CISCO

Official Cert Guide

Learn, prepare, and practice for exam success

- Master **CCNP SECURE 642-637** exam topics
- Assess your knowledge with **chapter-opening quizzes**
- Review key concepts with **exam preparation tasks**
- Practice with **realistic exam questions** on the CD-ROM

CCNP Security SECURE 642-637

SEAN WILKINS
FRANKLIN H. SMITH III

ciscopress.com

CCNP Security SECURE 642-637

Official Cert Guide

Sean Wilkins

Franklin H. Smith III

Cisco Press

800 East 96th Street

Indianapolis, IN 46240

CCNP Security SECURE 642-637 Official Cert Guide

Sean Wilkins, Franklin H. Smith III

Published by:
Cisco Press
800 East 96th Street
Indianapolis, IN 46240 USA

Printed in the United States of America

Second Printing: October 2013

The Library of Congress Cataloging-in-Publication Data is on file.

ISBN-13: 978-1-58714-280-2

ISBN-10: 1-58714-280-5

Warning and Disclaimer

This book is designed to provide information for the Cisco CCNP Security 642-637 SECURE exam. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an “as is” basis. The authors, Cisco Press, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc., cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Corporate and Government Sales

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact: U.S. Corporate and Government Sales 1-800-382-3419 corpsales@pearsontechgroup.com

For sales outside the United States, please contact: International Sales international@pearsoned.com

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members from the professional technical community.

Readers' feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this book, or otherwise alter it to better suit your needs, you can contact us through email at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

Publisher: Paul Boger

Associate Publisher: Dave Dusthimer

Executive Editor: Brett Bartow

Managing Editor: Sandra Schroeder

Senior Development Editor: Christopher Cleveland

Project Editor: Mandie Frank

Designer: Gary Adair

Cisco Representative: Erik Ullanderson

Cisco Press Program Manager: Anand Sundaram

Technical Editors: Sean Connelly and Robert Woods

Copy Editor: John Edwards

Editorial Assistant: Vanessa Evans

Proofreader: Sheri Cain

Composition: Mark Shirar

Indexer: Tim Wright

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems, Inc.
168 Robinson Road
#28-01 Capital Tower
Singapore 068912
www.cisco.com
Tel: +65 6317 7777
Fax: +65 6317 7799

Europe Headquarters
Cisco Systems International BV
Haarlerbergpark
Haarlerbergweg 13-19
1101 CH Amsterdam
The Netherlands
www-europe.cisco.com
Tel: +31 0 800 020 0791
Fax: +31 0 20 357 1100

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

About the Authors

Sean Wilkins is an accomplished networking consultant for SR-W Consulting (www.sr-wconsulting.com) and has been in the field of IT since the mid 1990s working with companies like Cisco, Lucent, Verizon, and AT&T, as well as several other private companies. Sean currently holds certifications with Cisco (CCNP/CCDP), Microsoft (MCSE), and CompTIA (A+ and Network+). He also has a Master of Science degree in information technology with a focus in network architecture and design, a Master of Science in organizational management, a Master's Certificate in network security, a Bachelor of Science degree in computer networking, and an Associate of Applied Science degree in computer information systems. In addition to working as a consultant, Sean spends a lot of his time as a technical writer and editor for various companies.

Franklin H. Smith III (Trey) is a senior network security architect with more than 15 years of experience in designing, deploying, and securing large enterprise and service provider networks. His background includes architect-level delivery for many enterprise, data center, and SMB networks. He holds a Bachelor of Business Administration degree in management information systems. Trey's certifications include CCSP, CCNP, CCDP, Microsoft (MCSE), and ISC2 (CISSP). His current focus is on strategic and tactical efforts related to Payment Card Industry (PCI) Data Security Standard (DSS) compliance for a Fortune 50 company.

About the Technical Reviewers

Sean Connelly, CCIE #17085 (R/S & Security), is a senior network design engineer for TASC, based in Washington, D.C. He has worked for two federal agencies over the last decade. Recent projects have included architecting a global 802.1X solution and the design and implementation of a large data center, along with active involvement in other federal cyber security initiatives. Before joining TASC, Sean was director of IT Services at ADCom, which included the design of many global WAN solutions. Aside from the two CCIEs, Sean holds a CISSP and a bachelor's degree in business administration, with a total of 14 years of IT experience.

Robert Woods is a seasoned information assurance professional with 21 years of experience in information and network security, compliance, and leadership. Recently most of his efforts have focused on securing enterprise networks for financial services organizations to satisfy regulatory and industry requirements. Specific areas of focus have included strategic and tactical efforts for the Payment Card Industry Data Security Standards (PCI DSS). Robert worked as a qualified security assessor (QSA) in a strategic role at the world's largest retailer and as a senior-level technical advisor at the largest automobile insurer in the United States. Professional certifications include CISSP, MCSE, and GSEC Gold. Robert holds a bachelor's degree in electronic systems technology (EST) from Southern Illinois University and a Master of Science degree in information assurance (MSIA) from Norwich University.

Dedications

I dedicate this book to my girls (Stacy, Anij, and Saliah), one of which was born during the development of this book. Without all of you, none of this would be possible.

—Sean Wilkins

To my wife and daughters (Jackie, Olivia, and Victoria): It is from you that I draw my strength, for you that I have the ambition to try to “do better,” and to you that I dedicate this book. Thank you for the support and understanding throughout this project.

—Franklin H. Smith III

Acknowledgments

We want to take this opportunity to thank all the people who took our words and transformed them into a readable, organized, and formatted text for all of you to read and learn from. Without their efforts, this book would not have been possible. Because we only work directly with a few of these people, there are many people we will be unable to directly thank. For these people, we take this opportunity to thank you for your work in developing this project and look forward to working with you in the future.

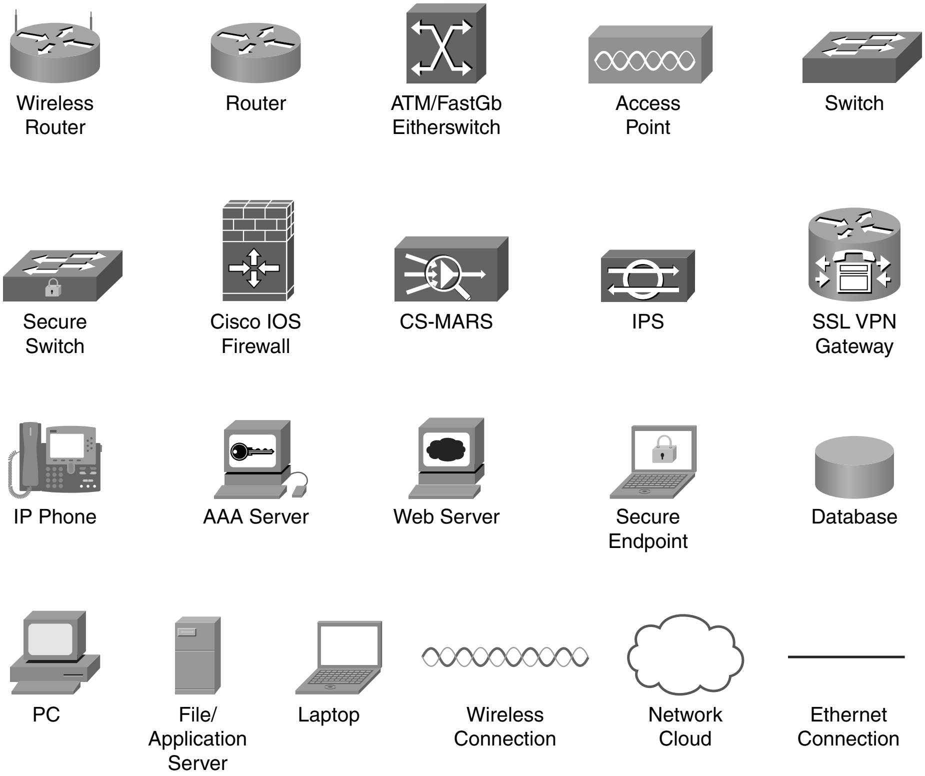
Contents at a Glance

Elements Available on CD

Contents

Elements Available on CD

Icons Used in This Book



Introduction

This book helps you prepare for the Cisco SECURE certification exam. The SECURE exam is one in a series of exams required for the Cisco Certified Network Professional - Security (CCNP - Security) certification. This exam focuses on the application of security principles with regard to Cisco IOS routers, switches, and Virtual Private Network (VPN) devices.

Who Should Read This Book

Network security is a complex business. It is important that you have extensive experience in and an in-depth understanding of computer networking before you can begin to apply security principles. The Cisco SECURE program was developed to introduce the security products associated with or integrated into Cisco IOS Software, explain how each product is applied, and explain how it can increase the security of your network. The SECURE program is for network administrators, network security administrators, network architects, and experienced networking professionals who are interested in applying security principles to their networks.

How to Use This Book

This book consists of 22 chapters. Each chapter tends to build upon the chapter that precedes it. The chapters that cover specific commands and configurations include case studies or practice configurations.

The chapters cover the following topics:

- **Chapter 1, "Network Security Fundamentals":** This chapter reviews the basic network security concepts and elements along with a review of the Cisco SAFE approach. It is this core of understanding that provides a good base for the other chapters.
- **Chapter 2, "Network Security Threats":** This chapter reviews the different methods used to exploit a network and the elements on it. With a better understanding of the methods used, network security personnel are better equipped to face these security challenges as they are found.
- **Chapter 3, "Network Foundation Protection (NFP) Overview":** NFP details a layered approach to protecting Cisco IOS Software–based devices. Attacks against the control, data, and management planes and the appropriate mitigation techniques are covered.
- **Chapter 4, "Configuring and Implementing Switched Data Plane Security Solutions":** This chapter reviews the different types of attacks that are focused at the data plane of the switches in the network. It then goes on to review the technologies that can be used to mitigate them and shows how to configure them to best protect the switched data plane.

- **Chapter 5, "802.1X and Cisco Identity-Based Networking Services (IBNS)":** This chapter reviews IEEE 802.1X and the Cisco IBNS framework that are both used to protect the network from unauthorized users. It goes into the basics of 802.1X, including the various Extensible Authentication Protocol (EAP) methods that can be used as well as the different IBNS features that can be used to secure the network.

- **Chapter 6, "Implementing and Configuring Basic 802.1X":** This chapter describes how to configure basic 802.1X authentication on a Cisco IOS Software–based device to prevent unauthorized clients (supplicants) from gaining access to the network.

- **Chapter 7, "Implementing and Configuring Advanced 802.1X":** This chapter describes how to configure advanced 802.1X authentication features on a Cisco IOS Software–based device to prevent unauthorized clients (supplicants) from gaining access to the network.

- **Chapter 8, "Implementing and Configuring Cisco IOS Routed Data Plane Security":** This chapter reviews the different types of attack that are focused at the data plane of the routers (or Layer 3 switches) in the network. It then reviews the different features that can be used to mitigate these threats and how to configure them.

- **Chapter 9, "Implementing and Configuring Cisco IOS Control Plane Security":** This chapter reviews the different types of attack that are focused at the control plane of the devices in the network. It then reviews the different features that can be used to mitigate these threats and how to configure them.

- **Chapter 10, "Implementing and Configuring Cisco IOS Management Plane Security":** This chapter reviews the different types of attack that are focused at the management plane of the devices in the network. It then reviews the different features that can be used to mitigate these threats and how to configure them.

- **Chapter 11, "Implementing and Configuring Network Address Translation (NAT) ":** This chapter reviews the Network Address Translation (NAT) feature and how it can be used in various ways on the network. NAT is an important feature that is used by almost everyone on a daily basis; a through understanding of it is vital now that the majority of the IPv4 address space has been depleted.

- **Chapter 12, "Implementing and Configuring Zone-Based Policy Firewalls":** This chapter reviews the Zone-Based Policy Firewall (ZBPFW) feature and how it can be used to secure the different parts of the network. In the modern network environment, there are a number of threats that exist that are focused on the network and the devices on it. The ZBPFW feature has a number of different capabilities that can be used to mitigate these threats and keep the network and the devices on it secure.

- **Chapter 13, "Implementing and Configuring IOS Intrusion Prevention System (IPS)":** The Cisco IOS Intrusion Prevention System (IPS) feature set is the evolution of the Cisco IOS Intrusion Detection System (IDS). Cisco IPS products go beyond the IDS signature matching by incorporating features such as stateful pattern recognition, protocol analysis, traffic anomaly detection, and protocol anomaly detection. This chapter discusses the security features of the Cisco IOS IPS.

- **Chapter 14, "Introduction to Cisco IOS Site-to-Site Security Solutions":** This chapter introduces site-to-site VPN technologies and an overview of the many topologies and technologies that are possible with IPsec VPNs.

- **Chapter 15, "Deploying VTI-Based Site-to-Site IPsec VPNs":** This chapter covers deployment of static and dynamic point-to-point VTI tunnels using Cisco IOS Software. IP Security (IPsec) Virtual Tunnel Interfaces (VTI) greatly simplify the configuration process that is required to create site-to-site VPN tunnels.

- **Chapter 16, "Deploying Scalable Authentication in Site-to-Site IPsec VPNs":** Cisco IOS devices are designed with a feature called CA interoperability support, which allows them to interact with a certificate authority (CA) when deploying IPsec. This functionality allows a scalable and manageable enterprise VPN solution.

- **Chapter 17, "Deploying DMVPNs":** Dynamic Multipoint Virtual Private Networks (DMVPN) are a feature of Cisco IOS Software that makes the deployment of large hub-and-spoke, partial mesh, and full mesh VPN topologies much easier. This chapter covers implementing DMVPN on Cisco IOS Software–based devices.

- **Chapter 18, "Deploying High Availability in Tunnel-Based IPsec VPNs":** This chapter describes the mechanisms that can be put in place to provide a high-availability solution that will protect an organization from outages.

- **Chapter 19, "Deploying GET VPNs":** This chapter covers the deployment of the Cisco Group Encrypted Transport Virtual Private Network (GET VPN) technology. It provides a solution that allows easy deployment of a complex, redundant, fully meshed VPN network.

- **Chapter 20, "Deploying Remote Access Solutions Using SSL VPNs":** Remote access VPN technologies allow mobile workers to access internal resources over untrusted networks. This chapter will discuss a comparison of remote access VPN technologies and then cover configuring, verifying, and troubleshooting a basic client-based and clientless SSL VPN solution on a Cisco ISR.

- **Chapter 21, " Deploying Remote Access Solutions Using EZVPNs":** Cisco Easy VPN is a client/server application that allows VPN security parameters to be "pushed out" to the remote locations that connect using a growing array of Cisco products.

- **Chapter 22, "Final Preparation":** This short chapter lists the exam preparation tools useful at this point in the study process and provides a suggested study plan now that you have completed all the earlier chapters in this book.

- **Appendix A, "Answers to Chapter DIKTA Quizzes and Fill in the Blanks Questions":** This appendix provides the answers to the Do I Know This Already? quizzes that you will find at the beginning of each chapter as well as the answers to the Fill in the Blanks questions that you will find at the end of each chapter.

- **Appendix B, "CCNP Security 642-637 SECURE Exam Updates, Version 1.0":** This appendix provides you with updated information if Cisco makes minor modifications to the exam upon which this book is based. When Cisco releases an entirely new exam, the changes are usually too extensive to provide in a simple update appendix. In those cases, you need to consult the new edition of the book for the updated content. This additional content about the exam will be posted as a PDF document on this book's companion website (www.ciscopress.com/title/9781587142802).

- **Appendix C, "Memory Tables" (CD only):** This appendix, which you will find in PDF form on the CD accompanying this book, provides a series of tables that highlight some of the key topics in each chapter. Each table provides some cues and clues that will enable you to complete the table and test your knowledge on the table topics.
- **Appendix D, "Memory Table Answers" (CD only):** This appendix, which you will find in PDF form on the CD accompanying this book, provides the completed memory tables from Appendix C so that you can check your answers. In addition, you can use this appendix as a standalone study tool to help you prepare for the exam.
- **Glossary:** This glossary defines the key terms that appear at the end of each chapter, for which you should be able to provide definitions on your own in preparation for the exam.

Each chapter follows the same format and incorporates the following tools to assist you by assessing your current knowledge and emphasizing specific areas of interest within the chapter:

- **Do I Know This Already? quiz:** Each chapter begins with a quiz to help you assess your current knowledge of the subject. The quiz is divided into specific areas of emphasis that enable you to best determine where to focus your efforts when working through the chapter.
- **Foundation Topics:** The foundation topics are the core sections of each chapter. They focus on the specific protocols, concepts, or skills that you must master to successfully prepare for the examination.
- **Exam Preparation:** Near the end of each chapter, the Exam Preparation section highlights the key topics from the chapter and the pages where you can find them for quick review. This section also refers you to the Memory Tables appendixes and provides a list of key terms that you should be able to define in preparation for the exam. It is unlikely that you will be able to successfully complete the certification exam by just studying the key topics, memory tables, and key terms, although they are a good tool for last-minute preparation just before taking the exam.
- **Fill in the Blanks:** Each chapter ends with a series of review questions to test your understanding of the material covered. These questions are a great way to ensure that you not only understand the material, but that you also exercise your ability to recall facts.
- **CD-ROM-based practice exam:** This book includes a CD-ROM containing a free, complete practice exam. It is recommended that you continue to test your knowledge and test-taking skills by using this exam. You will find that your test-taking skills will improve by continued exposure to the test format. Remember that the potential range of exam questions is limitless. Therefore, your goal should not be to "know" every possible answer but to have a sufficient understanding of the subject matter so that you can figure out the correct answer with the information provided.

Premium Edition

In addition to the free practice exam provided on the CD-ROM, you can purchase additional exams with expanded functionality directly from Pearson IT Certification. The Premium Edition of this title contains an additional two full practice exams as well as an eBook (in both PDF and ePub format). In addition, the Premium Edition title also has remediation for each question to the specific part of the eBook that relates to that question.

Because you have purchased the print version of this title, you can purchase the Premium Edition at a deep discount. A coupon code in the CD sleeve contains a one-time-use code as well as instructions for where you can purchase the Premium Edition.

To view the Premium Edition product page, go to
http://www.pearsonitcertification.com/store/product.aspx?isbn=1587142805.

Certification Exam and This Preparation Guide

The questions for each certification exam are a closely guarded secret. The truth is that if you had the questions and could only pass the exam, you would be in for quite an embarrassment as soon as you arrived at your first job that required these skills. The point is to know the material, not just to successfully pass the exam. We do know which topics you must know to successfully complete this exam because they are published by Cisco. Coincidentally, these are the same topics required for you to be proficient when configuring Cisco security devices. It is also important to understand that this book is a "static" reference, whereas the exam topics are dynamic. Cisco can and does often change the topics covered on certification exams. This exam guide should not be your only reference when preparing for the certification exam. You can find a wealth of information available at Cisco.com that covers each topic in painful detail. The goal of this book is to prepare you as well as possible for the SECURE exam. Some of this is completed by breaking a 600-page (average) implementation guide into 30-page chapters that are easier to digest. If you think that you need more detailed information on a specific topic, feel free to surf. Table I-1 lists each exam topic along with a reference to the chapter that covers the topic.

Table I-1 *SECURE Exam Topics and Chapter References*

Exam Topic	Chapter Where Topic Is Covered
Preproduction Design	
Choose Cisco IOS technologies to implement HLD (High Level Design)	Chapters 3, 4, 8, 9, 10, 11, 12
Choose Cisco products to implement HLD	Chapters 3, 4, 8, 9, 10, 11, 12
Choose Cisco IOS features to implement HLD 2	Chapters 3, 4, 5, 8, 9, 10, 11, 12

Table I-1 *SECURE Exam Topics and Chapter References*

Exam Topic	Chapter Where Topic Is Covered
Integrate Cisco network security solutions with other security technologies	Chapters 1, 3, 4, 5, 8, 9, 10, 11, 12
Create and test initial Cisco IOS configurations for new devices/services	Chapters 4, 5, 8, 9, 10, 11, 12
Configure and verify ASA VPN feature configurations	Chapters 20, 21
Complex Operations Support	
Optimize Cisco IOS security infrastructure device performance	Chapters 3, 4, 5, 8, 9, 10, 11, 12
Create complex network security rules to meet the security policy requirements	Chapters 1, 2
Optimize security functions, rules, and configuration	Chapters 3, 4, 5, 8, 9, 10, 11, 12
Configure and verify classic IOS firewall and NAT to dynamically mitigate identified threats to the network	Chapters 11, 12
Configure and verify IOS Zone-Based Firewalls including advanced application inspections and URL filtering	Chapter 12
Configure and verify the IPS features to identify threats and dynamically block them from entering the network	Chapters 2, 13
Maintain, update, and tune IPS signatures	Chapters 2, 13
Configure and verify IOS VPN features	Chapters 14–19
Configure and verify Layer 2 and Layer 3 security features	Chapters 4, 5, 8, 9, 10, 11, 12
Advanced Troubleshooting	
Advanced Cisco IOS security software configuration fault finding and repairing	Chapters 4, 8, 9, 10, 11, 12
Advanced Cisco routers and switches hardware fault finding and repairing	Chapters 4, 8, 9, 10, 11, 12

You will notice that not all the chapters map to a specific exam topic. This is because of the selection of evaluation topics for each version of the certification exam. Our goal is to provide the most comprehensive coverage to ensure that you are well prepared for the exam. To do this, we cover all the topics that have been addressed in different versions of this exam (past and present). Network security can (and should) be extremely complex and usually results in a series of interdependencies between systems operating in concert.

This book will show you how one system (or function) relies on another, and each chapter of the book provides insight into topics in other chapters. Many of the chapters that do not specifically address exam topics provide a foundation that is necessary for a clear understanding of network security. Your short-term goal might be to pass this exam, but your overall goal is to become a qualified network security professional.

Note that because security vulnerabilities and preventive measures continue apace, Cisco Systems reserves the right to change the exam topics without notice. Although you can refer to the list of exam topics listed in Table I-1, always check the Cisco Systems website to verify the actual list of topics to ensure that you are prepared before taking an exam. You can view the current exam topics on any current Cisco certification exam by visiting its website at Cisco.com, hovering over Training & Events, and selecting from the Certifications list. Note also that, if needed, Cisco Press might post additional preparatory content on the web page associated with this book at www.ciscopress.com/title/9781587142802. It's a good idea to check the website a couple of weeks before taking your exam to be sure that you have up-to-date content.

Overview of the Cisco Certification Process

The network security market is currently in a position where the demand for qualified engineers vastly surpasses the supply. For this reason, many engineers consider migrating from routing/networking to network security. Remember that "network security" is just "security" applied to "networks." This sounds like an obvious concept, but it is actually an important one if you are pursuing your security certification. You must be familiar with networking before you can begin to apply the security concepts. For example, the skills required to complete the CCNA exam will give you a solid foundation that you can expand into the network security field.

The requirements for and explanation of the CCNP certification are outlined at the Cisco Systems website. Go to Cisco.com, hover over Training & Events, and select CCNP from the Certifications list.

Taking the SECURE Certification Exam

As with any Cisco certification exam, it is best to be thoroughly prepared before taking the exam. There is no way to determine exactly what questions are on the exam, so the best way to prepare is to have a good working knowledge of all subjects covered on the exam. Schedule yourself for the exam and be sure to be rested and ready to focus when taking the exam.

The best place to find out the latest available Cisco training and certifications is under the Training & Events section at Cisco.com.

Tracking CCNP Status

You can track your certification progress by checking www.cisco.com/go/certifications/login. You must create an account the first time you log on to the site.

How to Prepare for an Exam

The best way to prepare for any certification exam is to use a combination of the preparation resources, labs, and practice tests. This guide has integrated some practice questions and labs to help you better prepare. If possible, you want to get some hands-on time with the Cisco IOS devices. There is no substitute for experience, and it is much easier to understand the commands and concepts when you can actually work with Cisco IOS devices. If you do not have access to Cisco IOS devices, you can choose from among a variety of simulation packages available for a reasonable price. Last, but certainly not least, Cisco.com provides a wealth of information about the Cisco IOS Software, all the products that operate using Cisco IOS Software, and the products that interact with Cisco devices. No single source can adequately prepare you for the SECURE exam unless you already have extensive experience with Cisco products and a background in networking or network security. At a minimum, you will want to use this book combined with the Technical Support and Documentation site resources (www.cisco.com/cisco/web/support/index.html) to prepare for this exam.

Assessing Exam Readiness

After completing a number of certification exams, we have found that you do not actually know whether you are adequately prepared for the exam until you have completed about 30 percent of the questions. At this point, if you are not prepared, it is too late. The best way to determine your readiness is to work through the “Do I Know This Already?” quizzes at the beginning of each chapter and the review questions in the “Fill in the Blanks” sections at the end of each chapter. It is best to work your way through the entire book unless you can complete each subject without having to do any research or look up any answers.

Cisco Security Specialist in the Real World

Cisco has one of the most recognized names on the Internet. You cannot go into a data center or server room without seeing some Cisco equipment. Cisco-certified security specialists can bring quite a bit of knowledge to the table because of their deep understanding of the relationship between networking and network security. This is why the Cisco certification carries such clout. Cisco certifications demonstrate to potential employers and contract holders a certain professionalism and the dedication required to complete a goal. Face it, if these certifications were easy to acquire, everyone would have them.

Cisco IOS Software Commands

A firewall or router is not normally something to play with. That is to say that after you have it properly configured, you will tend to leave it alone until there is a problem or you need to make some other configuration change. This is the reason that the question mark (?) is probably the most widely used Cisco IOS Software command. Unless you have constant exposure to this equipment, it can be difficult to remember the numerous commands required to configure devices and troubleshoot problems. Most engineers remember enough to go in the right direction but will use the ? to help them use the correct syntax. This is life in the real world. Unfortunately, the question mark is not always available in the testing environment. Many questions on this exam require you to select the best command to perform a certain function. It is extremely important that you familiarize yourself with the different commands and their respective functions.

Rules of the Road

We have always found it confusing when different addresses are used in the examples throughout a technical publication. For this reason, we use the address space defined in RFC 1918. We understand that these addresses are not routable across the Internet and are not normally used on outside interfaces. Even with the millions of IP addresses available on the Internet, there is a slight chance that we could have chosen to use an address that the owner did not want to have published in this book.

It is our hope that this will assist you in understanding the examples and the syntax of the many commands required to configure and administer Cisco IOS routers.

Exam Registration

The SECURE exam is a computer-based exam, with multiple-choice, fill-in-the-blank, list-in-order, and simulation-based questions. You can take the exam at any Pearson VUE (www.pearsonvue.com) testing center. Your testing center can tell you the exact length of the exam. Be aware that when you register for the exam, you might be told to allow a certain amount of time to take the exam that is longer than the testing time indicated by the testing software when you begin. This discrepancy is because the testing center will want you to allow some time to get settled and take the tutorial about the test engine.

Book Content Updates

Because Cisco Systems occasionally updates exam topics without notice, Cisco Press might post additional preparatory content on the web page associated with this book at www.ciscopress.com/title/9781587142802. It is a good idea to check the website a couple of weeks before taking your exam, to review any updated content that might be posted online. We also recommend that you periodically check back to this page on the Cisco Press website to view any errata or supporting book files that may be available.

This page intentionally left blank

This chapter covers the following subjects:

- **Routed data plane attack types:** Reviews the types of attack that are focused on the routed data plane.
- **Access control lists (ACL):** Covers the fundamentals of using ACLs and the configuration and verification commands to use.
- **Flexible Packet Matching (FPM):** Covers the steps involved in developing a traffic class and policy and assigning it to an interface. It also goes over the verification commands that can be used in configuration and troubleshooting.
- **Flexible NetFlow:** Reviews the fundamentals of Flexible NetFlow and describes the configuration and verification commands to use it.
- **Unicast Reverse Path Forwarding (Unicast RPF):** Covers the basics of how Unicast RPF functions and discusses the commands required to configure and verify it.

CHAPTER 8

Implementing and Configuring Cisco IOS Routed Data Plane Security

Several different parts of a network need to be secured from internal and external attack. The three planes as defined by Cisco include the data plane, management plane, and control plane, and these are split between those focused on the switched parts of the network and those focused on the routed parts of the network. This chapter addresses the routed data plane, including the Cisco IOS Software features that can be used to secure the network user data that traverses the network and discusses how to configure these features on the network devices within the network.

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz helps you decide whether you really need to read the entire chapter. If you already intend to read the entire chapter, you do not necessarily need to answer these questions now.

The ten-question quiz, derived from the major sections in the "Foundation Topics" section of this chapter, helps you determine how to spend your limited study time.

Table 8-1 outlines the major topics discussed in this chapter and the "Do I Know This Already?" quiz questions that correspond to those topics.

Table 8-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundation Topics Section	Questions Covered in This Section
Routed Data Plane Attack Types	1
Routed Data Plane Security Technologies	2–10

Caution: The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark this question wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. Which of the following are some of the most common types of routed data plane attacks?
   - **a.** Routing protocol spoofing
   - **b.** Slow-path denial of service
   - **c.** STP spoofing
   - **d.** Traffic flooding

2. Which of the following ACL ranges are used for standard access lists?
   - **a.** 100–199
   - **b.** 2000–2699
   - **c.** 1–99
   - **d.** 1300–1999

3. When using a reflexive access list, which of the following ACL types must be used?
   - **a.** Standard IP ACL
   - **b.** Extended IP ACL
   - **c.** Extended IP named ACL
   - **d.** Reflexive ACL
   - **e.** Standard IP named ACL

4. Which of the following are valid steps required for the creation of an FPM filtering policy?
   - **a.** Defining a service policy
   - **b.** Loading of a PCFD
   - **c.** Defining an access list
   - **d.** Loading of a PHDF

5. Which command are used to load a traffic classification file (TCDF)?
   - **a.** **load protocol**
   - **b.** **load classification**
   - **c.** **load tcdf**
   - **d.** **load class-file**

6. Which commands are used to configure matching for a traffic class?
   - **a.** **match field**
   - **b.** **match start**
   - **c.** **match beginning**
   - **d.** **match l2-layer**
   - **e.** **match packet**

7. Which of the following are restrictions when using FPM?

   a. Stateful inspection only

   b. IPv4/IPv6 unicast packets only

   c. IPv4 unicast packets only

   d. Cannot be used with IP options packets

8. Which of the following are benefits that are gained by using Flexible NetFlow?

   a. Flexible key and nonkey fields

   b. Version 5 export format

   c. Standardized key and nonkey fields

   d. Version 9 export format

9. Which of the following are Flexible NetFlow components?

   a. Flow sequencers

   b. Flow policers

   c. Flow monitors

   d. Flow samplers

10. Unicast RPF utilizes which of the following to compare source packet information?

   a. IP routing table

   b. CEF FIB

   c. Topology tables

   d. NetFlow records

The answers to the "Do I Know This Already?" quiz are found in Appendix A. The suggested choices for your next step are as follows:

- **8 or less overall score:** Read the entire chapter. This includes the "Foundation Topics" section.
- **9 or 10 overall score:** If you want more review on these topics, skip to the "Exam Preparation" section. Otherwise, move on to Chapter 9, "Implementing and Configuring Cisco IOS Control Plane Security."

Foundation Topics

Routed Data Plane Attack Types

As stated in previous chapters, understanding the attack makes the mitigation of the attack easier to accomplish. The routed infrastructure encompasses a large part of people's everyday lives, and because of this, it is a very large attack target. The following sections review the attacks that are targeted at the routed data plane.

The most common types of routed data plane attacks are

- IP spoofing
- Slow-path denial of service
- Traffic flooding

IP Spoofing

Although IP spoofing has been covered in earlier chapters, it is reviewed here. With IP spoofing, an attacker attempts to send and receive traffic on the network using an IP address of another known host or known network. The attacker is then able to use resources on the network that are associated with that specific IP address. IP spoofing is just as much of a threat on the routed network as it is with the switched network, but it is mitigated using different techniques and technologies. The three primary methods used to perform IP spoofing are as follows:

- Injecting packets with the IP address of an existing host
- Spoofing an existing host using source routing
- Injecting packets from nonexisting hosts to perform a denial of service attack.

Slow-Path Denial of Service

Generic denial of service attacks are well known because they are rather simple to understand. A host or group of hosts attempts to deny a specific service or services to their intended audience, typically through the flooding of traffic to the targeted sites. Slow-path denial of service looks to deny a service or services by sending a large number of packets through the routed pieces of equipment that are required to be process switched. Process switching compared with other alternatives is the "slow path" through the equipment. The CPU of each device is tasked to perform three functions:

- Process control plane traffic
- Process management plane traffic
- Process slow-path data plane traffic

This chapter focuses on the methods of mitigating slow-path data plane attacks.

Traffic Flooding

This attack type involves the flooding of packets at a specific target. Typically, these types of attack are focused on breaking down the functionality of the target host. The techniques described in this chapter can be used to mitigate traffic-flooding attacks on not only the data plane but also the control and management planes.

Routed Data Plane Security Technologies

A number of different security technologies can be used to mitigate the attacks covered in the previous sections. A description of these technologies and how to implement them will be covered in the following sections.

Access Control Lists (ACL)

ACLs are rules that deny or permit packets coming into or out of an interface. An ACL typically consists of multiple ACL entries (ACE), organized internally by the router. When a packet is subjected to access control, the router searches this linked list in order from top to bottom to find a matching element. The matching element is then examined to determine whether the packet is permitted or denied.

ACLs can be used to mitigate a number of attacks and can also be used in combination with other technologies to mitigate many more. ACLs can be used in small businesses or at the edge of larger businesses to mitigate both IP spoofing and slow-path denial of service attacks. These types of ACLs are called *infrastructure ACLs* because they look to protect not just the device but also the entire infrastructure. To mitigate IP spoofing attacks, an ACL is configured to automatically disallow inbound traffic that has a source IP address that is known to be inside the network. When being legitimately routed, these addresses, which are inside the network, will never be sourced outside the network. ACLs can also be used to screen traffic that has been sent in an effort to slow the device by forcing its traffic to be process switched. In small amounts, this is not an issue, but when a large number of packets need to be process switched, it has the chance of affecting the performance of the device.

Figure 8-1 shows the behavior of a router that has an ACL configured on its interfaces.

The function of ACLs includes their ability to

- Control the transmission of packets coming into or out of an interface
- Control virtual terminal line access
- Restrict contents of routing updates
- Define interesting traffic

There are two different methods to configure an ACL:

- **Numbered ACLs:** These are entered one line at a time, and the list is scanned for a match in that same order. If a change is required, the entire list must be reentered.
- **Named ACLs:** Theses provide a method of configuration that does not require the complete reentry of the ACL.

Key Topic

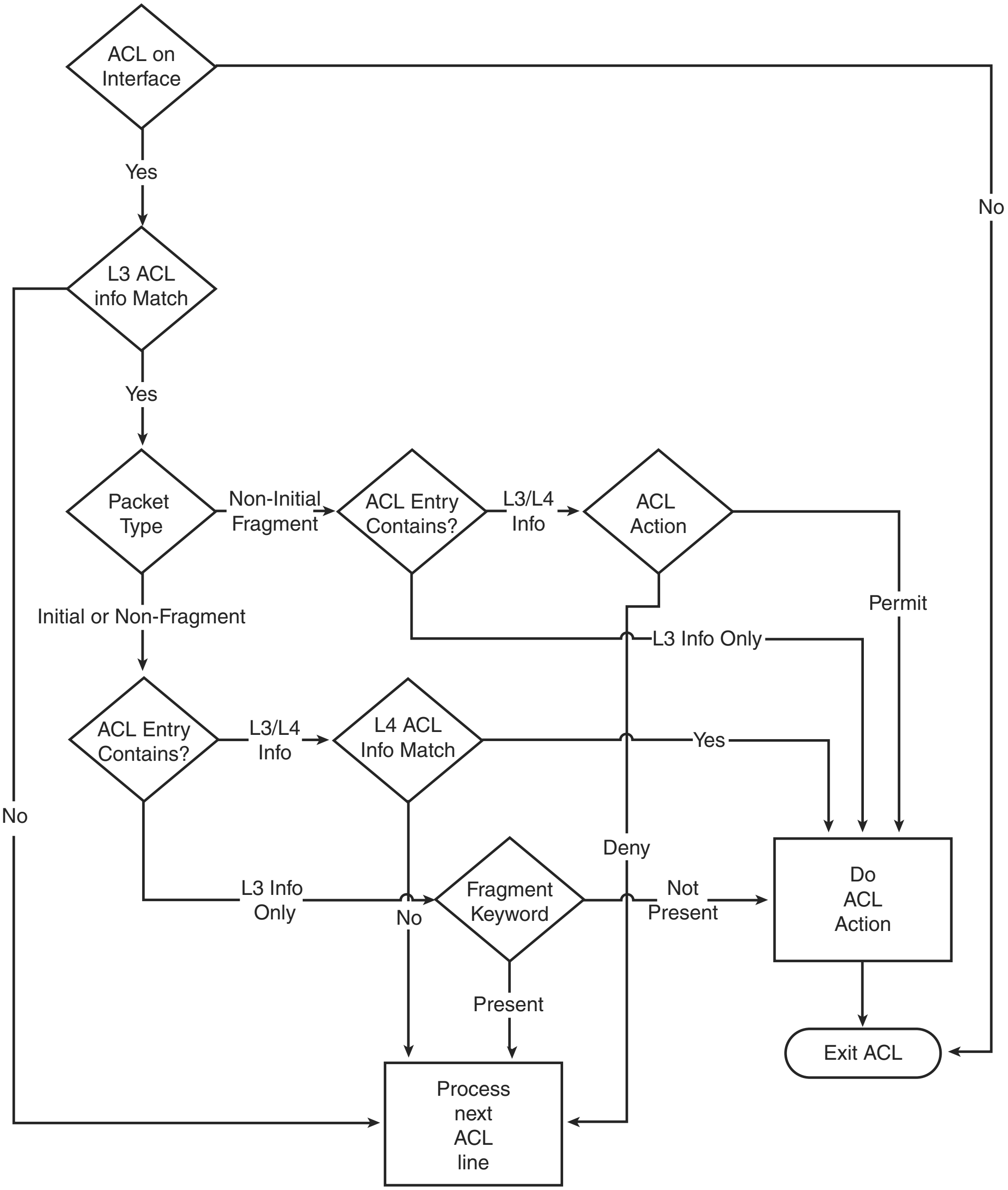


Figure 8-1 *High-Level Overview of How an ACL Is Processed by a Router*

The ACL criteria that can be used is quite large and includes information like the source and destination network layer information as well as a number of different fields provided by upper-layer protocols.

At the end of each ACL, there is an implied deny for traffic that has not been previously permitted. There must be at least one **permit** statement in an ACL; otherwise, all traffic will be blocked.

ACLs also have the capability to drop or ignore packets based on whether they contain any IP options. There are two ways in which this can be controlled: through the IP Options Selective Drop feature or through the use of the **option** keyword when creating an extended named access list. The IP Options Selective Drop feature is used by issuing the **ip options** {*drop* | *ignore*} **global configuration** command.

Determining Where and When to Configure Access Lists

To provide the security benefits of ACLs, at a minimum an ACL should be configured on the border routers, which are routers situated at the edges of the network. This setup provides a basic buffer from the outside network or from a less-controlled area of the network into a more sensitive area of the network.

An ACL can be configured so that inbound traffic or outbound traffic, or both, are filtered on an interface. ACLs should be defined on a per-protocol basis. In other words, an ACL should be defined for every protocol enabled on an interface if that protocols traffic is to be controlled.

Types of ACLs

Cisco IOS Software supports the following types of ACLs for IP:

Key Topic

- **Standard ACLs:** Use source addresses for matching operations.
- **Extended ACLs:** Use source and destination addresses for matching operations and optional protocol type information for finer granularity of control.
- **Reflexive ACLs:** Allow IP packets to be filtered based on session information. Reflexive ACLs contain temporary entries and are nested within extended-named IP ACLs.
- **Time-based ACLs:** As the name intuitively indicates, these ACLs are triggered by a time function.

The following sections discuss each type of ACL in detail.

Standard ACLs

Standard ACLs are the oldest type of ACLs, dating back as early as Cisco IOS Software Release 8.3. Standard ACLs control traffic by comparing the source address of the traffic to the addresses configured in the ACL.

The following is the command syntax format of a standard ACL:

```
router(config)# access-list access-list-number {permit | deny} {host | source
  source-wildcard | any} [log]
```

or

```
router(config)# ip access-list standard {access-list-number | access-list-name}
permit {host host | source source-wildcard | any} [log]
```

In all software releases, the access list number for standard IP access lists can be anything from 1 to 99. Table 8-2 shows the various protocol options and their corresponding number range for the ACL identification. In Cisco IOS Software Release 12.0.1, standard IP ACLs began using additional numbers (1300 to 1999). These additional numbers are referred to as *expanded IP ACLs*. In addition to using numbers to identify ACLs, Cisco IOS Software Release 11.2 and later added the ability to use the list *name* in standard IP ACLs.

Key Topic

Table 8-2 *Protocols and Their Corresponding Number Identification for an ACL*

Protocol	Range
Standard IP	1–99 and 1300–1999
Extended IP	100–199 and 2000–2699
Ethernet type code	200–299
Ethernet address	700–799
Transparent bridging (protocol type)	200–299
Transparent bridging (vendor code)	700–799
Extended transparent bridging	1100–1199
DECnet and extended DECnet	300–399
Xerox Network Systems (XNS)	400–499
Extended XNS	500–599
AppleTalk	600–699
Source-route bridging (protocol type)	200–299
Source-route bridging (vendor code)	700–799
Internetwork Packet Exchange (IPX)	800–899
Extended IPX	900–999
IPX Service Advertising Protocol (SAP)	1000–1099
Standard Virtual Integrated Network Service (VINES)	1–100
Extended VINES	101–200
Simple VINES	201–300

The **log** option enables the monitoring of how many packets are permitted or denied by a particular ACL, including the source address of each packet. The logging message includes the ACL number, whether the packet was permitted or denied, the source IP address of the packet, and the number of packets from that source permitted or denied in the prior 5-minute interval.

Wildcard masks are used in conjunction with IP addresses to identify the source address in an ACL. Wildcard masks are also known as *reverse netmasks* and are one of the topics that many people have considerable problem understanding. In an effort to make this a little clearer, an example will be shown. So, if the netmask normally is 255.255.255.0, it's this in binary:

11111111 11111111 11111111 00000000

Swapping the bits yields the reverse netmask, shown as follows:

00000000 00000000 00000000 11111111

or

0.0.0.255 (the wildcard mask)

Another way to calculate the wildcard mask is to take the network mask and subtract each octet from 255. If the network mask is 255.255.248.0, for example, the wildcard is calculated by subtracting it from 255 on each octet, yielding a 0.0.7.255 wildcard mask.

After defining an ACL, it must be applied to the interface (inbound or outbound):

```
router(config)# interface interface
router(config-if)# ip access-group number {in | out}
```

Example 8-1 shows the use of a standard IP ACL to block all traffic except that from source 192.168.100.x.

Example 8-1 *Sample ACL Configuration Permitting Network 192.168.100.0 into the FastEthernet 0/0 Interface and Implicitly Denying All Other IP Traffic*

```
router(config)# interface FastEthernet0/0
router(config-if)# ip address 192.168.100.1 255.255.255.0
router(config-if)# ip access-group 1 in
router(config)# access-list 1 permit 192.168.100.0 0.0.0.255
```

The terms *in*, *out*, *source*, and *destination* are used as referenced by the router. Traffic on the router could be compared to traffic on the highway. If a law enforcement officer in the United States wanted to stop a truck coming from Mexico and traveling to Canada, the truck's source would be Mexico and the truck's destination would be Canada. The roadblock could be applied at the U.S./Mexican border (in) or the U.S./Canadian border (out).

With regard to a router, these terms mean the following:

- **In:** Traffic that is arriving on the interface and that will go through the router; the source is where it has been, and the destination is where it is going.

- **Out:** Traffic that has already been through the router and is leaving the interface; the source is where it has been, and the destination is where it is going.

Extended IP ACLs

Extended IP ACLs were introduced in Cisco IOS Software Release 8.3. Extended IP ACLs can control traffic by not only comparing the source IP addresses but also comparing the destination IP address as well as other information, including the source and destination port numbers of the IP packets to those configured in the ACL.

The following is the command syntax format of extended IP ACLs:

```
router(config)# access-list access-list-number [dynamic dynamic-name [timeout
  minutes]]{deny | permit} protocol source source-wildcard destination
  destination-wildcard [precedence precedence] [tos tos] [log | log-input] [time-
  range time-range-name]
```

or

```
router(config)# ip access-list extended {access-list-number | access-list-name}
```

```
router(config-std-nacl)# [sequence-number] permit protocol source source-wildcard
destination destination-wildcard [option option-value] [precedence precedence]
[tos tos] [time-range time-range-name] [log]
```

or

```
router(config-ext-nacl)# [sequence-number] permit protocol source source-wildcard
destination destination-wildcard [option option-value] [precedence precedence]
[tos tos] [time-range time-range-name] [log]
```

In all software releases, the access list number for extended IP access lists can be 100 to 199. In Cisco IOS Software Release 12.0.1, extended IP ACLs began using additional numbers (2000 to 2699). These additional numbers are referred to as *expanded IP ACLs*. Cisco IOS Software Release 11.2 added the ability to use the list *name* in extended IP ACLs.

Example 8-2 shows an extended IP ACL used to permit traffic on the 192.168.100.x network (inside) and to receive ping responses from the outside while preventing unsolicited pings from people outside (permitting all other traffic).

Example 8-2 *Sample Configuration for an Extended IP ACL*

```
router(config)# access-list 101 deny icmp any 192.168.100.0 0.0.0.255 echo
router(config)# access-list 101 permit ip any 192.168.100.0 0.0.0.255
router(config)# interface FastEthernet0/0
router(config-if)# ip address 172.16.8.1 255.255.255.0
router(config-if)# ip access-group 101 in
```

Reflexive ACLs

Cisco IOS Software Release 11.3 introduced reflexive ACLs. Reflexive ACLs enable IP packets to be filtered based on upper-layer session information.

They are generally used in one of two ways:

- To allow outbound traffic out of an interface facing away from the internal network and filtering inbound traffic based on existing sessions originating inside the internal network
- To allow all inbound traffic to an interface facing toward the internal network and filtering outbound traffic based on the existing session originating inside the internal network

The former of these two is more typical with a network that does not utilize a demilitarized zone (DMZ). The latter is used to allow traffic into a DMZ but to not allow that traffic into the internal network without a previous connection initiated inside the internal network. Both of these are shown in Figures 8-2 and 8-3.

Reflexive ACLs can be defined only with extended named IP ACLs. They cannot be defined with numbered, standard named IP ACLs or with other protocol ACLs. Reflexive ACLs can be used in conjunction with other standard and static extended IP ACLs. The syntax for configuring a reflexive ACL is as follows:

```
router(config)# ip access-list extended {access-list-number | access-list-name}
  router(config-ext-nacl)# [sequence-number] permit protocol source source-wildcard
  destination destination-wildcard reflect name
```

and

```
router(config-ext-nacl)# evaluate
```

Example 8-3 demonstrates, by using Figure 8-2, the process of permitting all TCP traffic outbound and inbound TCP traffic that was initiated from inside the network.

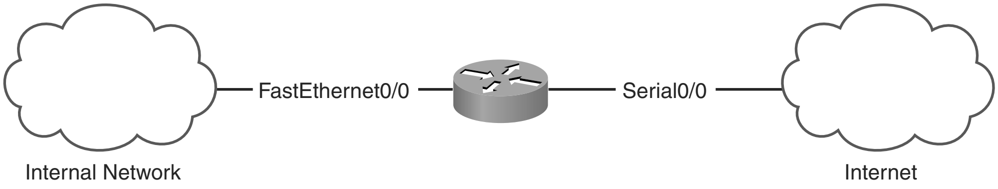


Figure 8-2 *Outbound Reflexive Diagram*

Example 8-3 *Sample Configuration for an Outbound Reflexive ACL*

```
router(config)# ip access-list extended outgoing
router(config-ext-nacl)# permit tcp any any reflect tcp-traffic
router(config)# ip access-list extended incoming
router(config-ext-nacl)# evaluate tcp-traffic
router(config)# interface Serial0/0
router(config-if)# ip address 192.168.100.1 255.255.255.0
router(config-if)# ip access-group incoming in
router(config-if)# ip accesss-group outgoing out
```

Example 8-4 demonstrates, by using Figure 8-3, the process of permitting all TCP traffic inbound and outbound TCP traffic that was initiated from inside the network.

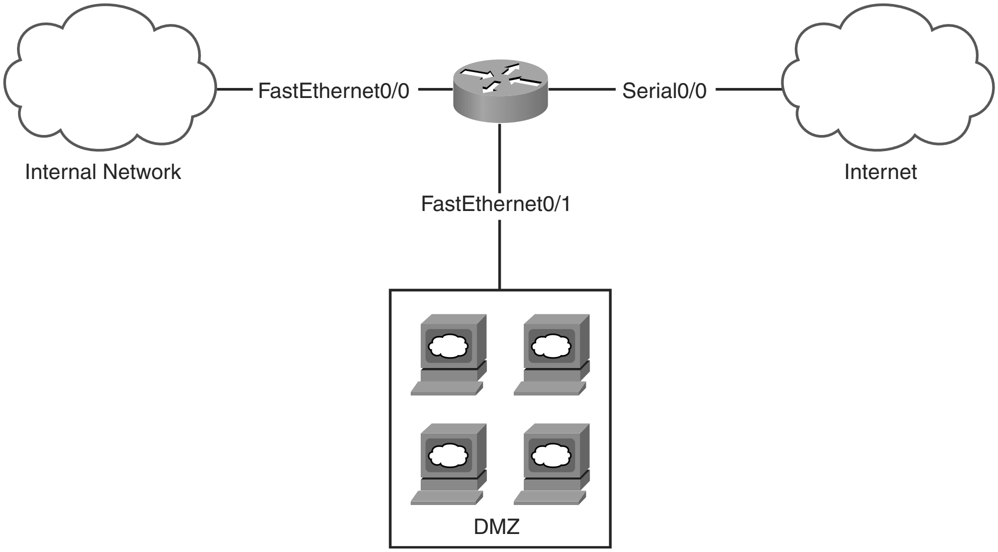


Figure 8-3 *Inbound Reflexive Diagram*

Example 8-4 *Sample Configuration for an Inbound Reflexive ACL*

```
router(config)# ip access-list extended incoming
router(config-ext-nacl)# permit tcp any any reflect tcp-traffic
router(config)# ip access-list extended outgoing
router(config-ext-nacl)# evaluate tcp-traffic
router(config)# interface FastEthernet0/0
router(config-if)# ip address 172.16.1.1 255.255.255.0
router(config-if)# ip access-group incoming in
router(config-if)# ip accesss-group outgoing out
```

Time-Based ACLs

Cisco IOS Software Release 12.0.1.T introduced time-based ACLs. Although similar to extended IP ACLs in function, they allow access control based on time. To implement time-based ACLs, a time range is created that defines specific times of the day and week. The time range is identified by a name and then referenced by a function. Therefore, the time restrictions are imposed on the function itself. The time range relies on the router's system clock. The router clock can be used solely, but the feature works best when Network Time Protocol (NTP) synchronization is used on the device.

Time-based ACL commands require the following syntax:

```
router(config)# time-range time-range-name
router(config-time-range)# periodic days-of-the-week hh:mm to [days-of-the-
  week] hh:mm
router(config-time-range)# absolute [start time date] [end time date]
```

and

```
router(config)# access-list access-list-number protocol source source-wildcard
  destination destination-wildcard [time-range time-range-name]
```

or

```
router(config)# ip access-list extended {access-list-number ¦ access-list-name}
router(config-ext-nacl)# [sequence-number] permit protocol source source-wildcard
  destination destination-wildcard [time-range time-range-name]
```

or

```
router(config)# ip access-list extended {access-list-number ¦ access-list-name}
router(config-ext-nacl)# [sequence-number] permit protocol source source-wildcard
  destination destination-wildcard [precedence precedence] [tos tos] [time-range
  time-range-name] [log]
```

Example 8-5 shows a Telnet connection permitted from the outside the network (172.16.1.0) to the inside of the network (192.168.1.0) on Monday, Tuesday, and Thursday during the hours of 7 a.m. through 6 p.m.

Example 8-5 *Sample Configuration for Time-Range ACL*

```
router(config)# interface FastEthernet0/0
router(config-if)# ip address 192.168.1.1 255.255.255.0
router(config)# interface FastEthernet0/1
router(config-if)# ip address 172.16.1.1 255.255.255.0
router(config-if)# ip access-group 101 in
router(config)# access-list 101 permit tcp 172.16.1.0 0.0.0.255 192.168.1.0
0.0.0.255 eq telnet time-range TelnetAccess
router(config)# time-range TelnetAccess
router(config-time-range)# periodic Monday Tuesday Thursday 7:00 to 18:00
```

Time ranges offer many possible benefits, including the following:

- The network administrator has more control over permitting or denying a user access to resources. These resources include an application (identified by an IP address/mask pair and a port number), policy routing, or an on-demand link (identified as interesting traffic to the dialer).

- When provider access rates vary by time of day, it is possible to automatically reroute traffic cost-effectively.
- Service providers can dynamically change a committed access rate (CAR) configuration to support the quality of service (QoS) service-level agreements (SLA) that are negotiated for certain times of day.
- Network administrators can control logging messages. ACL entries can log traffic at certain times of the day but not constantly. Therefore, administrators can just deny access without analyzing the many logs generated during peak hours.
- Policy-based routing and queuing functions are enhanced.

ACL Verification

There are a number of different **show** commands that can be used to verify ACL configuration.

To display the contents of all current access lists, enter the following command:

```
show access-list [access-list-number | access-list-name}
```

To display the contents of all current IP access lists, enter the following command:

```
show ip access-list [access-list-number | access-list-name}
```

Flexible Packet Matching

Flexible Packet Matching (FPM) was created to be a more thorough and customized packet filter option. FPM enables the user to configure match parameters based on arbitrary bits of a packet and arbitrary depths within the packet header and payload. This technique can be used to mitigate several different types of attack, including slow-path denial of service and zero-day virus and malware.

FPM is implemented using a filtering policy that is divided into four tasks:

- Loading of a Protocol Header Description File (PHDF)
- Defining a class map and a specific protocol stack chain (traffic class)
- Defining a service policy (traffic policy)
- Application of a service policy on a specific interface

Key Topic

FPM Restrictions

As with all technologies, a number of different restrictions must be known before attempting to configure FPM. The main restrictions for FPM include

- FPM is stateless; it cannot keep track of traffic flows through the configured interface (for example, port numbers).
- FPM inspects only IPv4 unicast packets.
- FPM cannot classify packets with IP options.

- FPM is not supported on tunnel or Multiprotocol Label Switching (MPLS) interfaces.
- FPM cannot be configured on FlexWAN cards.
- Noninitial fragments will not be matched by FPM.

Protocol Header Description File

With FPM, two different methods can be used to match specific traffic: the use of a Protocol Header Description File (PHDF) and/or the direct matching of traffic based on length and offset, or a mix of the two. A PHDF is used to define the various field names within a specific protocol. For example, the IP.phdf file has a field defined for each field in an IP header (that is, Version, Header Length, ToS, and so on), and TCP.phdf has a field defined for each field in the TCP header (that is, Source and Destination Port, Sequence Number, Acknowledgment Number, and so on). To take advantage of these field names, this file must first be loaded with the **load protocol** global configuration command. Loading a PHDF file also provides the ability to use the **match field** class map configuration command, which provides the ability to match based on this PHDF field information. Without loading the PHDF file, only the **match start** class map configuration command is supported, which provides the capability to match based on specific length and offset information. Both of these commands provide different methods for matching specific information within the packet and will be covered in more depth in the following sections. It is also possible for PHDFs to be custom written for other protocols; PHDFs are XML files and can be easily adapted for these purposes. The specific command syntax required to load the PHDF files is as follows:

```
router(config)# load protocol location:filename
```

Example 8-6 demonstrates the loading of both the IP and TCP PHDF files for use with the **match field** command.

Example 8-6 *Sample Configuration for the* **load protocol** *Command*

```
router(config)# load protocol system:fpm/phdf/ip.phdf
router(config)# load protocol system:fpm/phdf/tcp.phdf
```

Defining a Traffic Class

When creating a traffic class, its purpose is to define a number of criteria that are used to match specific traffic based on stateless packet classification. A simple example of this would be to match based on TCP traffic with a port number equal to 80 (web traffic). Of course, this type of example is simple and can be accomplished with common access list commands that are used more often for these types of matches. However, FPM provides the capability to not only match based on a specific criteria like a TCP port number but also based on a specific set of criteria, such as TCP port 80, with an IP packet length of less than 400 bytes, with a specific pattern 4 bytes long at offset 400. Now at first glance, why would someone need this capability? Well in the modern world, a number of threats exist, many of which are being created every day. Many of these are caught and prevented using tools such as intrusion protection systems (IPS); however, some attacks are so new that a signature is not yet available for the IPS. This is where the flexibility of FPM comes in handy. If an attack is occurring and a pattern is able to be distinguished, FPM can be used to surgically drop these attack packets inline without interruption of other uninfected traffic.

With FPM, two different methods can be used to configure traffic classes:

- Traffic can be classified using a Traffic Classification Definition File (TCDF).
- Traffic can be classified through the CLI using class maps.

When using a TCDF, a file must be created and then loaded. The TCDF file uses XML and is rather simple to create. TCDFs offer a method of implementing the same matching criteria as the CLI commands, but allow them to be repetitively used over a number of different devices without the hassle of manually adding commands on each device. The steps used to create the match criteria are the same as when using the CLI. These specific steps will be covered in the text that follows in CLI terms, and specific examples will be included showing the correct TCDF format. Use of a TCDF requires the **load classification** command to load the TCDF file on the device. The command syntax required for this command is as follows:

```
router(config)# load classification location:filename
```

Because CLI configuration is the most commonly understood method of configuration, this type of configuration will be covered in depth. The first thing that must be configured with the CLI is a class map; this is done using the **class-map** command. This command is well known because it is used for many other tasks within IOS and is configured similarly. Two class map types are used with FPM:

Key Topic

- **Stack:** Specifies the specific protocol stacks that will be used to match (for example, IP, TCP, UDP) and can be only used with the **match-all** keyword.
- **Access control:** Matches specific patterns within the traffic of interest.

The command syntax required to create these class maps is as follows:

CLI:

```
router(config)# class-map type [stack | access-control] [match-all | match-any]
  class-map-name
```

TCDF:

```
<?xml version"1.0" encoding="UTF-8"?>
<tcdf>
    <class name="class-name" type="stack | access-control" match="any |
      all"></class>
    ...
</tcdf>
```

The second part of this process is configuring specific match criteria; to do this, the **match start** and the **match field** commands are used. As stated earlier, the **match field** command only works after a PHDF has been loaded. The **match field** command is used to match based on the PHDF fields loaded. The **match start** command is used to match a specific pattern based on a specific offset and length and whether to begin inspection at the beginning of the Layer 3 packet header or at the beginning of the Layer 2 frame header. The command syntax for these commands is as follows:

CLI:

```
router(config-cmap)# match field protocol protocol-field [eq | neq | gt | lt | range
   range] value next next-protocol

router(config-cmap)# match start [l2-start | l3-start] offset offset size size
  [eq | neq | gt | lt | range range] value
```

TCDF:

```
<?xml version"1.0" encoding="UTF-8"?>
<tcdf>
     ...
          <match>
               <[eq | neq | gt | lt] field="field-name" value="value"></[eq | neq | gt
                  | lt]>
               <range field="field-name" from="beginning-value" to="ending-
                  value"></range>
          </match>
     ...
</tcdf>
```

To wrap up all the different commands required for a traffic class, Example 8-7 shows a sample configuration. In this example, two different separate class maps are being created:

- **tcp-class:** This class map is configured to match the IP protocol header field when it is equal to 0x6 (TCP) and tells FPM that the next protocol to be analyzed will be TCP.
- **sample-match:** This class map is configured to match traffic that has a TCP destination port that is equal to 0x50 (80) *and* has the contents “0x1234” at offset 200 in the IP packet.

Example 8-7 *Sample Traffic Class Configuration*

```
CLI:


router(config)# class-map type stack match-all tcp-class
router(config-cmap)# match field ip protocol eq 0x6 next tcp

router(config)# class-map type access-control match-all sample-match
router(config-cmap)# match field tcp dest-port eq 0x50
router(config-cmap)# match start l3-start offset 200 size 2 eq 0x1234

TCDF:

<?xml version"1.0" encoding="UTF-8"?>
<tcdf>
     <class name="tcp-class" type="stack" match="all">
```

```
        <match>
            <eq field="ip.protocol" value="0x6" next="tcp"></eq>
        </match>
    </class>
    <class name="sample-match" type="access-control" match="all">
        <match>
            <eq field="tcp.dest-port" value="0x50"></eq>
            <eq start="l3-start" offset="200" size="2" value="0x1234"></eq>
        </match>
    </class>
</tcdf>
```

Defining a Traffic Policy

The next step is to configure what to do with the traffic that was matched with the class map; this is done through the creation of a traffic policy. The policy must use one (or more) of the configured traffic classes to match specific traffic and then configure what to do with this traffic after it is found.

The first part required for the configuration of a traffic policy is the creation of a policy map using the **policy-map** command; the command syntax for this command is as follows:

CLI:

```
router(config)# policy-map type access-control policy-map-name
```

TCDF:

```
<?xml version"1.0" encoding="UTF-8"?>
<tcdf>
    ...
    <policy name="policy-name"></policy>
    ...
</tcdf>
```

The second part of the process is specifying a traffic class that is configured using the **class** command; the command syntax for this command is as follows:

CLI:

```
router(config-pmap)# class class-name
```

TCDF:

```
<?xml version"1.0" encoding="UTF-8"?>
<tcdf>
    ...
    <class name="policy-name"></class>
    ...
</tcdf>
```

The final part of the process is configuring what action will be taken should a match occur; the command syntax for this command is as follows:

CLI:

```
router(config-pmap-c)# drop
```

TCDF:

```
<?xml version"1.0" encoding="UTF-8"?>
<tcdf>
    ...
    <action>Drop</action>
    ...
</tcdf>
```

An additional part can be added to a traffic policy by nesting policies. To take advantage of this functionality, the **service-policy** command is used, and the syntax is as follows:

CLI:

```
router(config-pmap-c)# service-policy policy-map-name
```

TCDF:

```
<?xml version"1.0" encoding="UTF-8"?>
<tcdf>
    ...
    <action>service-policy policy-map-name</action>
    ...
</tcdf>
```

There also seems to be a caveat when utilizing nesting policies with a TCDF file: The action tag will only allow a policy map name of up to 16 characters, which is not true when configuring nesting using only the CLI.

To wrap up the different commands required for a traffic policy, Example 8-8 shows a sample configuration. In this example, two different policy maps are created. One of these policy maps is then configured to nest inside the other. This traffic policy would be processed like this:

Step 1. Within the tcp-policy policy map, all traffic that is matched with the class map tcp-class will be sent to the tcp-policy-nest policy. (This would include all TCP traffic.)

Step 2. Within the tcp-policy-nest policy map, all traffic matching the class map sample-match would be dropped. (This would include traffic with a TCP destination port 0x50 (80) *and* has the contents “0x1234” at offset 200 in the IP packet.)

Example 8-8 *Sample Traffic Policy Configuration*

```
CLI:


router(config)# policy-map type access-control tcp-policy-nest
router(config-pmap)# class sample-match
router(config-pmap-c)# drop

router(config)# policy-map type access-control tcp-policy
router(config-pmap)# class tcp-class
router(config-pmap-c)# service-policy tcp-policy-nest

TCDF:

<?xml version"1.0" encoding="UTF-8"?>
<tcdf>
    ...
    <policy type="access-control" name="tcp-policy-nest">
        <class name="sample-match"></class>
            <action>Drop</action>
    </policy>

    <policy type="access-control" name="tcp-policy">
        <class name="tcp-class"></class>
            <action>service-policy tcp-policy-nest</action>
    </policy>
    ...
</tcdf>
```

Application of a Traffic Policy

The final step in this process is the application of the traffic policy on a specific interface. This application can be configured in either an incoming (input) or an outgoing (output) direction. The **service-policy type access-control** command is used to apply a specific traffic policy to an interface; the syntax for this command is as follows:

```
router(config-if)# service-policy type access-control [input | output] policy-
  map-name
```

Example 8-9 shows the application of the tcp policy policy map onto the FastEthernet0/0 interface.

Example 8-9 *Sample Traffic Policy Application Configuration*

```
CLI:


router(config)# interface FastEthernet0/0
router(config-if)# service-policy type access-control input tcp-policy
```

FPM Verification

Many different **show** commands can be used to verify FPM configuration.

To show which specific PHDFs are loaded and which fields are supported, enter the following command:

```
show protocols phdf phdf-name
```

To display the current traffic classes configured and matching criteria, enter the following command:

```
show class-map type [stack | access-control]
```

To display the current traffic policies, enter the following command:

```
show policy-map type access-control {interface interface}
```

Flexible NetFlow

As the name suggests, Flexible NetFlow is a more flexible version of NetFlow that allows additional options that make it superior to the original in many ways. These additional benefits include

- Scalable, aggregatable high-capacity flow information
- Enhanced flow structure focused on additional security-monitoring capabilities
- Flexible key and nonkey field configuration
- NetFlow Version 9 export format (flexible structure)
- Comprehensive IP and BGP accounting

Both NetFlow and Flexible NetFlow work by identifying and recording flow information. A *flow* is a group of packets that have the same key fields. With the original version of NetFlow, these key fields were static and included: source and destination IP addresses, source and destination ports, protocol, interface, and class of service (CoS). Along with this key field information, nonkey fields, including the number of packets and number of total bytes, were also recorded. Flexible NetFlow allows these key and nonkey fields to be customizable and thus can be used in a large number of ways, depending on the specific information that is being sought.

So, the next question is "What can Flexible NetFlow be used for?" There are a number of things, from traffic accounting to capacity planning to security monitoring. This includes the ability to track slow-path and normal denial of service attacks and attack attempts. The following is a list (per Cisco) of typical Flexible NetFlow uses:

- Network monitoring
- Application monitoring and profiling
- User monitoring and profiling
- Network planning and analysis

- Security analysis
- Billing and accounting
- Data warehousing and data mining

Components

A couple of main components must be understood to configure Flexible NetFlow, as outlined in Table 8-3.

Key Topic

Table 8-3 *Flexible NetFlow Components*

Component	Description
NetFlow records	As information is collected by NetFlow, flows are defined by the configured key and nonkey fields. When there is a unique match of key fields, the matching traffic information will be recorded in a cache as a NetFlow record. As additional matching traffic occurs, the record is updated with this additional information (for example, byte counts increase if the field is configured). With Flexible NetFlow, there are predefined and user-defined record layout possibilities.
Flow monitors	Flow monitors are attached to interfaces and perform the network-monitoring tasks. The flow monitor is configured with a specific record format, an optional flow exporter, and a cache.
Flow exporters	A flow exporter's job is rather self explanatory: It exports data from the NetFlow cache to a remote system. With Flexible NetFlow, this is typically done using the NetFlow Data Export Format, V9.
Flow samplers	A flow sampler reduces the load on the network device. By default, NetFlow records flows based on all the traffic in a specific direction (or both if configured). Because this can constitute a large amount of traffic on busy devices, the concept of a flow sampler was created. A flow sampler is configured to change the number of packet captures from all packets to a sampled number of packets based on configuration (for example, every other packet—50%).

NetFlow Records

The first thing that must be covered with NetFlow records is key and nonkey fields and the difference between them. The difference is rather simple: A key field is used to identify a specific flow, whereas a nonkey field is simply recorded as part of an already identified flow. This difference is important when utilizing the user-defined options available with Flexible NetFlow.

Now with the original NetFlow, the key and nonkey fields were static and provided no flexibility. Flexible NetFlow resolved this by allowing user-defined record structures. However, because original NetFlow has an established configuration base, it was important to include backward compatibility. Table 8-4 displays the NetFlow Original/NetFlow IPv4 Original Input record format, and Table 8-5 displays the NetFlow IPv4 Original Output record format; both of the tables include key/nonkey field information.

Table 8-4 *NetFlow Original/NetFlow IPv4 Original Input Format*

Key Topic

Field	Key or Nonkey	Description
IP ToS	Key	Value of the IP ToS field
IP Protocol	Key	Value of the IP Protocol field
IP Source Address	Key	—
IP Destination Address	Key	—
Transport Source Port	Key	Transport layer source port
Transport Destination Port	Key	Transport layer destination port
Interface Input	Key	Receiving interface
Flow Sampler ID	Key	ID of the flow sampler (if used)
IP Source AS	Nonkey	Source Autonomous System Number
IP Destination AS	Nonkey	Destination Autonomous System Number
IP Next Hop Address	Nonkey	Next-hop IP address
IP Source Mask	Nonkey	—
IP Destination Mask	Nonkey	—
TCP Flags	Nonkey	Value of the TCP Flag field
Interface Output	Nonkey	Transmitting Interface
Counter Bytes	Nonkey	—
Counter Packets	Nonkey	—
Time Stamp System Uptime First	Nonkey	System uptime, when the first packet was switched
Time Stamp System Uptime Last	Nonkey	System uptime, when the last packet was switched

Table 8-5 *NetFlow IPv4 Original Output Format*

Key Topic

Field	Key or Nonkey	Description
IP ToS	Key	Value of the IP ToS field
IP Protocol	Key	Value of the IP Protocol field

Table 8-5 *NetFlow IPv4 Original Output Format*

Field	Key or Nonkey	Description
IP Source Address	Key	—
IP Destination Address	Key	—
Transport Source Port	Key	Transport layer source port
Transport Destination Port	Key	Transport layer destination port
Interface Output	Key	Transmitting interface
Flow Sampler ID	Key	ID of the flow sampler (if used)
IP Source AS	Nonkey	Source Autonomous System Number
IP Destination AS	Nonkey	Destination Autonomous System Number
IP Next Hop Address	Nonkey	Next-hop IP address
IP Source Mask	Nonkey	—
IP Destination Mask	Nonkey	—
TCP Flags	Nonkey	Value of the TCP Flag field
Interface Input	Nonkey	Receiving interface
Counter Bytes	Nonkey	—
Counter Packets	Nonkey	—
Time Stamp System Uptime First	Nonkey	System uptime, when the first packet was switched
Time Stamp System Uptime Last	Nonkey	System uptime, when the last packet was switched

As can be seen, a large amount of information was recorded in this original format. Flexible NetFlow provides the capability to pare down these fields to only those needed. To use only specific fields, a user-defined record format would be defined. The command syntax required to create this record format is as follows:

```
router(config)# flow record flow-record-name
```

To specify key fields:

```
router(config-flow-record)# match [ipv4 | ipv6 | datalink | routing | flow |
  interface} options
```

To specify nonkey fields:

```
router(config-flow-record)# collect [counter | ipv4 | ipv6 | datalink | routing |
  flow | interface | timestamp] options
```

Example 8-10 shows the commands that would be required to identify flows by the source and destination IP addresses and TCP source and destination port information and to record the packet and byte counts for each flow.

Example 8-10 *Sample Flow Record Configuration*

```
router(config)# flow record test-record-name
router(config-flow-record)# match ipv4 source address
router(config-flow-record)# match ipv4 destination address
router(config-flow-record)# match transport tcp source-port
router(config-flow-record)# match transport tcp destination-port
router(config-flow-record)# collect counter packets
router(config-flow-record)# collect counter bytes
```

Flow Monitors

The flow monitors attach to an interface where the traffic information is captured, either in an incoming (input) or outgoing (output) direction. However, before the flow monitor is assigned to an interface, it must be configured. The flow monitor requires that at least a NetFlow record format is configured to operate. At this point, a flow exporter can also be configured; this is covered in the next section. Also note that the support for IPv6 records was added in Release 12.3(20)T. The command syntax required for flow monitor configuration is as follows:

```
router(config)# flow monitor flow-monitor-name

router(config-flow-monitor)# record [flow-record-name | netflow | netflow-
  original] {ipv4 | ipv6} {original-input | original-output}
```

Example 8-11 shows the configuration of the test monitor name flow monitor with a custom flow record named test-record-name.

Example 8-11 *Sample Flow Monitor Configuration*

```
router(config)# flow monitor test-monitor-name
router(config-flow-monitor)# record test-record-name
```

Flow Exporter

A flow exporter is used to take the inactive (default = 15 seconds) or long-active (default = 30 minutes) records and export them to a remote system for analysis and/or storage. The command syntax required for flow exporter configuration is as follows:

```
router(config)# flow exporter flow-exporter-name

    router(config-flow-exporter)# destination [hostname | ip-address]
    router(config-flow-exporter)# transport udp port
router(config-flow-monitor)# exporter flow-exporter-name
```

Example 8-12 shows the configuration of a flow exporter named test-exporter-name with a destination address of 192.168.1.1 using UDP port 1234. This example then shows the flow exporter being applied to a flow monitor.

Example 8-12 *Sample Flow Exporter Configuration*

```
router(config)# flow exporter test-exporter-name
router(config-flow-exporter)# destination 192.168.1.1
router(config-flow-exporter)# transport udp 1234

router(config-flow-monitor)# exporter test-exporter-name
```

Flow Sampler

A flow sampler, as stated previously, is used when the amount of processing is either too much for the device to handle or is simply higher than acceptable. Two modes of sampling can be utilized:

Key Topic

- **Deterministic:** When using the deterministic mode, traffic is sampled at a configured interval; this mode requires less overhead than random mode. Deterministic mode is recommended when traffic patterns are random in nature.
- **Random:** When using random mode, traffic is sampled randomly; this mode should be used to eliminate any potential monitoring bias and to counter any user attempting to avoid monitoring.

The command syntax required for flow sampler configuration is shown as follows:

```
router(config)# sampler sampler-name
router(config-sampler)# mode {deterministic | random} 1 out-of window-size
```

Example 8-13 shows the configuration of a flow sampler named test-sampler with deterministic sampling with a window size of 2.

Example 8-13 *Sample Flow Sampler Configuration*

```
router(config)# sampler test-sampler
router(config-sampler)# mode deterministic 1 out-of 2
```

Application of a Flow Monitor

The final step in this process is the application of the flow monitor on a specific interface. This application can be configured in either an incoming (input) or an outgoing (output) direction. The **ip flow-monitor** command is used to apply a specific flow monitor to an interface; the syntax for this command is as follows:

```
router(config-if)# ip flow monitor flow-monitor-name {sampler sampler-name}
  [multicast | unicast] [input | output]
```

Example 8-14 shows the application of a flow monitor named test-monitor-name onto the FastEthernet0/0 interface using the flow sampler named test-sampler on input traffic.

Example 8-14 *Sample Flow Monitor Application Configuration*

```
router(config)# interface FastEthernet0/0
router(config-if)# ip flow monitor test-monitor-name sampler test-sampler input
```

Flexible NetFlow Verification

Many different **show** commands can be used to verify Flexible NetFlow configuration.

To verify flow monitor configuration, enter the following command:

```
show flow monitor
```

To verify that a flow monitor is enabled on an interface, enter the following command:

```
show flow interface interface
```

To verify flow exporter configuration, enter the following commands:

```
show flow exporter
show running-config flow exporter flow-exporter-name
```

To view the NetFlow cache, enter the following command:

```
show flow monitor name flow-monitor-name cache format [csv | record | table]
```

To view flow sampler configuration, enter the following command:

```
show sampler
```

Additional Flexible NetFlow Information

A lot more information is available at Cisco.com that was not possible to fit into this book. To gain access to this information, go to www.cisco.com/en/US/docs/ios/fnetflow/configuration/guide/12_4t/fnf_12_4t_book.pdf.

Unicast Reverse Path Forwarding (Unicast RPF)

On modern networks, one of the most common attack types involves the forging or spoofing of IP source addresses. The configuration of ACLs for this purpose on large networks can be very cumbersome and hard to maintain. In an attempt to develop a technology to deal with these issues, Unicast Reverse Path Forwarding (URPF) was developed. Unicast RPF provides a source validation step to packet handling; it does this by verifying the source information of a packet to information contained within the Cisco Express Forwarding (CEF) Forwarding Information Base (FIB). The CEF FIB is a table that contains packet-switching information that mirrors that of the routing table; this is used by the device to increase the speed of packets being forwarding through the device. Because Unicast RPF relies on CEF's FIB, CEF must be configured on the device before Unicast RPF is configured.

Unicast RPF operates in one of two modes:

Key Topic

- **Strict (normal):** When in strict mode, Unicast RPF verifies that the source address is in the FIB *and* that the source address was received on the best return route interface as determined by CEF. This operation, while thorough, can also be troublesome if routing is multihomed. This is because the best return path might not be the same as the receiving interface; because of this, strict Unicast RPF is limited to single-homed connections. Unicast RPF will also work in situations where there are multiple equal-metric best paths available; this includes Enhanced IGRP (EIGRP) configurations where metric variance is configured. The recommended applications for strict Unicast RPF include (a) where only single connections are available to enter/exit the network, including the edge of a network, or (b) where single-homed customer connections connect into the core network because this would meet the single-homed requirement.
- **Loose:** Loose mode verifies only that the source address exists within the FIB and *not* the interface. Loose mode allows additional flexibility to implement Unicast RPF in locations where multihoming is common, including within a network.

Another important thing to understand about Unicast RPF is that it only works on incoming (input) interfaces. So, if a single-homed connection existed between the network and an ISP, RPF would be configured to monitor traffic coming from the ISP only. The use of Unicast RPF also increases the performance of the device over using traditional ACL methods of spoofing protection; this is because, unlike ACLs, Unicast RPF operates at CEF forwarding rates. When configuring Unicast RPF on interfaces over 1 Mbps, this processing difference is important.

Unicast RPF Configuration

The first thing that must be configured before starting Unicast RPF configuration is to enable the use of CEF. The **ip cef** *distributed* command enables the use of CEF; the syntax for this command is as follows:

```
ip cef {distributed}
```

The next part requires enabling Unicast RPF on the incoming interface. The **ip verify unicast source reachable-via** command is used to enable the use of Unicast RPF on an interface; the syntax for this command is as follows:

```
ip verify unicast source reachable-via [rx | any] {access-list}
```

The use of the **rx** or **any** keyword determines which mode that Unicast RPF will operate in; **rx** is used for strict mode and **any** is used for loose mode. An access list can also be specified with this command; its purpose is to determine whether the traffic will be dropped (default behavior - deny) or forwarded (permit). It is important to understand that this access list is not considered unless the packet fails the Unicast RPF check.

Unicast RPF Verification

A few commands can be used to verify the operation of Unicast RPF.

To verify that Unicast RPF is operational, enter the following command:

```
show cef interface interface
```

To verify global Unicast RPF packet count, enter the following command:

```
show ip traffic
```

To verify the number of interface Unicast RPF packet drops (verification drops)/forwards (suppressed verification drops), enter the following command:

```
show ip interface interface
```

Exam Preparation

As mentioned in the section, “How to Use This Book,” in the Introduction, you have several choices for exam preparation: the exercises here, the memory tables in Appendix D, the final exam preparation chapter, and the exam simulation questions on the CD-ROM. The following questions present a bigger challenge than the exam itself because they use an open-ended question format. By using this more difficult format, you exercise your memory better and prove your conceptual and factual knowledge of this chapter. You can find the answers to these questions in Appendix A, “Answers to the DIKTA Quizzes and Fill in the Blanks Questions.”

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topics icon in the margin of the page. Table 8-6 lists a reference of these key topics and the page numbers on which each is found.

Table 8-6 *Key Topics*

Key Topic Element	Description	Page
Figure 8-1	High-level overview of how an ACL is processed by a router	188
List	ACL types	189
Table 8-2	Protocols and their corresponding number identification for an ACL	190
List	FPM restrictions	196
List	FPM class-map types	198
Table 8-3	Flexible NetFlow components	204
Table 8-4	NetFlow original/NetFlow IPv4 original input format	205
Table 8-5	NetFlow IPv4 original output format	205
List	Flow sampling modes	208
List	Unicast RPF modes	210

Complete Tables and Lists from Memory

Print a copy of Appendix C, “Memory Tables” (found on the CD), or at least the section for this chapter, and complete the tables and lists from memory. Appendix D, “Memory Table Answers,” also on the CD, includes completed tables and lists to check your work.

Define Key Terms

Define the following key terms from this chapter, and check your answers in the Glossary:

access control list (ACL), stateless

Use Command Reference to Check Your Memory

Table 8-7 lists the important commands from this chapter. To test your memory, cover the right side of the table with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

Table 8-7 *Command Reference*

Task	Command Syntax
Create a standard access list	**access-list** *access-list-number* {**permit | deny**} {*host* | *source source-wildcard* | **any**} [**log**] or **ip access-list standard** {*access-list-number* | *access-list-name*} **permit** {**host** *host* | *source source-wildcard* | **any**} [**log**]
Create an extended access list	**access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]]{**deny | permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log |** log-input] **[time-range** *time-range-name*] or **ip access-list extended** {*access-list-number* | *access-list-name*} [*sequence-number*] {**deny | permit**} *protocol source source-wildcard destination destination-wildcard* [option *option-value*] [**precedence** *precedence*] [**tos** *tos*] [**time-range** *time-range-name*] [**log**]
Assign an access list to an interface	**ip access-group** *number* {**in** | **out**}
Create a reflexive access list	**ip access-list extended** {*access-list-number* | *access-list-name*} [*sequence-number*] {**deny | permit**} *protocol source source-wildcard destination destination-wildcard* reflect *name* and **evaluate**

Table 8-7 *Command Reference*

Task	Command Syntax
Create a time-based access list	**time-range** *time-range-name* **periodic** *days-of-the-week hh:mm* to [days-of-the-week] *hh:mm* **absolute** [**start** *time date*] [**end** *time date*] **access-list** *access-list-number protocol source source-wildcard destination destination-wildcard* [**time-range** *time-range-name*] or **ip access-list extended** {*access-list-number* | *access-list-name*} [*sequence-number*] {**deny | permit**} *protocol source source-wildcard destination destination-wildcard* [**time-range** *time-range-name*]
Load a specific PHDF file	**load protocol** *location:filename*
Load a specific TCDF file	**load classification** *location:filename*
Create an FPM class map	**class-map type** [**stack** | **access-control**] [**match-all** | **match-any**] *class-map-name*
Match specific traffic to classify within a class map	**match field** *protocol protocol-field* [**eq** | **neq** | **gt** | **lt** | **range** range] *value* **next** *next-protocol* **match start** [**l2-start** | **l3-start**] **offset** *offset* **size** *size* [**eq** | **neq** | **gt** | **lt** | **range** *range*] *value*
Create an FPM policy map	**policy-map type access-control** *policy-map-name*
Associate a class map with a policy map	**class** *class-name*
Specify a policy map action	**drop** or **service-policy** *policy-map-name*
Assign a policy map to an interface	**service-policy type access-control** [**input** | **output**] *policy-map-name*
Create a user-defined NetFlow flow record format	**flow record** *flow-record-name*
Specify NetFlow key fields	**match** [**ipv4** | **ipv6** | **datalink** | **routing** |**flow** | **interface**} *options*

Table 8-7 *Command Reference*

Task	**Command Syntax**
Specify NetFlow nonkey fields	**collect** [**counter** | **ipv4** | **ipv6** | **datalink** | **routing** |**flow** | **interface** | **timestamp**] *options*
Configure a NetFlow flow monitor	**flow monitor** *flow-monitor-name*
Specify a NetFlow record format	**record** [*flow-record-name* | **netflow** | **netflow-original**] {**ipv4** | **ipv6**} {**original-input** | **original-output**}
Configure a NetFlow flow exporter	**flow exporter** *flow-exporter-name*
Specify a NetFlow flow exporter server	**destination** [*hostname* | *ip-address*]
Specify a NetFlow flow exporter server port	**transport udp** *port*
Configure a NetFlow flow exporter with a flow monitor	**exporter** *flow-exporter-name*
Configure a NetFlow flow sampler	**sampler** *sampler-name*
Specify a NetFlow flow sampler mode	**mode** {**deterministic** | **random**} **1 out-of** *window-size*
Associate a NetFlow flow monitor with an interface	**ip flow monitor** *flow-monitor-name* {**sampler** *sampler-name*} [**input** | **output**]
Enable CEF	**ip cef** {**distributed**}
Configure Unicast RPF on a specific interface	**ip verify unicast source reachable-via** [**rx** | **any**] {*access-list*}
Display the contents of all current access lists	**show access-list** [*access-list-number* | *access-list-name*}
Display the contents of all current IP access lists	**show ip access-list** [*access-list-number* | *access-list-name*}
Display which specific PHDFs are loaded and which fields are supported	**show protocols phdf** *phdf-name*
Display the current traffic classes configured and their matching criteria	**show class-map type** [**stack** | **access-control**]
Display the current traffic policies	**show policy-map type access-control** {**interface** *interface*}
Display NetFlow flow monitor configuration	**show flow monitor**
Display NetFlow flow monitor interface configuration	**show flow interface** *interface*

Table 8-7 *Command Reference*

Task	Command Syntax
Display NetFlow flow exporter configuration	**show flow exporter**
Display NetFlow cache	**show flow monitor name** flow-monitor-name **cache format** [**csv** | **record** | **table**]
Display NetFlow sampler configuration	**show sampler**
Display Unicast RPF status	**show cef interface** *interface*
Display global Unicast RPF packet count	**show ip traffic**
Display the number of interface Unicast RPF packet drops	**show ip interface** *interface*

Fill in the Blanks

1. There is a(n) _____ at the end of each access list.
2. An extended access list can use the number ranges of _____ and _____.
3. The wildcard mask that would be used with a subnet mask of 255.255.255.192 would be _____.
4. When assigning reflexive access lists to an interface, they are typically placed _____ on an interface facing away from the internal network or _____ on an interface facing toward the internal network.
5. Both PHDF and TCDF are formatted using _____.
6. When using FPM, traffic can be classified using _____ files or using the _____.
7. FPM is only able to inspect _____ unicast packets.
8. _____ fields are used by NetFlow to identify specific flows.
9. Unicast RPF can operate in _____ or _____ mode.
10. When configuring Unicast RPF, the first thing that must be configured is _____.

This page intentionally left blank

Index