- management port, 1–2, 14
- match address VPN-ACL command, 89–90, 99, 102

- max-data parameter, 128
- Maximum Segment Size (MSS), 58, 65–66, 78
- Maximum Transmission Unit. *See* MTU
- Message Digest Algorithm (md5), 86
- message of the day (MOTD) commands, 214
- message threshold, setting, 131–133
- monitor logging, disabling, 69. *See also* logging
- MOTD (message of the day) commands, 214
- MSS (Maximum Segment Size), 58, 65–66, 78
- MTU (Maximum Transmission Unit)
 - checking, 66–67
 - overview, 58
 - setting size, 62
- MYNET DHCP pool, 232

N

- NAS (Network Access Server), 60
- NAT (Network Address Translation)
 - ACL rules, creating, 35
 - applying, 36
 - commands
 - cable modem configuration, 185
 - DSL and PPPoE configuration, 192
 - IPSec VPN over DSL configuration, 200–201
 - overview, 216–218
 - configuring
 - to forward traffic to LAN host, 41–42
 - on gateway router, 165–166
 - overview, 34–36, 52
 - on virtual WAN interface, 64–65, 77
 - extended ACLs, 35
 - inside address translation, configuring, 35
 - overview, 34
 - verifying setup, 46–47
- NAT-ACL, 97, 231
- negotiated IP addresses, 59
- netbios-name-server command, 23–24, 50

- network 192.168.2.0 command, 159–161, 169–170
- Network Access Server (NAS), 60
- Network Address Translation. *See* NAT
- network addresses, defining, 23
- network command, 17, 21, 23, 49–50, 160, 169
- network number, 146–147
- Network Operations Center (NOC), 75
- network segments, 143
- no auto-summary command, 180
- no boot system command, 121
- no cdp run command, 127, 140
- no debug pppoe negotiation command, 72
- no hostname command, 20
- no ip access-list extended IPFW-ACL command, 41
- no ip address command, 61
- no ip ftp passive command, 117, 123
- no ip nat inside source list NAT-ACL int e1 overload command, 84
- no ip proxy-arp command, 37
- no ip redirects command, 36–37, 52
- no ip unreachable command, 36, 52
- no logging console command, 20, 69
- no logging monitor command, 69
- no shut command, 216
- no shutdown command, 21, 31, 33, 61
- NOC (Network Operations Center), 75
- NO-NAT routing policy, 232
- NONAT-LAN routing policy, 84, 232
- NONAT-LAN-ACL, 97, 231

O

- octets, 146–147
- open LCP link state, 71
- OSI (Open Systems Interconnection), 127
- OSPF (Open Shortest Path First), 157, 221–222
- outbound traffic, 38

P

- packet switching, 157
- PADI (PPPoE Active Discovery Initiation), 69
- PAP (Password Authentication Protocol), 58, 63
- passive interface, configuring, 161
- passive mode, 117
- passive-interface command, 161, 169
- passive-interface fa4 command, 161, 170
- Password Authentication Protocol (PAP), 58, 63
- password command, 27
- passwords
 - Catalyst Switch commands, 226
 - checking, 66–67
 - commands
 - cable modem configuration, 185
 - DSL and PPPoE configuration, 192
 - IPSec VPN over DSL configuration, 200
 - creating, 106
 - encrypting
 - cable modem configuration, 188
 - DSL and PPPoE configuration, 195
 - IPSec VPN over DSL configuration, 205
 - encryption commands, 219
 - privileged EXEC mode, 11–12, 27
 - protecting, 126–127, 139
 - recovering lost, 109–113, 137, 212
 - setting commands, 219
 - VTY login, 27–28
- path determination, 157
- Perfect Forward Security (PFS) group, 90
- perimeter LANs, 163
- permit esp any any command, 91
- permit ip 192.168.1.0 0.0.0.255 any command, 35, 64
- permit ip statement, 52
- permit tcp any host eq smtp command, 41, 54

- permit tcp any host established command, 125, 126
 - permit udp any any eq isakmp command, 91, 98, 100, 102
 - permit udp any eq isakmp any command, 91, 98, 100, 102
 - PFS (Perfect Forward Security) group, 90
 - physical WAN interface. *See* WAN interface
 - ping utility, 31, 40, 92, 163
 - Point-to-Point Protocol over Ethernet. *See* PPPoE
 - Point-to-Point Protocol (PPP), 57, 62–63, 71–74, 219–220
 - Point-to-Point Tunneling Protocol (PPTP), 39
 - pool name, defining, 23
 - port forwarding, configuring, 166
 - port security commands, 226
 - ports
 - console port, 1–2
 - LAN Ethernet, 2, 8, 14
 - overview, 1
 - WAN Ethernet, 2, 8, 14
 - power up process, 5–6
 - PPP (Point-to-Point Protocol), 57, 62–63, 71–74, 219–220
 - ppp authentication chap pap callin command, 62–63, 77
 - PPPoE (Point-to-Point Protocol over Ethernet)
 - commands
 - DSL and PPPoE configuration, 191
 - IPSec VPN over DSL configuration, 199
 - debugger, 68, 75
 - default gateway, setting, 65, 78
 - ISPs, 59, 75–76
 - MSS, adjusting on LAN interface, 65–66, 78
 - overview, 57–58
 - responses, 69–71
 - troubleshooting, 66–68, 78
 - VPDN, 59–60, 76
 - WAN interface, 60–65, 77
 - PPPoE Active Discovery Initiation (PADI), 69
 - pppoe enable group global command, 60–61, 77
 - pppoe statement, 61
 - pppoe-client dial-pool-number 1 command, 60–61, 77
 - PPTP (Point-to-Point Tunneling Protocol), 39
 - presared keys, 86–88
 - printing router configuration, 67
 - private internal networks, 159
 - private IP addresses, 34
 - private keys, 87
 - privilege level 1, 106
 - privilege level 15, 106, 136
 - privileged EXEC mode, 9, 11–12, 15, 27, 106
 - protocol pppoe command, 59–60, 76
 - proxy ARP, disabling, 37
 - Public Key/Private Key encryption concept, 87
 - public keys, 87
- ## R
- recovering lost passwords
 - bypassing IOS, 110
 - commands, 212
 - configuration register
 - modifying, 110
 - resetting, 112–113
 - copying configuration, 111–112
 - overview, 109
 - process, 109
 - resetting password, 111–112
 - reload command, 18, 112, 118, 120–121
 - request-dialin command, 59–60, 76
 - restoring default router configuration, 44–54
 - RIP (Routing Information Protocol)
 - commands, 222–223
 - configuring
 - on neighbor router, 160–161, 169–170
 - on router, 158–160, 169
 - overview, 157, 158
 - verifying, 161–163

ROM Monitor Mode, 109
route summarization, 178–181
route-map NO-NAT permit 10 command, 83
router rip command, 160–161, 169–170
routers, 1, 16
 CLI, 6–10
 connecting to, 2–6
 displaying and saving configuration, 16
 EXEC mode commands, 15
 global configuration mode, 10–13, 16
 overview, 1
 ports, 1–2, 14
routing by rumor method, 158
Routing Information Protocol. *See* RIP
routing policies, 83, 232
routing protocols, 157–158
routing table, 158
routing-update, 163
RSA keys, generating, 107–108
running-config file commands, 223

S

saving router configuration, 43, 54
SDM (Security Device Manager), 2
Secure Hash Algorithm (SHA), 86
Secure Shell. *See* SSH
securing interfaces, 36–37, 52
Security Device Manager (SDM), 2
security-hardened servers, 164
serial port, 1
service password-encryption command, 126–127, 139, 188, 195, 205
service timestamps command, 133–134
set peer 24.237.8.112 command, 89–90, 99
set pfs group2 command, 90
set transform-set SET1 command, 89–90, 99, 102
sh flash command, 54, 122
sh ip access-list IPFW-ACL command, 124, 139

sh run command, 12–13, 16, 94
sh start command, 13, 16
sh webflash command, 54
SHA (Secure Hash Algorithm), 86
show access-list command, 47, 55
show clock command, 9, 15
show command, 55, 92, 125
show commands, 44
show crypto ipsec sa command, 92
show crypto isakmp sa command, 92
show flash command, 7, 15, 44, 113, 121
show interface command, 44–45, 55
show ip access-list command, 139
show ip dhcp binding command, 26
show ip inspect interfaces command, 48, 55
show ip interface brief command, 8, 15, 44, 55, 67
show ip interface command, 44–45, 55
show ip nat statistics command, 46, 55
show ip nat translations command, 47, 55
show ip protocols command, 161–162, 170
show ip route command, 161–163, 170–171
show log command, 69–71, 73
show run command, 12, 47, 126
show running-configuration command, 12, 22, 29, 55, 67
show startup-configuration command, 12–13
show version command, 7, 15
show webflash command, 44
shutdown command, 22
Simple Mail Transport Protocol (SMTP), 40
SmartNet, 113
SMTP (Simple Mail Transport Protocol), 40
speed auto command, 30–33, 51
SSH (Secure Shell)
 commands
 cable modem configuration, 188
 DSL and PPPoE configuration, 195
 IPSec VPN over DSL configuration, 204

- overview, 223
- configuring, 107–109, 136
- standard setup
 - cable modem configuration, 184
 - DSL and PPPoE configuration, 190
 - IPSec VPN over DSL configuration, 198
- startup configuration, erasing, 17–19, 49
- startup-config file commands, 223
- stateful packet inspection firewall, 37
- static IP, 32–34, 184
- static routes commands, 223
- stopping
 - all debugging, 74
 - buffered logging, 75
- subnet masks, 146–155, 177, 179
- subnetting
 - Base 10, 144, 155
 - Base 2, 145–155
 - dividing networks, 147–156
 - overview, 143–144
 - subnet masks, 146–155
- supernetting, 178–181
- syslog server, 68
- System Logger, 129

T

TAB key, 19

TAC (Technical Assistance Center), 113

TCP/IP networking, 154

Technical Assistance Center (TAC), 113

telnet

- commands, 224
- configuring, 26–28, 50

terminal emulation program, 3

TFTP (Trivial File Transfer Protocol) servers

- backing up
 - existing image files to, 116
 - IOS to, 122
- copying new image files from, 118–119
- overview, 115

timestamping logs, 134

traceroute requests, 39–40

transform set, 88–90

transport input ssh statement, 108

transport input telnet ssh statement, 108

trap levels, viewing, 133–134

Triple Data Encryption Standard (3DES), 86

Trivial File Transfer Protocol servers. *See* TFTP servers

troubleshooting

- DSL using PPPoE, 66–68, 78
- general network settings, 94–103
- IKE Phase 1 settings, 94–103
- IPSec Phase 2 settings, 95–103
- overview, 94

U

undebug all command, 74

unprivileged access, 106

upgrading IOS

- backing up existing image file, 115–117
- booting new image file, 120–121
- deleting old image file, 117–118
- flash memory, displaying contents of, 113–115
- installing new image file, 118–120
- overview, 113, 138

username command, 105–108, 117, 123, 136

usernames, checking, 66–67

users, creating, 106

V

Variable Length Subnet Mask networking.
See VLSM networking

verifying

- RIP, 161–163, 170–171
- router setup, 44–48, 55
- VPN connection, 92–93

version 2 command, 160–162, 169–170

Virtual Private Dialup Networking (VPDN), 59–60, 76

Virtual Private Networks. *See* VPNs

Virtual Terminal line. *See* VTY

virtual WAN interface. *See* WAN interface

VLAN, Catalyst Switch commands, 227–230

VLSM (Variable Length Subnet Mask)

- networking
- overview, 173–175
- planning, 175–181
- route summarization, 178–181

VPDN (Virtual Private Dialup Networking), 59–60, 76

vpdn enable command, 59, 76

vpdn-group command, 59

vpdn-group PPPOE command, 59–60, 76

VPN-ACL, 89, 90, 97, 224, 231

VPN-friendly ACL, 83

VPNs (Virtual Private Networks)

- branch office configuration, 98–100
- corporate office configuration, 100–102
- creating VPN-friendly ACL, 83
- DMZs and, 167–168
- IKE Phase 1, 85–88
- IPSec Phase 2, 88–90
- IPSec VPN over DSL configuration, 202
- modifying IPFW-ACL, 90–91
- overview, 81
- preparing sites, 81–82
- routing policy, 83–84
 - applying to NAT, 84
 - defining, 83–84
- setting peer, 90
- setting up, 95–97
- troubleshooting, 94–103

verifying connection, 92–93

VTY (Virtual Terminal line)

- cable modem configuration, 187
- DSL and PPPoE configuration, 194
- IPSec VPN over DSL configuration, 204
- password, 224
- setting transport input type, 108
- setting up and securing, 27–29, 50

VTY-ACL, 231

■ W

WAN (Wide Area Network) interface

- cable modem configuration, 184
- configuring
 - with dynamic IP, 30–32, 51
 - with static IP, 32–34, 51
- creating ACLs for, 39–40
- physical
 - configuring, 60–61, 77
 - DSL and PPPoE configuration, 191
 - IPSec VPN over DSL configuration, 199
- virtual
 - configuring, 61–63, 77
 - configuring NAT on, 64–65, 77
 - DSL and PPPoE configuration, 191
 - IPSec VPN over DSL configuration, 199

WAN Ethernet port, 2, 8, 14

web flash memory, 5

Wide Area Network interface. *See* WAN interface

WINS (Windows Internet Name Service)

- server, 24

workstations, testing DHCP servers using, 25–26