cisco

Official Cert Guide

Learn, prepare, and practice for exam success

CCNA Security 640-554

- Master **CCNA Security 640-554** 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

KEITH BARKER, CCIE® No. 6783
SCOTT MORRIS, CCIE No. 4713
KEVIN WALLACE, CCIE No. 7945
MICHAEL WATKINS

ciscopress.com

CCNA Security 640-554

Official Cert Guide

Keith Barker, CCIE No. 6783

Scott Morris, CCIE No. 4713

Cisco Press

800 East 96th Street

Indianapolis, IN 46240

CCNA Security 640-554 Official Cert Guide

Keith Barker, CCIE No. 6783

Scott Morris, CCIE No. 4713

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

Printed in the United States of America 1 2 3 4 5 6 7 8 9 0

First Printing July 2012

Library of Congress Cataloging-in-Publication data is on file.

ISBN13: 978-1-58720-446-3

ISBN: 1-58720-446-0

Warning and Disclaimer

This book is designed to provide information about selected topics for the CCNA Security 640-554 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.

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 about 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.

Corporate and Government Sales

Cisco Press offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact: U.S. Corporate and Government Sales 1-800-382-3419 corpsales@pearsontechgroup.com

For sales outside of the U.S., please contact: International Sales international@pearsoned.com

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.

Publisher: Paul Boger

Associate Publisher: Dave Dusthimer

Executive Editor: Brett Bartow

Managing Editor: Sandra Schroeder

Senior Project Editor: Tonya Simpson

Indexer: Heather McNeill

Book Designer: Gary Adair

Manager, Global Certification: Erik Ullanderson

Business Operation Manager, Cisco Press: Anand Sundaram

Technical Editors: Brandon Anastasoff and David Burns

Development Editor: Andrew Cupp

Editorial Assistant: Vanessa Evans

Copy Editor: Keith Cline

Compositor: Mark Shirar

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

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

Keith Barker, CCIE No. 6783 (R&S and Security), is a 27-year veteran of the networking industry. He currently works as a network engineer and trainer for Copper River IT. His past experience includes EDS, Blue Cross, Paramount Pictures, and KnowledgeNet, and he has delivered CCIE-level training over the past several years. As part of the original set of Cisco VIPs for the Cisco Learning Network, he continues to give back to the community in many ways. He is CISSP and CCSI certified, loves to teach, and keeps many of his video tutorials at http://www.youtube.com/keith6783. He can be reached at Keith.Barker@CopperRiverIT.com or by visiting http://www.CopperRiverIT.com.

Scott Morris, CCIE No. 4713 (R&S, ISP/Dial, Security, and Service Provider), has more than 25 years in the industry. He also has CCDE and myriad other certifications, including nine expert-level certifications spread over four major vendors. Having traveled the world consulting for various enterprise and service provider companies, Scott currently works at Copper River IT as the chief technologist. He, too, has delivered CCIE-level training and technology training for Cisco Systems and other technology vendors. Having spent a "past life" (early career) as a photojournalist, he brings interesting points of view from entering the IT industry from the ground up. As part of the original set of Cisco VIPs for the Cisco Learning Network, he continues to give back to the community in many ways. He can be reached at smorris@CopperRiverIT.com or by visiting http://www.CopperRiverIT.com.

About the Contributing Authors

Kevin Wallace, CCIE No. 7945, is a certified Cisco instructor holding multiple Cisco certifications, including CCSP, CCVP, CCNP, and CCDP. With Cisco experience dating back to 1989, Kevin has been a network design specialist for the Walt Disney World Resort, a senior technical instructor for SkillSoft/Thomson NETg/KnowledgeNet, and a network manager for Eastern Kentucky University. Kevin holds a bachelor of science degree in electrical engineering from the University of Kentucky. Kevin has also authored or co-authored multiple books for Cisco Press, including: *CCNP TSHOOT 642-832 Cert Kit*, *CCNP TSHOOT 642-832 Official Certification Guide*, *CCNP ROUTE 642-902 Cert Kit*, and *CCNP Routing and Switching Official Certification Library*, all of which target the current CCNP certification.

Michael Watkins, CCNA/CCNP/CCVP/CCSP, is a full-time senior technical instructor with SkillSoft. With 12 years of network management, training, and consulting experience, Michael has worked with organizations such as Kraft Foods, Johnson and Johnson, Raytheon, and the United States Air Force to help them implement and learn the latest network technologies. In addition to holding over more than 20 industry certifications in the areas of networking and programming technologies, Michael holds a bachelor of arts degree from Wabash College.

About the Technical Editors

Brandon Anastasoff has been a systems engineer with Cisco Systems since October 2007, when he moved from a lead network architect role in a major newspaper-publishing firm. He has spent more than 20 years in the industry, focusing on security for the past 10 and obtaining certifications inside and outside of Cisco, with his CISSP, CCSP, and most recently, the Security CCIE. After studying in the United Kingdom, Brandon took a year off in Saudi Arabia to see what a real job would be like before proceeding to college, but found the lure of an income too irresistible and never went back for the degree. Brandon had to make a choice early in his career to either follow the art of computer animation or the up-and-coming PC networking boom, and he has never regretted the decision to enter networking. He moved from early versions of Windows and Macintosh operating systems through Novell's NetWare, and then moved more into the infrastructure side, focusing mostly on Cisco LAN/WAN equipment. After Y2K, the focus became more security oriented, and Brandon became familiar with virus and Trojan analysis and forensic investigations. Today, Brandon is glad to be where he is and enjoys talking about security whenever the opportunity presents itself.

David Burns has in-depth knowledge of routing and switching technologies, network security, and mobility. He is currently a systems engineering manager for Cisco covering various U.S. service provider accounts. In July 2008, Dave joined Cisco as a lead systems engineer in a number of areas, including Femtocell, Datacenter, MTSO, and Security Architectures working for a U.S.-based SP Mobility account. He came to Cisco from a large U.S.-based cable company where he was a senior network and security design engineer. Dave held various roles before joining Cisco during his 10-plus years in the industry, working in SP operations, SP engineering, SP architecture, enterprise IT, and U.S. military intelligence communications engineering. He holds various sales and industry/Cisco technical certifications, including the CISSP, CCSP, CCDP, and two associate-level certifications. Dave recently passed the CCIE Security Written, and is currently preparing for the CCIE Security Lab. Dave is a big advocate of knowledge transfer and sharing and has a passion for network technologies, especially as related to network security. Dave has been a speaker at Cisco Live on topics such as Femtocell (IP mobility) and IPS (security). Dave earned his Bachelor of Science degree in telecommunications engineering technology from Southern Polytechnic State University, Georgia, where he currently serves as a member of the Industry Advisory Board for the Computer & Electrical Engineering Technology School.

Dedications

From Keith:

To my parents for bringing me into this world, to my children for perpetuating this world, and to my wonderful wife, Jennifer, for making my current world a better place. I love you, Jennifer.

From Scott:

The variety of inspirations and muses that affect a person's life vary over time. Every one of them affects us in different ways to help shape or drive us to where we are today. I certainly enjoy all the influences that have helped to shape (or warp) me to where I currently am. To my friend and co-author Keith, for convincing me that this was a good idea and a lot of fun to do (and gently "reminding" me of that along the way). To my dear friend Amy (who is smarter than I am) for continuing to tell me that I need to get my CCIE Voice taken care of and prodding me along now and then, motivating me to be something more than what I am currently. To my dear friend Angela, who enjoys keeping me both sane and humble by poking holes in my plans and helping me make things even better while keeping my sense of humor intact. And to my two little girls, who help keep my perspective on the world both healthy and a little off-kilter.

Acknowledgments

We want to thank many people for helping us put this book together.

The Cisco Press team: Brett Bartow, the executive editor, was the catalyst for this project, coordinating the team and ensuring that sufficient resources were available for the completion of the book. Andrew Cupp, the development editor, has been invaluable in producing a high-quality manuscript. His great suggestions and keen eye caught some technical errors and really improved the presentation of the book. We would also like to thank Tonya Simpson and the production team for their excellent work in shepherding this book through the editorial process and nipping at our heels where necessary. Many thanks go to Keith Cline for going the extra mile during the copy edit.

The technical reviewers: We want to thank the technical reviewers of this book, Brandon Anastasoff and David Burns, for their thorough, detailed review and very valuable input.

Our families: Of course, this book would not have been possible without the constant understanding and patience of our families. They have lived through the long days and nights it took to complete this project, and have always been there to poke, prod, motivate, and inspire us. We thank you all.

Each other: Last, but not least, this book is a product of work by two co-workers and colleagues, who have worked together at three different companies over the past 5 years and still manage to stay friends, which made it even more of a pleasure to complete.

Contents at a Glance

Contents

Command Syntax Conventions

The conventions used to present command syntax in this book are the same conventions used in the IOS Command Reference. The Command Reference describes these conventions as follows:

- **Boldface** indicates commands and keywords that are entered literally as shown. In actual configuration examples and output (not general command syntax), boldface indicates commands that are manually input by the user (such as a **show** command).
- *Italic* indicates arguments for which you supply actual values.
- Vertical bars (|) separate alternative, mutually exclusive elements.
- Square brackets ([ ]) indicate an optional element.
- Braces ({ }) indicate a required choice.
- Braces within brackets ([{ }]) indicate a required choice within an optional element.

Introduction

Congratulations! If you are reading this, you have in your possession a powerful tool that can help you to

- Improve your awareness and knowledge of network security
- Increase your skill level related to the implementation of that security
- Prepare for the CCNA Security certification exam

When writing this book, it was done with you in mind, and together we will discover the critical ingredients that make up the recipe for a secure network and work through examples of how to implement these features. By focusing on both covering the objectives for the CCNA Security exam and integrating that with real-world best practices and examples, Scott Morris and I created this content with the intention of being your personal tour guides, as we take you on a journey through the world of network security.

The 640-554 *Implementing Cisco IOS Network Security (IINSv2)* exam is required for the CCNA Security certification. The prerequisite for CCNA Security is the CCNA Route/Switch certification (or any CCIE certification). The CCNA Security exam tests your knowledge of securing Cisco routers and switches and their associated networks, and this book prepares you for that exam. This book covers all the topics listed in Cisco's exam blueprint, and each chapter includes key topics and preparation tasks to assist you in mastering this information. The CD that accompanies this book also includes bonus videos to assist you in your journey toward becoming a CCNA in Security. Of course, the CD included with the printed book also includes several practice questions to help you prepare for the exam.

About the 640-554 Implementing Cisco IOS Network Security (IINSv2) Exam

Cisco's objective of the CCNA Security exam is to verify the candidate's understanding, implementation, and verification of security best practices on Cisco hardware and software. The focus points for the exam (which this book prepares you for) are as follows:

- **Cisco routers and switches**
  - Common threats, including blended threats, and how to mitigate them.
  - The lifecycle approach for a security policy
  - Understanding and implementing network foundation protection for the control, data, and management planes
  - Understanding, implementing, and verifying AAA (authentication, authorization, and accounting), including the details of TACACS+ and RADIUS
  - Understanding and implementing basic rules inside of Cisco Access Control Server (ACS) Version 5.x, including configuration of both ACS and a router for communications with each other

- Standard, extended, and named access control lists used for packet filtering and for the classification of traffic
- Understanding and implementing protection against Layer 2 attacks, including CAM table overflow attacks, and VLAN hopping

- **Cisco firewall technologies**
  - Understanding and describing the various methods for filtering implemented by firewalls, including stateful filtering. Compare and contrast the strengths and weaknesses of the various firewall technologies.
  - Understanding the methods that a firewall may use to implement Network Address Translation (NAT) and Port Address Translation (PAT).
  - Understanding, implementing, and interpreting a Zone-Based Firewall policy through Cisco Configuration Professional (CCP).
  - Understanding and describing the characteristics and defaults for interfaces, security levels, and traffic flows on the Adaptive Security Appliance (ASA).
  - Implementing and interpreting a firewall policy on an ASA through the GUI tool named the ASA Security Device Manager (ASDM).

- **Intrusion prevention systems**
  - Comparing and contrasting intrusion prevention systems (IPS) versus intrusion detection systems (IDS), including the pros and cons of each and the methods used by these systems for identifying malicious traffic
  - Describing the concepts involved with IPS included true/false positives/negatives
  - Configuring and verifying IOS-based IPS using CCP

- **VPN technologies**
  - Understanding and describing the building blocks used for virtual private networks (VPN) today, including the concepts of symmetrical, asymmetrical, encryption, hashing, Internet Key Exchange (IKE), public key infrastructure (PKI), authentication, Diffie-Hellman, certificate authorities, and so on
  - Implementing and verifying IPsec VPNs on IOS using CCP and the command-line interface (CLI)
  - Implementing and verifying Secure Sockets Layer (SSL) VPNs on the ASA firewall using ASDM

As you can see, it is an extensive list, but together we will not only address and learn each of these, but we will also have fun doing it.

You can take the exam at Pearson VUE testing centers. You can register with VUE at http://www.vue.com/cisco/.

640-554 IINSv2 Exam

Table I-1 lists the topics of the 640-554 IINSv2 exam and indicates the parts in the book where these topics are covered.

Table I-1 *640-554 CCNA Security (IINSv2) Exam Topics*

Exam Topic	Part
Common Security Threats	
Describe common security threats	I, II, III
Security and Cisco Routers	
Implement security on Cisco routers	II, III
Describe securing the control, data, and management plane	II
Describe Cisco Security Manager	II, III
Describe IPv4 to IPv6 transition	II
AAA on Cisco Devices	
Implement AAA (authentication, authorization, and accounting)	II
Describe TACACS+	II
Describe RADIUS	II
Describe AAA	II
Verify AAA functionality	II
IOS ACLs	
Describe standard, extended, and named IP IOS access control lists (ACLs) to filter packets	III
Describe considerations when building ACLs	III
Implement IP ACLs to mitigate threats in a network	III
Secure Network Management and Reporting	
Describe secure network management	II
Implement secure network management	II
Common Layer 2 Attacks	
Describe Layer 2 security using Cisco switches	II
Describe VLAN security	II
Implement VLANs and trunking	II
Implement spanning tree (securely)	II
Cisco Firewall Technologies	

Exam Topic	Part
Describe operational strengths and weaknesses of the different firewall technologies	III
Describe stateful firewalls	III
Describe the types of NAT used in firewall technologies	III
Implement zone-based policy firewall using CCP	III
Implement the Cisco Adaptive Security Appliance (ASA)	III
Implement Network Address Translation (NAT) and Port Address Translation (PAT)	III
Cisco IPS	
Describe Cisco Intrusion Prevention System (IPS) deployment considerations	III
Describe IPS technologies	III
Configure Cisco IOS IPS using CCP	III
VPN Technologies	
Describe the different methods used in cryptography	IV
Describe VPN technologies	IV
Describe the building blocks of IPsec	IV
Implement an IOS IPsec site-to-site VPN with pre-shared key authentication	IV
Verify VPN operations	IV
Implement Secure Sockets Layer (SSL) VPN using ASA Device Manager	IV

About the Implementing Cisco IOS Network Security (IINSv2) 640-554 Official Cert Guide

This book maps to the topic areas of the 640-554 exam and uses a number of features to help you understand the topics and prepare for your exam.

Objectives and Methods

This book uses several key methodologies to help you discover the exam topics for which you need more review, to help you fully understand and remember those details, and to help you prove to yourself that you have retained your knowledge of those topics. So, this book does not try to help you pass the exams only by memorization, but by truly learning and understanding the topics. This book is designed to assist you in the exam by using the following methods:

- Using a conversational style that reflects the fact that we wrote this book as if we made it just for you, as a friend, discussing the topics with you, one step at a time

- Helping you discover which exam topics you may want to invest more time studying, to really "get it"
- Providing explanations and information to fill in your knowledge gaps
- Supplying three bonus videos (on the CD) to reinforce some of the critical concepts and techniques that you have learned from in your study of this book
- Providing practice questions to assess your understanding of the topics

Book Features

To help you customize your study time using this book, the core chapters have several features that help you make the best use of your time:

- **"Do I Know This Already?" quiz:** Each chapter begins with a quiz that helps you determine how much time you need to spend studying that chapter.
- **Foundation Topics:** These are the core sections of each chapter. They explain the concepts for the topics in that chapter.
- **Exam Preparation Tasks:** After the "Foundation Topics" section of each chapter, the "Exam Preparation Tasks" section lists a series of study activities that you should do when you finish the chapter. Each chapter includes the activities that make the most sense for studying the topics in that chapter:
  - **Review All the Key Topics:** The Key Topic icon appears next to the most important items in the "Foundation Topics" section of the chapter. The "Review All the Key Topics" activity lists the key topics from the chapter, along with their page numbers. Although the contents of the entire chapter could be on the exam, you should definitely know the information listed in each key topic, so you should review these.
  - **Complete the Tables and Lists from Memory:** To help you memorize some lists of facts, many of the more important lists and tables from the chapter are included in a document on the CD. This document lists only partial information, allowing you to complete the table or list.
  - **Define Key Terms:** Although the exam is unlikely to ask a "define this term" type of question, the CCNA exams do require that you learn and know a lot of networking terminology. This section lists the most important terms from the chapter, asking you to write a short definition and compare your answer to the glossary at the end of the book.
  - **Command Reference to Check Your Memory:** Review important commands covered in the chapter.
- **CD-based practice exam:** The companion CD contains an exam engine that enables you to review practice exam questions. Use these to prepare with a sample exam and to pinpoint topics where you need more study.

How This Book Is Organized

This book contains 21 core chapters. Chapter 22 includes some preparation tips and suggestions for how to approach the exam. Each core chapter covers a subset of the topics on the CCNA Security exam. The core chapters are organized into parts. They cover the following topics:

Part I: Fundamentals of Network Security

- **Chapter 1, "Networking Security Concepts":** This chapter covers the need for and the building blocks of network and information security, threats to our networks today, and fundamental principles of secure network design.
- **Chapter 2, "Understanding Security Policies Using a Lifecycle Approach":** This chapter covers risk analysis and management and security policies.
- **Chapter 3, "Building a Security Strategy":** This chapter covers securing borderless networks and controlling and containing data loss.

Part II: Protecting the Network Infrastructure

- **Chapter 4, "Network Foundation Protection":** This chapter covers introduction to securing the network using the *network foundation protection (NFP)* approach, the management plane, the control plane, and the data plane.
- **Chapter 5, "Using Cisco Configuration Professional to Protect the Network Infrastructure":** This chapter covers introduction to Cisco Configuration Professional, CCP features and the GUI, setting up a new devices, CCP building blocks, and CCP audit features.
- **Chapter 6, "Securing the Management Plane on Cisco IOS Devices":** This chapter covers management traffic and how to make it more secure and the implementation of security measures to protect the management plane.
- **Chapter 7, "Implementing AAA Using IOS and the ACS Server":** This chapter covers the role of Cisco Secure ACS and the two primary protocols used with it, RADIUS and TACACS. It also covers configuration of a router to interoperate with an ACS server and configuration of the ACS server to interoperate with a router. The chapter also covers router tools to verify and troubleshoot router-to-ACS server interactions.
- **Chapter 8, "Securing Layer 2 Technologies":** This chapter covers VLANs and trunking fundamentals, spanning-tree fundamentals, and common Layer 2 threats and how to mitigate them.
- **Chapter 9, "Securing the Data Plane in IPv6":** This chapter covers IPv6 (basics, configuring, and developing a security plan for IPv6).

Part III: Mitigating and Controlling Threats

- **Chapter 10, "Planning a Threat Control Strategy":** This chapter covers the design considerations for threat mitigation and containment and the hardware, software, and services used to implement a secure network.

- **Chapter 11, "Using Access Control Lists for Threat Mitigation":** This chapter covers the benefits and fundamentals for *access control lists (ACL)*, implementing IPv4 ACLs as packet filters, and implementing IPv6 ACLs as packet filters.
- **Chapter 12, "Understanding Firewall Fundamentals":** This chapter covers firewall concepts and the technologies used by them, the function of *Network Address Translation (NAT)*, including its building blocks, and the guidelines and considerations for creating and deploying firewalls.
- **Chapter 13, "Implementing Cisco IOS Zone-Based Firewalls":** This chapter covers the operational and functional components of the IOS Zone-Based Firewall and how to configure and verify the IOS Zone-Based Firewall.
- **Chapter 14, "Configuring Basic Firewall Policies on Cisco ASA":** This chapter covers the *Adaptive Security Appliance (ASA)* family and features, ASA firewall fundamentals, and configuring the ASA.
- **Chapter 15, "Cisco IPS/IDS Fundamentals":** This chapter compares intrusion *prevention systems (IPS)* to *intrusion detection systems (IDS)* and covers how to identify malicious traffic on the network, manage signatures, and monitor and manage alarms and alerts.
- **Chapter 16, "Implementing IOS-Based IPS":** This chapter covers the features included in IOS-based IPS (in software) and installing the IPS feature, working with signatures in IOS-based IPS, and managing and monitoring IPS alarms.

Part IV: Using VPNs for Secure Connectivity

- **Chapter 17, "Fundamentals of VPN Technology":** This chapter covers what VPNs are and why we use them and the basic ingredients of cryptography.
- **Chapter 18, "Fundamentals of the Public Key Infrastructure":** This chapter covers the concepts, components, and operations of the *public key infrastructure (PKI)* and includes an example of putting the pieces of PKI to work.
- **Chapter 19, "Fundamentals of IP Security":** This chapter covers the concepts, components, and operations of IPsec and how to configure and verify IPsec.
- **Chapter 20, "Implementing IPsec Site-to-Site VPNs":** This chapter covers planning and preparing to implement an IPsec site-to-site VPN and implementing and verifying the IPsec site-to-site VPN.
- **Chapter 21, "Implementing SSL VPNs Using Cisco ASA":** This chapter covers the functions and use of SSL for VPNs, configuring SSL clientless VPN on the ASA, and configuring the full SSL AnyConnect VPN on the ASA.
- **Chapter 22, "Final Preparation":** This chapter identifies tools for final exam preparation and helps you develop an effective study plan.

Appendixes

- **Appendix A, "Answers to the 'Do I Know This Already?' Quizzes":** Includes the answers to all the questions from Chapters 1 through 21.

- **Appendix B, "CCNA Security 640-554 (IINSv2) Exam Updates":** This appendix provides instructions for finding updates to the exam and this book when and if they occur.

CD-Only Appendixes

- **Appendix C, "Memory Tables":** This CD-only appendix contains the key tables and lists from each chapter, with some of the contents removed. You can print this appendix and, as a memory exercise, complete the tables and lists. The goal is to help you memorize facts that can be useful on the exams. This appendix is available in PDF format on the CD; it is not in the printed book.
- **Appendix D, "Memory Tables Answer Key":** This CD-only appendix contains the answer key for the memory tables in Appendix C. This appendix is available in PDF format on the CD; it is not in the printed book.

Premium Edition eBook and Practice Test

This Cert Guide contains a special offer for a 70% discount off the companion CCNA Security 640-554 Official Cert Guide Premium Edition eBook and Practice Test. The Premium Edition combines an eBook version of the text with an enhanced Pearson IT Certification Practice Test. By purchasing the Premium Edition, you get access to two eBook versions of the text: a PDF version and an EPUB version for reading on your tablet, eReader, or mobile device. You also get an enhanced practice test that contains an additional two full practice tests of unique questions. In addition, all the practice test questions are linked to the PDF eBook, allowing you to get more detailed feedback on each question instantly. To take advantage of this offer, you will need the coupon code included on the paper in the CD sleeve. Just follow the purchasing instructions that accompany the code to download and start using your Premium Edition today!

This page intentionally left blank

This chapter covers the following subjects:

- Securing management traffic
- Implementing security measures to protect the management plane

CHAPTER 6

Securing the Management Plane on Cisco IOS Devices

Accessing and configuring Cisco devices is a common occurrence for an administrator. Malicious router management traffic from an unauthorized source can pose a security threat. For example, an attacker could compromise router security by intercepting login credentials (such as the username and password). This chapter introduces the concept of the *management plane* (which is a collection of protocols and access methods we use to configure, manage, and maintain a network device) and examines how to protect it.

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz helps you determine your level of knowledge of this chapter's topics before you begin. Table 6-1 details the major topics discussed in this chapter and their corresponding quiz questions.

Table 6-1 *"Do I Know This Already?" Section-to-Question Mapping*

Foundation Topics Section	Questions
Securing Management Traffic	1–4, 6
Implementing Security Measures to Protect the Management Plane	5, 7–10

1. Which one of the following follows best practices for a secure password?

a. ABC123!

b. SlE3peR1#

c. tough-passfraze

d. InterEstIng-PaSsWoRd

2. When you connect for the first time to the console port on a new router, which privilege level are you using initially when presented with the command-line interface?

   a. 0

   b. 1

   c. 15

   d. 16

3. Which of the following is *not* impacted by a default login authentication method list?

   a. AUX line

   b. HDLC interface

   c. Vty line

   d. Console line

4. You are trying to configure a method list, and your syntax is correct, but the command is not being accepted. Which of the following might cause this failure? (Choose all that apply.)

   a. Incorrect privilege level

   b. AAA not enabled

   c. Wrong mode

   d. Not allowed by the view

5. Cisco recommends which version of Simple Network Management Protocol (SNMP) on your network if you need it?

   a. Version 1

   b. Version 2

   c. Version 3

   d. Version 4

6. How can you implement role-based access control (RBAC)? (Choose all that apply.)

   a. Provide the password for a custom privilege level to users in a given role

   b. Associate user accounts with specific views

   c. Use access lists to specify which devices can connect remotely

   d. Use AAA to authorize specific users for specific sets of permissions

7. Which of the following indirectly requires the administrator to configure a host name?

   a. Telnet

   b. HTTP

   c. HTTPS

   d. SSH

8. What are the two primary benefits of using NTP along with a syslog server? (Choose all that apply.)

   a. Correlation of syslog messages from multiple different devices

   b. Grouping of syslog messages into summary messages

   c. Synchronization in the sending of syslog messages to avoid congestion

   d. Accurate accounting of when a syslog message occurred

9. Which of the following commands result in a secure bootset? (Choose all that apply.)

   a. **secure boot-set**

   b. **secure boot-config**

   c. **secure boot-files**

   d. **secure boot-image**

10. What is a difference between a default and named method list?

    a. A default method list can contain up to four methods.

    b. A named method list can contain up to four methods.

    c. A default method list must be assigned to an interface or line.

    d. A named method list must be assigned to an interface or line.

Foundation Topics

Securing Management Traffic

It is tricky to fix a problem if you are unaware of the problem. So, this first section starts by classifying and describing management traffic and identifying some of the vulnerabilities that exist. It also identifies some concepts that can help you to protect that traffic. This chapter then provides implementation examples of the concepts discussed earlier.

What Is Management Traffic and the Management Plane?

When you first get a new router or switch, you connect to it for management using a blue rollover cable that connects from your computer to the console port of that router or switch. This is your first exposure to the concept of management traffic. By default, when you connect to a console port you are not prompted for a username or any kind of password. By requiring a username or password, you are taking the first steps toward improving what is called the *management plane* on this router or switch.

The management plane includes not only configuration of a system, but also who may access a system and what they are allowed to do while they are logged in. The management plane also includes messages to or from a Cisco router or switch that is used to maintain or report on the current status of the device, such as a management protocol like *Simple Network Management Protocol (SNMP)*.

Beyond the Blue Rollover Cable

Using the blue rollover cable directly connected to the console port is fairly safe. Unfortunately, it is not very convenient to require the use of a console port when you are trying to manage several devices that are located in different buildings, or on different floors of the same building. A common solution to this problem is to configure the device with an IP address that you can then use to connect to that device remotely. It is at this moment that the security risk goes up. Because you are connecting over IP, it might be possible for an unauthorized person to also connect remotely. The management plane, if it were secure, would enable you to control who may connect to manage the box, when they may connect, what they may do, and report on anything that they did. At the same time, you want to ensure that all the packets that go between the device being managed and the computer where the administrator is sitting are encrypted so that anyone who potentially may capture the individual packets while going through the network could not interpret the contents of the packets (which might contain sensitive information about the configuration or passwords used for access).

Management Plane Best Practices

When implementing a network, remember the following best practices. Each one, when implemented, improves the security posture of the management plane for your network:

- **Strong passwords:** Make passwords very difficult to break. Whenever you use passwords, make them complex and difficult to guess. An attacker can break a password in several ways, including a dictionary and/or a brute force attack. A dictionary attack automates the process of attempting to log in as the user, running through a long list of words (potential passwords); when one attempt fails, the attack just tries the next one (and so on). A brute-force attack doesn't use a list of words, but rather tries thousands or millions of possible character strings trying to find a password match (modifying its guesses progressively if it incorrectly guesses the password or stops before it reaches the boundary set by the attacker regarding how many characters to guess, with every possible character combination being tried.). A tough password takes longer to break than a simple password.
- **User authentication and AAA:** Require administrators to authenticate using usernames and passwords. This is much better than just requiring a password and not knowing exactly who the user is. To require authentication using usernames and passwords, you can use a method *authentication, authorization, and accounting (AAA)*. Using this, you can control which administrators are allowed to connect to which devices and what they can do while they are there, and you can create an audit trail (accounting records) to document what they actually did while they were logged in.
- **Role-based access control (RBAC):** Not every administrator needs full access to every device, and you can control this through AAA and custom privilege levels/parser views. For example, if there are junior administrators, you might want to create a group that has limited permissions. You could assign users who are junior administrators to that group; they then inherit just those permissions. This is one example of using RBAC. Another example of RBAC is creating a custom privilege level and assigning user accounts to that level. Regardless of how much access an administrator has, a change management plan for approving, communicating, and tracking configuration changes should be in place and used before changes are made.
- **Encrypted management protocols:** When using either in-band or out-of-band management, encrypted communications should be used, such as *Secure Shell (SSH)* or *Hypertext Transfer Protocol Secure (HTTPS)*. *Out-of-band (OOB)* management implies that there is a completely separate network just for management protocols and a different network for end users and their traffic. In-band management is when the packets used by your management protocols may intermingle with the user packets (considered less secure than OOB). Whether in-band or OOB, if a plaintext management protocol must be used, such as Telnet or HTTP, use it in combination with a *virtual private network (VPN)* tunnel that can encrypt and protect the contents of the packets being used for management.

- **Logging:** Logging is a way to create an audit trail. Logging includes not only what administrators have changed or done, but also system events that are generated by the router or switch because of some problem that has occurred or some threshold that has been reached. Determine the most important information to log, and identify logging levels to use. A logging level simply specifies how much detail to include in logging messages, and may also indicate that some less-serious logging messages do not need to be logged. Because the log messages may include sensitive information, the storage of the logs and the transmission of the logs should be protected to prevent tampering or damage. Allocate sufficient storage capacity for anticipated logging demands. Logging may be done in many different ways, and your logging information may originate from many different sources, including messages that are automatically generated by the router or switch and sent to a syslog server. A syslog server is a computer that is set up to receive and store syslog messages generated from network devices. If SNMP is used, preferably use Version 3 because of its authentication and encryption capabilities. You can use SNMP to change information on a router or switch, and you can also use it to retrieve information from the router or switch. An *SNMP trap* is a message generated by the router or switch to alert the manager or management station of some event.
- **Network Time Protocol (NTP):** Use NTP to synchronize the clocks on network devices so that any logging that includes time stamps may be easily correlated. Preferably, use NTP Version 3, to leverage its ability to provide authentication for time updates. This becomes very important to correlate logs between devices in case there is ever a breach and you need to reconstruct (or prove in a court of law) what occurred.
- **Secure system files:** Make it difficult to delete, whether accidentally or on purpose, the startup configuration files and the IOS images that are on the file systems of the local routers and switches. You can do so by using built-in IOS features discussed later in this chapter.

Note Even though OOB management is usually preferred over in-band management, some management applications benefit from in-band management. For example, consider a network management application that checks the reachability of various hosts and subnets. To check this reachability, an application might send a series of pings to a remote IP address, or check the availability of various Layer 4 services on a remote host. To perform these "availability" checks, the network management application needs to send traffic across a production data network. Also, in-band network management often offers a more economic solution for smaller networks. Even if using in-band management, it should be a separate subnet/VLAN, and one that only a select few people/devices have access to get to. This reduces your footprint for possible attack vectors.

Password Recommendations

Using passwords is one way to provide access. Using passwords alone is not as good as requiring a user ID or login name associated with the password for a user.

Here are some guidelines for password creation:

- It is best to have a minimum of eight characters for a password; bigger is better. This rule can be enforced by the local router if you are storing usernames and passwords on the router in the running config. The command **security passwords min-length** followed by the minimum password length enforces this rule on new passwords that are created, including the enable secret and line passwords on the vty, AUX, and console 0. Preexisting passwords will still operate even if they are less than the new minimum specified by the command.
- Passwords can include any alphanumeric character, a mix of uppercase and lowercase characters, and symbols and spaces. As a general security rule, passwords should not use words that may be found in a dictionary, because they are easier to break. Leading spaces in a password are ignored, but any subsequent spaces, including in the middle or at the end of a password, literally become part of that password and are generally a good idea. Another good practice is using special characters or even two different words (that are not usually associated with each other) as a passphrase when combined together. Caution should be used to not require such a complex password that the user must write it down to remember it, which increases the chance of it becoming compromised.
- Passwords in a perfect environment should be fairly complex, and should be changed periodically. The frequency of requiring a change in passwords depends on your security policy. Passwords changed often are less likely to be compromised.
- From a mathematical perspective, consider how many possibilities someone would need to try to guess a password. If only capital letters are used, you have 26 possibilities for each character. If your password is one character long, that is 261, or 26 possible variants. If you have a two-character password, that is 262, or 676 possible variants. If you start using uppercase (26) and lowercase (26), numerals (10), and basic special characters (32), your starting set becomes 94 possible variants per character. Even if we look at using an eight-character password, that is 948 or 6,095,689,385,410,816 (6.1 quadrillion) possibilities.

Using AAA to Verify Users

Unauthorized user access to a network creates the potential for network intruders to gain information or cause harm or both. Authorized users need access to their network resources, and network administrators need access to the network devices to configure and manage them. AAA offers a solution for both. In a nutshell, the goal of AAA is to identify who users are before giving them any kind of access to the network, and once they are identified, only give them access to the part they are authorized to use, see, or manage. AAA can create an audit trail that identifies exactly who did what and when

they did it. That is the spirit of AAA. User accounts may be kept on the local database or on a remote server. The *local database* is a fancy way of referring to user accounts that are created on the local router and are part of the running configuration.

AAA Components

Providing network and administrative access in a Cisco environment—regardless of whether it involves administrators managing the network or users getting access through network resources—is based on a modular architecture composed of the following three functional components:

- **Authentication:** Authentication is the process by which individuals prove that they are who they claim to be. The network environment has a variety of mechanisms for providing authentication, including the use of a username and password, token cards, and challenge and response. A common use is authenticating an administrator's access to a router console port, auxiliary port, or vty lines. An analogy is a bank asking you to prove that you are who you say you are before allowing you to make a transaction. As an administrator, you can control how a user is authenticated. Choices include referring to the local running configuration on the router to look for the username, going to an external server that holds the username and password information, and other methods. To specify the method to use, you create an authentication "method list" that specifies how to authenticate the user. There can be custom named method lists or default method lists. Examples of each are shown later in this chapter.
- **Authorization:** After the user or administrator has been authenticated, authorization can be used to determine which resources the user or administrator is allowed to access, and which operations may be performed. In the case of the average user, this might determine what hours that user is allowed on the network. In the case of an administrator, it could control what the administrator is allowed to look at or modify. An analogy is a bank (after having already authenticated who you are) determining whether you are authorized to withdraw some amount of money (probably based on your balance in your account at the bank). You can create authorization method lists to specify how to authorize users on the network.
- **Accounting and auditing:** After being authenticated and possibly authorized, the user or administrator begins to access the network. It is the role of accounting and auditing to record what the user or administrator actually does with this access, what he accesses, and how long he accesses it. This is also known as *creating an audit trail*. An analogy is a bank documenting and debiting your account for the money you withdraw. You can create and assign accounting method lists to control what is accounted for and where the accounting records will be sent.

Options for Storing Usernames, Passwords, and Access Rules

Cisco provides many ways to implement AAA services for Cisco devices, many of which use a centralized service to keep usernames, passwords, and configured rules about who can access which resources. Over the years, there have been many names and access methods

associated with the central server, including calling it an authentication server, AAA server, ACS server, TACACS server, or RADIUS server. These all refer to the same type of function: a server that contains usernames, passwords, and rules about what may be accessed. A router or switch acts like a client to this server and can send requests to the server to verify the credentials of an administrator or user who is trying to access a local router or switch. The following list describes a few of these centralized server types:

- **Cisco Secure ACS Solution Engine:** This is a dedicated server that contains the usernames, their passwords, and other information about what users are allowed to access and when. In the past, this was sold as a server appliance with the *Access Control Server (ACS)* software preinstalled. A router or switch becomes a client to the server. The router can be configured to require authentication from a user or administrator before providing access, and the router sends this request to the ACS server and lets the ACS server make the decision about allowing the user or administrator to continue. The protocol used between the router and the ACS server is normally TACACS+ if you are authenticating an administrator who is seeking command-line access. The protocol used between the router and the ACS server is normally RADIUS if you are authenticating an end user for network access. These are not hard-and-fast rules, and you can use either of the two protocols for similar features in many cases.
- **Cisco Secure ACS for Windows Server:** This software package may be used for user and administrator authentication. AAA services on the router or *network access server (NAS)* contact an external Cisco Secure ACS (running on a Microsoft Windows system). This is an older flavor of ACS, but may still be relevant to the certification exams.
- **Current flavors of ACS functionality:** The most common way that ACS services are implemented today is through a virtual machine running on some flavor of VMware. Another up-and-coming service to support similar services to ACS is called the Cisco *Identity Services Engine (ISE)*, which can be bundled in a single physical or logical device or appliance.
- **Self-contained AAA:** AAA services may be self-contained in the router itself. Implemented in this fashion, this form of authentication and authorization is also known as *local* authentication and authorization. The database that contains the usernames and passwords is the running configuration of the router or IOS device, and from a AAA perspective is referred to as the *local database* on the router. So, if you create a user locally on the router, you can also say that you created a user in the local database of the router. It is the same thing. In this case, because the router is acting as its own AAA server, you do not use TACACS+ or RADIUS as a protocol to connect to a remote ACS server, because you are not using an ACS server.

Authorizing VPN Users

One common implementation of AAA is its use in authenticating users accessing the corporate LAN through a remote-access IPsec VPN.

Let's see how authentication and authorization applies to users who are trying to access our network through a VPN. The first step is to authenticate users to find out who they are, and after we find out who they are, we can then control what they are authorized for. For example, if a user connects via a VPN, that user may or may not be allowed access to certain portions of the network based on who the user is. This type of access is sometimes called *packet mode*, as in a user attempting to send packets through the network instead of trying to get a *command-line interface (CLI)* like an administrator would. A user connecting over a dial-up connection (older technology) could very likely be authenticated via a PPP connection using the same concepts. In either case, we authenticate the users by asking for their username and password, and then check the rules to see what they are authorized to access. If we use the remote *Access Control Server (ACS)* server for the authentication and authorization for an end user, we would very likely use the RADIUS protocol between the router and the AAA server.

AAA access control is supported using either a local username-password database or through a remote server (such as an ACS server). To provide access to a small group of network users, or as a backup in case the ACS server cannot be reached, a local security database can be configured in the router using the **username** command.

Router Access Authentication

Note that we must choose authentication first if we want to also use authorization for a user or administrator. We cannot choose authorization for a user without knowing who that user is through authentication first.

Typically, if we authenticate an administrator, we also authorize that administrator for what we want to allow him to do. Administrators traditionally are going to need access to the CLI. When an administrator is at the CLI, that interface is provided by something called an EXEC shell. If we want to authorize the router to provide this CLI, that is a perfect example of using AAA to first authenticate the user (in this case, the administrator) and then authorize that user to get a CLI prompt (the EXEC shell) and even place the administrator at the correct privilege level. This type of access (CLI) could also be referred to as *character mode*. Simply think of an administrator at a CLI typing in characters to assist you in remembering that this is "character" mode. With the administrator, we would very likely authenticate his login request and authorize that administrator to use an EXEC shell. If we were using a remote ACS server for this authentication and authorization of an administrator, we would very likely use TACACS+ (between the router and the ACS server) because it has the most granular control, compared with RADIUS, which is the alternative. TACACS+ and RADIUS are both discussed in another chapter of this book in greater detail.

Table 6-2 identifies some of the terms that refer to the type of access and the likely protocols used between the router acting as a client and the ACS server acting as the AAA server.

Table 6-2 *AAA Components to Secure Administrative and Remote LAN Access*

Access Type Mode	Mode	Where These Are Likely to Be Used	AAA Command Element
Remote administrative access Usually TACACS+ between the router and the ACS	Character (line or EXEC mode)	Lines: vty, AUX console, and tty	**login**, **enable**, **exec**
Remote network access end users Usually RADIUS between the router and the ACS	Packet (interface mode) such as an interface with PPP requiring authentication	Interfaces: async, group-async, BRI, PRI, Other functionality: VPN user authentication	**ppp**, **network**, **vpn groups**

The AAA Method List

To make implementing AAA modular, we can specify individual lists of ways we want to authenticate, authorize, and account for the users. To do this, we create a *method list* that defines what resource will be used (such as the local database, an ACS server via TACACS+ protocol or an ACS server via RADIUS protocol, and so forth). To save time, we can create a default list or custom lists. We can create method lists that define the authentication methods to use, authorization method lists that define which authorization methods to use, and accounting method lists that specify which accounting method lists to use. A default list, if created, applies to the entire router or switch. A custom list, to be applied, must be both created and then specifically referenced in line or interface configuration mode. You can apply a custom list over and over again in multiple lines or interfaces. The type of the method list may be authentication, authorization, or accounting.

The syntax for a method list is as follows:

```
aaa type {default | list-name} method-1 [method-2 method-3 method-4]
```

The commands for a method list, along with their descriptions, are shown in Table 6-3.

Table 6-3 *Method List Options*

Command Element	Description
type	Identifies the type of list being created. Relevant options are **authentication**, **authorization**, or **accounting**.
default	Specifies the default list of methods to be used based on the methods that follow this argument. If you use the keyword **default**, a custom name is not used.

Command Element	Description
list-name	Used to create a custom method list. This is the name of this list, and is used when this list is applied to a line, such as to vty lines 0–4.
method	At least one method must be specified. To use the local user database, use the **local** keyword. A single list can contain up to 4 methods, which are tried in order, from left to right. In the case of an authentication method list, methods include the following: **enable:** The enable password is used for authentication. This might be an excellent choice as the last method in a method list. This way, if the previous methods are not available (such as the AAA server, which might be down or not configured), the router times out on the first methods and eventually prompts the user for the enable secret as a last resort. **krb5:** Kerberos 5 is used for authentication. **krb5-telnet:** Kerberos 5 Telnet authentication protocol is used when using Telnet to connect to the router. **line:** The line password (the one configured with the password command, on the individual line) is used for authentication. **local:** The local username database (running config) is used for authentication. **local-case:** Requires case-sensitive local username authentication. **none:** No authentication is used. **group radius:** A RADIUS server (or servers) is used for authentication. **group tacacs+:** A TACACS+ server (or servers) is used for authentication. **group** *group-name*: Uses either a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

Role-Based Access Control

The concept of *role-based access control (RBAC)* is to create a set of permissions or limited access and assign that set of permissions to users or groups. Those permissions are used by individuals for their given roles, such as a role of administrator or a role of a help desk person and so on There are different ways to implement RBAC, including creating custom privilege levels and creating parser views (coming up later in this section). In either case, the custom level or view can be assigned the permissions needed for a specific

function or role, and then users can use those custom privilege levels or parser views to carry out their job responsibilities on the network, without being given full access to all configuration options.

Custom Privilege Levels

When you first connect to a console port on the router, you are placed into user mode. User mode is really privilege level 1. This is represented by a prompt that ends with **>**. When you move into privileged mode by typing the **enable** command, you are really moving into privilege level 15. A user at privilege level 15 has access and can issue all the commands that are attached to or associated with level 15 and below. Nearly all the configuration commands, and the commands that get us into configuration mode, are associated by default with privilege level 15.

By creating custom privilege levels (somewhere between levels 2 and 14, inclusive), and assigning commands that are normally associated with privilege level 15 to this new level, you can give this subset of new commands to the individual who either logs in at this custom level or to the user who logs in with a user account that has been assigned to that level.

Limiting the Administrator by Assigning a View

Working with individual commands and assigning them to custom privilege levels is tedious at best, and it is for that reason that method is not used very often. So, what can be done if we need users to have a subset of commands available to them, but not all of them? In an earlier chapter, we looked at how *Cisco Configuration Professional (CCP)* could restrict the visibility of the features in the navigation pane by using user profiles. This technique, however, did not protect the router against a user connecting with Telnet or SSH, and if that user had level 15 permissions, the router would once again be unprotected at the CLI.

A solution to this is to use *parser views*, also referred to as simply a *view*. You can create a view and associate it with a subset of commands. When the user logs in using this view, that same user is restricted to only being able to use the commands that are part of his current view. You can also associate multiple users with a single view.

Encrypted Management Protocols

It is not always practical to have console access to the Cisco devices you manage. There are several options for remote access via IP connectivity, and the most common is an application called Telnet. The problem with Telnet is that it uses plain text, and anyone who gets a copy of those packets can identify our usernames and passwords used for access and any other information that goes between administrator and the router being managed (over the management plane). One solution to this is to not use Telnet. If Telnet must be used, it should only be used out of band, or placed within a VPN tunnel for privacy, or both.

Secure Shell provides the same functionality as Telnet, in that it gives you a CLI to a router or switch; unlike Telnet, however, SSH encrypts all the packets used in the session. So, with SSH, if a packet is captured and viewed by an unauthorized individual, it will not have any meaning because the contents of each packet are encrypted and the attacker or unauthorized person will not have the keys or means to decrypt the information. The encryption provides the feature of confidentiality.

With security, bigger really is better. With SSH, Version 2 is bigger and better than Version 1. Either version, however, is better than the unencrypted Telnet protocol. When you type in **ip ssh version 2**, (to enable version 2), the device may respond with a Version "1.99" is active. This is a function of a server that runs 2.0 but also supports backward compatibility with older versions. For more information, see RFC4253, section 5.1. You should use SSH rather than Telnet whenever possible.

For GUI management tools such as CCP, use HTTPS rather than HTTP because it encrypts the session which provides confidentiality for the packets in that session.

Using Logging Files

I still recall an incident on a customer site when a database server had a failed disk and was running on its backup. It was like that for weeks until they noticed a log message. If a second failure had occurred, the results would have been catastrophic. Administrators *should*, on a regular basis, analyze logs, especially from their routers, in addition to logs from other network devices. Logging information can provide insight into the nature of an attack. Log information can be used for troubleshooting purposes. Viewing logs from multiple devices can provide event correlation information (that is, the relationship between events occurring on different systems). For proper correlation of events, accurate time stamps on those events are important. Accurate time can be implemented through *Network Time Protocol (NTP)*.

Cisco IOS devices can send log output to a variety of destinations, including the following:

- **Console:** A router's console port can send log messages to an attached terminal (such as your connected computer, running a terminal emulation program).
- **vty lines:** Virtual tty (vty) connections (used by SSH and Telnet connections) can also receive log information at a remote terminal (such as an SSH or Telnet client). However, the **terminal monitor** command should be issued to cause log messages to be seen by the user on that vty line.
- **Buffer:** When log messages are sent to a console or a vty line, those messages are not later available for detailed analysis. However, log messages can be stored in router memory. This "buffer" area can store messages up to the configured memory size, and then the messages are rotated out, with the first in being the first to be removed. When the router is rebooted, these messages in the buffer memory are lost.
- **SNMP server:** When configured as an SNMP device, a router or switch can generate log messages, in the form of SNMP traps and send them to an SNMP manager (server).

- **Syslog server:** A popular choice for storing log information is a syslog server, which is easily configured and can store a large volume of logs. Syslog messages can be directed to one or more syslog servers from the router or switch.

A syslog logging solution consists of two primary components: syslog servers and syslog clients. A syslog server receives and stores log messages sent from syslog clients such as routers and switches.

Not all syslog messages are created equal. Specifically, they have different levels of severity. Table 6-4 lists the eight levels of syslog messages. The higher the syslog level, the more detailed the logs. Keep in mind that more-detailed logs require a bit more storage space, and also consider that syslog messages are transmitted in clear text. Also consider that the higher levels of syslog logging consume higher amounts of CPU processing time. For this reason, take care when logging to the console at the debugging level.

Table 6-4 *Syslog Severity Levels*

Level	Name	Description
0	Emergencies	System is unusable.
1	Alerts	Immediate action needed.
2	Critical	Critical conditions.
3	Errors	Error conditions.
4	Warnings	Warning conditions.
5	Notifications	Normal, but significant conditions.
6	Informational	Informational messages.
7	Debugging	Highly detailed information based on current debugging that is turned on.

The syslog log entries contain time stamps, which are helpful in understanding how one log message relates to another. The log entries include severity level information in addition to the text of the syslog messages. Having synchronized time on the routers, and including time stamps in the syslog messages, makes correlation of the syslog messages from multiple devices more meaningful.

Understanding NTP

Network Time Protocol (NTP) uses UDP port 123, and it allows network devices to synchronize their time. Ideally, they would synchronize their time to a trusted time server. You can configure a Cisco router to act as a trusted NTP server for the local network, and in the same way, that trusted NTP server could turn around and be an NTP client to a trusted NTP server either on the Internet or reachable via network connectivity. NTP Version 3 supports cryptographic authentication between NTP devices, and for this reason its use is preferable over any earlier versions.

One benefit of having reliable synchronized time is that log files and messages generated by the router can be correlated. In fact, if we had 20 routers, and they were all reporting various messages and all had the same synchronized time, we could very easily correlate the events across all 20 routers if we looked at those messages on a common server. A common server that is often used is a syslog server.

Protecting Cisco IOS Files

Similar to the computers that we use every day, a router also uses an operating system. The Cisco operating system on the router is called *IOS*. When a router first boots, it performs a power-on self-test, and then looks for an image of IOS on the flash. After loading the IOS into RAM, the router then looks for its startup configuration. If for whatever reason an IOS image or the startup configuration cannot be found or loaded properly, the router will effectively be nonfunctional as far as the network is concerned.

To help protect a router from accidental or malicious tampering of the IOS or startup configuration, Cisco offers a resilient configuration feature. This feature maintains a secure working copy of the router IOS image and the startup configuration files at all times. Once enabled, the administrator cannot disable the features remotely (but can if connected directly on the console). The secure files are referred to as a *secure bootset*.

Implement Security Measures to Protect the Management Plane

The first section of this chapter covered some best practices to protect the management plane. With that in mind, you can now leverage what you have learned and look at some practical examples of implementing those best practices. It requires both the understanding and implementation of these best practices to secure your networks.

Implementing Strong Passwords

The privileged EXEC secret (the one used to move from user mode to privileged mode) should not match any other password that is used on the system. Many of the other passwords are stored in plain text (such as passwords on the vty lines). If an attacker discovers these other passwords, he might try to use them to get into privileged mode, and that is why the enable secret should be unique. Service password encryption scrambles any plaintext passwords as they are stored in the configuration. This is useful for preventing someone who is looking over your shoulder from reading a plaintext password that is displayed in the configuration on the screen. Any new plaintext passwords are also scrambled as they are stored in the router's configuration.

Example 6-1 shows the use of strong passwords.

Example 6-1 *Using Strong Passwords*

```
! Use the "secret" keyword instead of the "password" for users
! This will create a secured password in the configuration by default
! The secret is hashed using the MD5 algorithm as it is stored in the
! configuration
R1(config)# username admin secret CeyeSc01$24

! At a minimum, require a login and password for access to the console port
! Passwords on lines, including the console, are stored as plain text, by
! default, in the configuration
R1(config)# line console 0
R1(config-line)# password k4(1fmMsS1#
R1(config-line)# login
R1(config-line)# exit

! At a minimum, require a login and password for access to the VTY lines which
! is where remote users connect when using Telnet
! Passwords on lines, including the vty lines, are stored as plain text, by
! default, in the configuration
R1(config)# line vty 0 4
R1(config-line)# password 8wT1*eGP5@
R1(config-line)# login

! At a minimum, require a login and password for access to the AUX line
! and disable the EXEC shell if it will not be used
R1(config-line)# line aux 0
R1(config-line)# no exec
R1(config-line)# password 1wT1@ecP27
R1(config-line)# login
R1(config-line)# exit

! Before doing anything else, look at the information entered.
R1(config)# do show run | include username
username admin secret 5 $1$XJdX$9hqvG53z3lesP5BLOqggO.
R1(config)#
R1(config)# do show run | include password
no service password-encryption
 password k4(1fmMsS1#
 password 8wT1*eGP5@
 password 1wT1@ecP27
R1(config)#
```

```
! Notice that we can not determine the admin user's password, since
! it is automatically hashed using the MD5 algorithm because of using
! the secret command, however, we can still see all the other plain text
! passwords.

! Encrypt the plain text passwords so that someone reading the configuration
! won't know what the passwords are by simply looking at the configuration.
R1(config)# service password-encryption

! Verify that the plain text passwords configured are now scrambled due to the
! command "service password-encryption"
R1(config)# do show run | begin line
line con 0
 password 7 04505F4E5E2741631A2A5454
 login
line aux 0
 no exec
 login
 password 7 075E36781F291C0627405C
line vty 0 4
 password 7 065E18151D040C3E354232
 login
!
end
```

User Authentication with AAA

Example 6-2 shows the use of method lists, both named and default.

Example 6-2 *Enabling AAA Services and Working with Method Lists*

```
! Enable aaa features, if not already present in the running configuration
R1(config)# aaa new-model

! Identify a AAA server to be used, and the password it is expecting with
! requests from this router. This server would need to be reachable and
! configured as a TACACS+ server to support R1's requests
R1(config)# tacacs-server host 50.50.4.101
R1(config)# tacacs-server key ToUgHPaSsW0rD-1#7

! configure the default method list for the authentication of character
! mode login (where the user will have access to the CLI)
! This default method list, created below has two methods listed "local"
! and "enable"
```

```
! This list is specifying that the local database (running-config) will
!  be used first to look for the username.  If the username isn't in the
! running-config, then it will go to the second method in the list.
! The second method of "enable" says that if the user account isn't found
! in the running config, then to use the enable secret to login.
! This default list will apply to all SSH, Telnet, VTY, AUX and Console
! sessions unless there is another (different) custom method list that is
! created and directly applied to one of those lines.
R1(config)# aaa authentication login default local enable

! The next authentication method list is a custom authentication
! method list named MY-LIST-1.This method list says that the first attempt
! to verify the user's name and password should be done through one of the
! tacacs servers (we have only configured one so far), and then if that server
! doesn't respond, use the local database (running-config), and if the
! username isn't in the running configuration to then use the enable secret
! for access to the device.  Note: this method list is not used until
! applied to a line elsewhere in the configuration.
R1(config)# aaa authentication login MY-LIST-1 group tacacs local enable

! These next method lists are authorization method lists.
! We could create a default one as well, using the key
! word "default" instead of a name.    These custom method lists for
! authorization won't be used until we apply them
! elsewhere in the configuration, such as on a VTY line.
! The first method list called TAC1 is an authorization
! method list for all commands at user mode (called privilege level 1).
! The second method list called TAC15 is an
! authorization method list for commands at level 15 (privileged exec mode).
! If these method lists are applied to a line, such as the
! console or VTY lines, then before any commands
! are executed at user or privileged mode, the router will check
! with an ACS server that is one of the "tacacs+" servers, to see if the user
! is authorized to execute the command.    If a tacacs+ server isn't
! reachable, then the router will use its own database of users (the local
! database) to determine if the user trying to issue the command
! is at a high enough privilege level to execute the command.
R1(config)# aaa authorization commands 1 TAC1 group tacacs+ local
R1(config)# aaa authorization commands 15 TAC15 group tacacs+ local
```

```
! The next 2 method lists are accounting method lists that will record the
! commands issued at level 1 and 15 if the lists are applied to a line, and
! if an administrator connects to this device via that line.
! Accounting method lists can have multiple methods, but can't log to the
! local router.
R1(config)# aaa accounting commands 1 TAC-act1 start-stop group tacacs+
R1(config)# aaa accounting commands 15 TAC-act15 start-stop group tacacs+

! Creating a user with level 15 access on the local router is a good idea,
! in the event the ACS server can't be
! reached, and a backup method has been specified as the local database.
R1(config)# username admin privilege 15 secret 4Je7*1swEsf

! Applying the named method lists is what puts them in motion.
! By applying the method lists to the VTY lines
! any users connecting to these lines will be authenticated by the
! methods specified by the lists that are applied
! and also accounting will occur, based on the lists that are applied.
R1(config)# line vty 0 4
R1(config-line)# login authentication MY-LIST-1
R1(config-line)# authorization commands 1 TAC1
R1(config-line)# authorization commands 15 TAC15
R1(config-line)# accounting commands 1 TAC-act1
R1(config-line)# accounting commands 15 TAC-act15

! Note: on the console and AUX ports, the default list will be applied,
! due to no custom method list being applied
! directly to the console or AUX ports.
```

Using **debug** as a tool to verify what you think is happening is a good idea. In Example 6-3, we review and apply AAA and perform a **debug** verification.

Example 6-3 *Another Example of Creating and Applying a Custom Method List to vty Lines*

```
! Creating the method list, which has 3 methods.  First the local database
! (if the username exists in the configuration, and if not
! then the enable secret (if configured),  and if not then no
! authentication required
! (none)
R2(config)# aaa authentication login MY-AUTHEN-LIST-1 local enable none

! Applying the method list to the VTY lines 0-4
R2(config)# line vty 0 4
R2(config-line)# login authentication MY-AUTHEN-LIST-1
R2(config-line)# exit
```

```
! Creating a local username in the local database (running-config)
R2(config)# username bob secret ciscobob

! Setting the password required to move from user mode to privileged mode
R2(config)# enable secret ciscoenable
R2(config)# interface loopback 0

! Applying an IP address to test a local telnet to this same local router
! Not needed if the device has another local IP address that is in use
R2(config-if)# ip address 2.2.2.2 255.255.255.0
R2(config-if)# exit

! Enable logging so we can see results of the upcoming debug
R2(config)# logging buffered 7
R2(config)# end

! Enabling debug of aaa authentication, so we can see what the router is
! thinking regarding aaa authentication
R2# debug aaa authentication
AAA Authentication debugging is on

R2# clear log
Clear logging buffer [confirm]


! Telnet to our own address
R2# telnet 2.2.2.2
Trying 2.2.2.2 ... Open

User Access Verification

Username: bob
AAA/BIND(00000063): Bind i/f
AAA/AUTHEN/LOGIN (00000063): Pick method list 'MY-AUTHEN-LIST-1'
Password: [ciscobob] password not shown when typing it in

R2>

!  We can see that bob is connected via line vty 0, and that from the debug
!  the correct authentication list was used.
R2>who
    Line       User       Host(s)              Idle       Location
   0 con 0                 2.2.2.2              00:00:00
*  2 vty 0     bob        idle                 00:00:00 2.2.2.2
R2> exit
```

```
! If we exit back out, and remove all the users in the local database,
! (including bob) then the same login authentication will fail on the first
! method of the "local" database (no users there), and will go to the second
! method in the list, which is "enable", meaning use the enable secret if
! configured.

! As soon as I supply a username, the router discovers that there are no
! usernames
! configured in running configuration (at least none that match the user
! who is trying to
! login), and fails on the first method "local" in the list
! It then tries the next method of just caring about the enable secret.

R2# telnet 2.2.2.2
Trying 2.2.2.2 ... Open
User Access Verification

AAA/BIND(00000067): Bind i/f
AAA/AUTHEN/LOGIN (00000067): Pick method list 'MY-AUTHEN-LIST-1'

! Note: bertha in not a configured user in the local database on the router
Username: bertha
Password: [ciscoenable} not shown while typing.  This is the enable secret
 we set.
AAA/AUTHEN/ENABLE(00000067): Processing request action LOGIN
AAA/AUTHEN/ENABLE(00000067): Done status GET_PASSWORD

R2>
AAA/AUTHEN/ENABLE(00000067): Processing request action LOGIN
AAA/AUTHEN/ENABLE(00000067): Done status PASS
R2> exit

! One more method exists in the method list we applied to the VTY lines.
! If the local fails, and the enable secret fails (because neither of these
! is configured on the router, then the third method in the method list
! 'MY-AUTHEN-LIST-1' will be tried. The third method we specified is none,
! meaning no authentication required, come right in.  After removing the
! enable secret, we try once more.

R2# telnet 2.2.2.2
Trying 2.2.2.2 ... Open

User Access Verification
```

```
AAA/BIND(00000068): Bind i/f
AAA/AUTHEN/LOGIN (00000068): Pick method list 'MY-AUTHEN-LIST-1'
Username: doesn't matter
R2>
AAA/AUTHEN/ENABLE(00000068): Processing request action LOGIN
AAA/AUTHEN/ENABLE(00000068): Done status FAIL - secret not configured
R2>
! No password was required.   All three methods of the method list were
! tried.
! The first two methods failed, and the third of "none" was accepted.
```

Using the CLI to Troubleshoot AAA for Cisco Routers

One tool you can use when troubleshooting AAA on Cisco routers is the **debug** command. You may use three separate **debug** commands to troubleshoot the various aspects of AAA:

- **debug aaa authentication:** Use this command to display debugging messages for the authentication functions of AAA.
- **debug aaa authorization:** Use this command to display debugging messages for the authorization functions of AAA.
- **debug aaa accounting:** Use this command to display debugging messages for the accounting functions of AAA.

Each of these commands is executed from privileged EXEC mode. To disable debugging for any of these functions, use the **no** form of the command, such as **no debug aaa authentication**.

Example 6-4 shows an example of debugging login authentication, EXEC authorization, and commands at level 15 authorization. As shown in the example, you can use **debug** not only for verification, as in the preceding example, but also as a troubleshooting method.

Example 6-4 *Using* debug *Commands*


```
! R4 will have a loopback, so we can telnet to ourselves to test
R4(config-if)# ip address 4.4.4.4 255.255.255.0
R4(config-if)# exit

! Local user in the database has a privilege level of 15
R4(config)# username admin privilege 15 secret cisco
```

```
! This method list, if applied to a line, will specify local authentication
R4(config)# aaa authentication login AUTHEN_Loc local

! This next method list, if applied to a line, will require authorization
! before giving the administrator an exec shell.   If the user has a valid
! account in the running configuration, the exec shell will be created for
! the authenticated
! user, and it will place the user in their privilege level automatically
R4(config)# aaa authorization exec AUTHOR_Exec_Loc local

! This method list, if applied to a line, will require authorization for
! each and every level 15 command issued.   Because the user is at
! privilege level 15 the router will say "yes" to any level 15 commands
! that may be issued by the user
R4(config)# aaa authorization commands 15 AUTHOR_Com_15 local

! Next we will apply the 3 custom method lists to vty lines 0-4, so that
! when anyone connects via these vty lines, they will be subject to the
! login authentication, the exec authorization, and the level 15 command
! authorizations for the duration of their session.

R4(config)# line vty 0 4
R4(config-line)# login authentication AUTHEN_Loc
R4(config-line)# authorization exec AUTHOR_Exec_Loc
R4(config-line)# authorization commands 15 AUTHOR_Com_15
R4(config-line)# exit
R4(config)#
R4(config)# do debug aaa authentication
AAA Authentication debugging is on
R4(config)# do debug aaa authorization
AAA Authorization debugging is on
R4(config)# exit

! Now test to see it all in action.
R4# telnet 4.4.4.4
Trying 4.4.4.4 ... Open
User Access Verification

Username: admin
Password: [cisco] password not displayed when entering

! It  picked the login authentication list we specified
AAA/BIND(00000071): Bind i/f
AAA/AUTHEN/LOGIN (00000071): Pick method list 'AUTHEN_Loc'
```

```
! It picked the authorization list we specified for the exec shell
R4#
AAA/AUTHOR (0x71): Pick method list 'AUTHOR_Exec_Loc'
AAA/AUTHOR/EXEC(00000071): processing AV cmd=
AAA/AUTHOR/EXEC(00000071): processing AV priv-lvl=15
AAA/AUTHOR/EXEC(00000071): Authorization successful

! It picked the command level 15 authorization list, when we issued the
! configure terminal command, which is a level 15 command.
R4# config t
Enter configuration commands, one per line.  End with CNTL/Z.
R4(config)#
AAA/AUTHOR: auth_need : user= 'admin' ruser= 'R4' rem_addr= '4.4.4.4' priv=
 15 list=
'AUTHOR_Com_15' AUTHOR-TYPE= 'command'
AAA: parse name=tty2 idb type=-1 tty=-1
AAA: name=tty2 flags=0x11 type=5 shelf=0 slot=0 adapter=0 port=2 channel=0
AAA/MEMORY: create_user (0x6A761F34) user='admin' ruser='R4' ds0=0
 port='tty2'
rem_addr='4.4.4.4' authen_type=ASCII service=NONE priv=15 initial_task_
 id='0',
vrf= (id=0)
tty2 AAA/AUTHOR/CMD(1643140100): Port='tty2' list='AUTHOR_Com_15'
 service=CMD
AAA/AUTHOR/CMD: tty2(1643140100) user='admin'
tty2 AAA/AUTHOR/CMD(1643140100): send AV service=shell
tty2 AAA/AUTHOR/CMD(1643140100): send AV cmd=configure
tty2 AAA/AUTHOR/CMD(1643140100): send AV cmd-arg=terminal
tty2 AAA/AUTHOR/CMD(1643140100): send AV cmd-arg=<cr>
tty2 AAA/AUTHOR/CMD(1643140100): found list "AUTHOR_Com_15"
tty2 AAA/AUTHOR/CMD(1643140100): Method=LOCAL
AAA/AUTHOR (1643140100): Post authorization status = PASS_ADD
AAA/MEMORY: free_user (0x6A761F34) user='admin' ruser='R4' port='tty2'
rem_addr='4.4.4.4' authen_type=ASCII service=NONE priv=15 vrf= (id=0)
R4(config)#
! It made a big splash, with lots of debug output, but when you boil it all
! down it means the user was authorized to issue the configure terminal
! command.
```

There is also a **test aaa** command that is very useful when verifying connectivity with a remote ACS server.

This section walked you through the details of AAA using the command line with very exact examples because you need to understand how it works. Now that you have taken

a look at how it works, you should know that you can also use CCP as a GUI to implement the AAA.

Let's take a moment to review where you can find the AAA elements inside CCP. In the configuration section, using the navigation pane on the left, go to **Configure > Router > AAA > AAA Summary**. You will see there an overview of what authentication policies have been created on a router and any authorization or accounting policies, as shown in Figure 6-1.

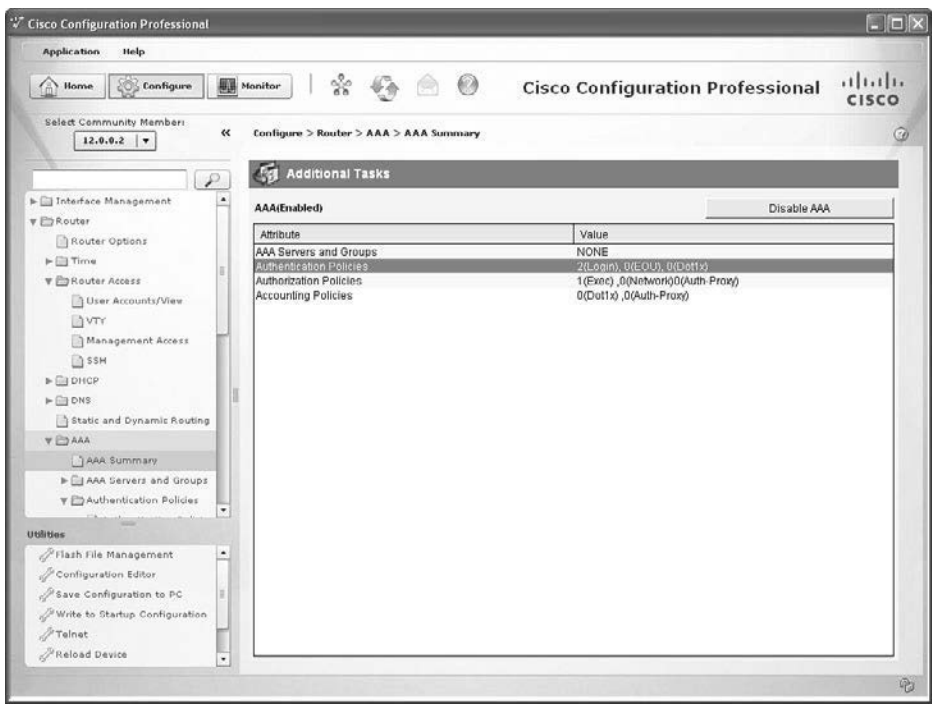

Figure 6-1 *Using CPP to View AAA Policies*

If you wanted to add, edit, or modify your authentication policies, you just navigate to **Configure > Router > AAA > Authentication Policies > Login**, as shown in Figure 6-2.

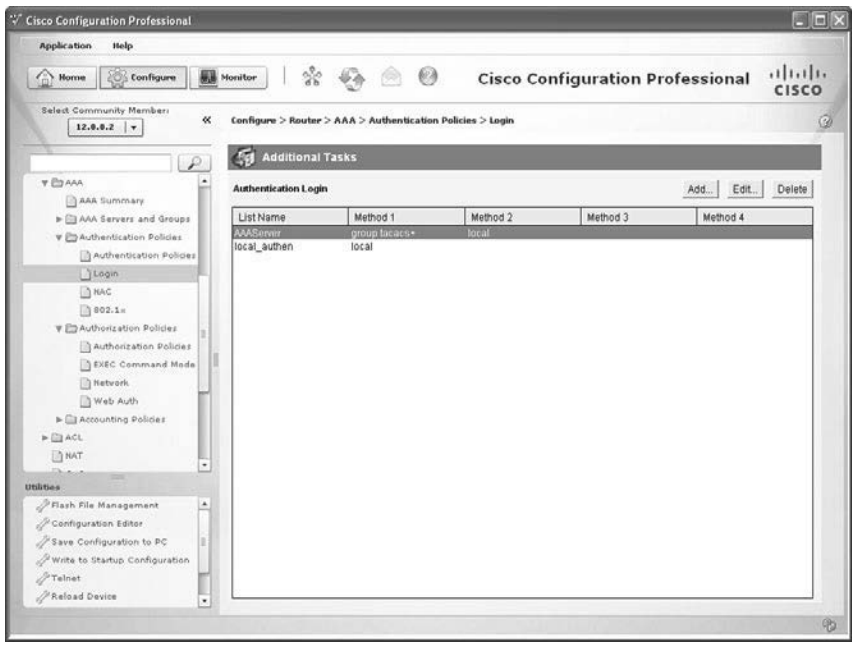

Figure 6-2 *Using CCP to See Method Lists for Login*

If you want to see which method lists were applied to your vty lines, just navigate to **Configure > Router > Router Access > VTY**, as shown in Figure 6-3.

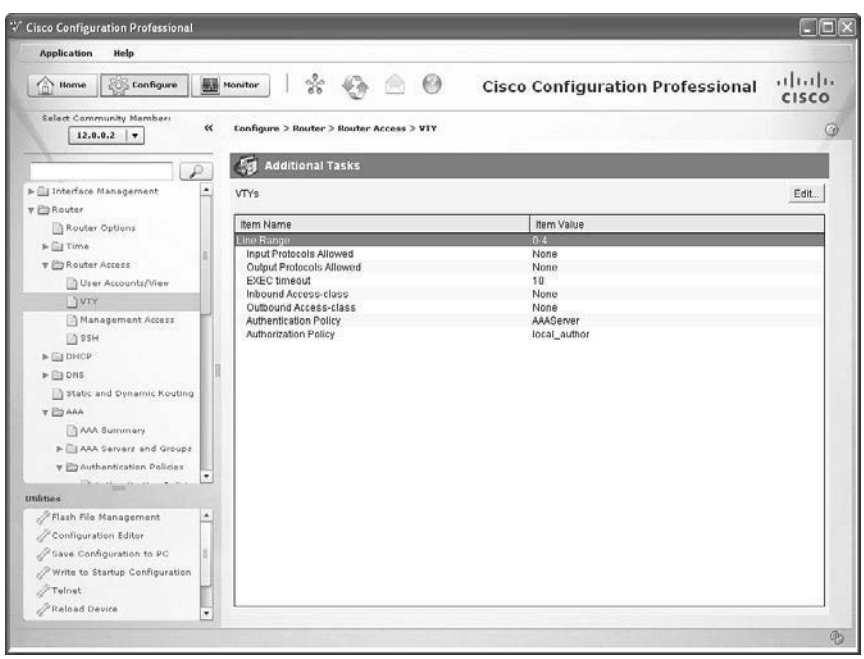

Figure 6-3 *Using CCP to See Which Methods Have Been Applied to the vty Lines*

From here, you can also modify which AAA policies are applied to vty lines by clicking **Edit**, which prompts the opening of an Edit VTY Lines dialog, as shown in Figure 6-4.

Figure 6-4 *Using CPP to Edit vty Line Properties, Including AAA Method Lists Applied*

RBAC Privilege Level/Parser View

You may implement RBAC through AAA, with the rules configured on an ACS server, but you may implement it in other ways, too, including creating custom privilege levels and having users enter those custom levels where they have a limited set of permissions, or creating a *parser view* (also sometimes simply called a *view*), which also limits what the user can see or do on the Cisco device. Each options can be tied directly to a username, so that once users authenticate they may be placed at the custom privilege level, or in the view that is assigned to them.

Let's implement a custom privilege level first, as shown in Example 6-5. The example includes explanations throughout.

Key Topic

Example 6-5 *Creating and Assigning Commands to a Custom Privilege Level*

```
! By default,  we use privilege level 1 (called user mode), and privilege
! level 15 (called privileged mode).   By creating custom levels, (between
! 1-15) and assigning commands to those levels, we are creating custom
! privilege levels
! A user connected at level 8, would have any of the new commands
! associated with level 8, as well as any commands that have been custom
! assigned or defaulted to levels 8 and below.    A user at level 15 has
! access to all commands at level 15 and below.
```

```
! This configuration assigns the command "configure terminal" to privilege
! level 8
R2(config)# privilege exec level 8 configure terminal

! This configuration command assigns the password for privilege level 8
! the keyword "password" could be used instead of secret, but is less secure
! as the "password" doesn't use the MD5 hash to protect the password
! The "0" before the password, implies that we are inputting a non-hashed
! (to begin with) password.  The system will hash this for us, because we
! used the enable "secret" keyword.
R2(config)# enable secret level 8 0 NewPa5s123&
R2(config)# end
R2#
%SYS-5-CONFIG_I: Configured from console by console

! To enter this level, use the enable command, followed by the level you want
! to enter.   If no level is specified, the default level is 15
R2# disable
! Validate that user mode is really privilege level 1
R2> show privilege
Current privilege level is 1
! Context sensitive help shows that we can enter a level number after the
! word enable
R2> enable ?
  <0-15>  Enable level
  view    Set into the existing view
  <cr>

R2> enable 8
Password: [NewPa5s123&] ! note: password doesn't show when typing it in
R2# show privilege
Current privilege level is 8
! We can go into configuration mode, because "configure terminal" is at our
! level
R2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
! Notice we don't have further ability to configure the router, because
! level 8 doesn't include the interface configuration or other router
! configuration commands.
R2(config)# ?
Configure commands:
  beep     Configure BEEP (Blocks Extensible Exchange Protocol)
  call     Configure Call parameters
  default  Set a command to its defaults
```

```
  end      Exit from configure mode
  exit     Exit from configure mode
  help     Description of the interactive help system
  netconf  Configure NETCONF
  no       Negate a command or set its defaults
  oer      Optimized Exit Routing configuration submodes
  sasl     Configure SASL
  wsma      Configure Web Services Management Agents
```

If we are requiring login authentication, we can associate a privilege level with a given user account, and then when users authenticate with their username and password they will automatically be placed into their appropriate privilege level. Example 6-6 shows an example of this.

Example 6-6 *Creating a Local User and Associating That User with Privilege Level 8 and Assigning Login Requirements on the vty Lines*

```
! Create the user account in the local database (running-config) and
! associate that user with the privilege level you want that user to use.
R2(config)# username Bob privilege 8 secret Cisco123
R2(config)# line vty 0 4

! "login local" will require a username and password for access if the "aaa
! new-model" command is not present.   If we have set the aaa new-model,
! then we would also want to create a default or named method list that
! specifies we want to use the local database for authentication.
R2(config-line)# login local

! Note:  Once bob logs in, he would have access to privilege level 8 and
! below, (including all the normal show commands at level 1)
```

Implementing Parser Views

To restrict users without having to create custom privilege levels, you can use a *parser* view, also referred to as simply a *view*. A view can be created with a subset of privilege level 15 commands, and when the user logs in using this view, that same user is restricted to only being able to use the commands that are part of his current view.

To create a view, an enable secret password must first be configured on the router. AAA must also be enabled on the router (**aaa new-model** command).

Example 6-7 shows the creation of a view.

Example 6-7 *Creating and Working with Parser Views*


```
! Set the enable secret, and enable aaa new-model (unless already in
! place)
R2(config)# enable secret aBc!2#&iU
R2(config)# aaa new-model
R2(config)# end

! Begin the view creation process by entering the "default" view, using the
! enable secret
R2# enable view
Password: [aBc!2#&iU] note password not shown when typed

R2#
%PARSER-6-VIEW_SWITCH: successfully set to view 'root'.
R2# configure terminal

! As the administrator in the root view, create a new custom view
R2(config)# parser view New_VIEW
%PARSER-6-VIEW_CREATED: view 'New_VIEW' successfully created.

! Set the password required to enter this new view
R2(config-view)# secret New_VIEW_PW

! Specify which commands you want to include as part of this view.
! commands "exec" refer to commands issued from the command prompt
! commands "configure" refer to commands issued from privileged mode
R2(config-view)# commands exec include ping
R2(config-view)# commands exec include all show
R2(config-view)# commands exec include configure

! This next line adds the ability to configure "access-lists" but nothing
! else
R2(config-view)# commands configure include access-list
R2(config-view)# exit
R2(config)# exit

! Test the view, by going to user mode, and then back in using the new view
R2# disable
R2>enable view New_VIEW
Password: [New_VIEW_PW] Password not shown when typed in

! Console message tells us that we are using the view
%PARSER-6-VIEW_SWITCH: successfully set to view 'New_VIEW'.
```

```
! This command reports what view we are currently using
R2# show parser view
Current view is 'New_VIEW'

! We can verify that the commands assigned to the view work
! Note: we only assigned configure, not configure terminal so we have to
! use the configure command, and then tell the router we are configuring
! from the terminal.   We could have assigned the view "configure terminal"
! to avoid this
R2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

! Notice that the only configuration options we have are for access-list,
! per the view
R2(config)# ?
Configure commands:
  access-list  Add an access list entry
  do           To run exec commands in config mode
  exit          Exit from configure mode
```

We could also assign this view to a user account, so that when users log in with their username and password, they are automatically placed into their view, as shown in Example 6-8.

Example 6-8 *Associating a User Account with a Parser View*

```
R2(config)# username Lois view New_VIEW secret cisco123
```

Note This creation of a username and assigning that user to a view needs to be done by someone who is at privilege level 15.

SSH and HTTPS

Because Telnet sends all of its packets as plain text, it is not secure. SSH allows remote management of a Cisco router or switch, but unlike Telnet, SSH encrypts the contents of the packets to protect it from being interpreted if they fall into the wrong hands.

To enable SSH on a router or switch, the following items need to be in place:

- Hostname other than the default name of “router”
- Domain name
- Generating a public/private key pair, used behind the scenes by SSH
- Requiring user login via the vty lines, instead of just a password. Local authentication or authentication using an ACS server are both options.

- Having at least one user account to log in with, either locally on the router, or on an ACS server

Example 6-9 shows how to implement these components, along with annotations and examples of what happens when the required parts are not in place. If you have a non-production router or switch handy, you might want to follow along.

Example 6-9 *Preparing for SSH*


```
! To create the Public/Private key pair used by SSH, we would issue the
! following command.   Part of the key pair, will be the hostname and the
! domain name.
! If these are not configured first, the crypto key generate command will
! tell you as shown in the next few lines.
Router(config)# crypto key generate rsa
% Please define a hostname other than Router.
Router(config)# hostname R1
R1(config)# crypto key generate rsa
% Please define a domain-name first.
R1(config)# ip domain-name cisco.com

! Now with the host and domain name set, we can generate the key pair
R1(config)# crypto key generate rsa
The name for the keys will be: R1.cisco.com
Choose the size of the key modulus in the range of 360 to 2048 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

! Bigger is better with cryptography, and we get to choose the size for the
! modulus
! The default is 512 on many systems, but you would want to choose 1024 or
! more to improve security.   SSH has several flavors, with version 2 being
! more secure than version 1.   To use version 2, you would need at least a
! 1024 size for the key pair
How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

R1(config)#
%SSH-5-ENABLED: SSH 1.99 has been enabled
! Note the "1.99" is based on the specifications for SSH from RFC 4253
! which indicate that an SSH server may identify its version as 1.99 to
! identify that it is compatible with current and older versions of SSH.

! Create a user in the local database
R1(config)# username Keith secret Ci#kRk*ks
```

```
! Configure the vty lines to require user authentication
R1(config)# line vty 0 4
R1(config-line)# login local

! Alternatively, we could do the following for the requirement of user
! authentication
! This creates a method list which points to the local database, and then
! applies that list to the VTY lines
R1(config)# aaa new-model
R1(config)# aaa authentication login Keith-List-1 local
R1(config)# line vty 0 4
R1(config-line)# login authentication Keith-List-1

! To test this we could SSH to ourselves from the local machine, or from
! another router that has IP connectivity to this router.

R1# ssh ?
  -c    Select encryption algorithm
  -l    Log in using this user name
  -m    Select HMAC algorithm
  -o    Specify options
  -p    Connect to this port
  -v    Specify SSH Protocol Version
  -vrf  Specify vrf name
  WORD  IP address or hostname of a remote system

! Note: one of our local IP addresses is 10.1.0.1
R1# ssh -l Keith 10.1.0.1

Password: <password for Keith goes here>

R1>
! to verify the current SSH session(s)
R1>show ssh
Connection Version Mode Encryption  Hmac       State            Username
0          2.0     IN   aes128-cbc  hmac-sha1  Session started  Keith
0          2.0     OUT  aes128-cbc  hmac-sha1  Session started  Keith
%No SSHv1 server connections running.
R1>
```

Perhaps you want to manage a router via HTTPS. If so, you can use CCP or a similar tool and implement HTTPS functionality, as shown in Example 6-10.

Example 6-10 *Preparing for HTTPS*

```
! Enable the SSL service on the local router.   If it needs to generate
! keys for this feature, it will do so on its own in the background.
R1(config)# ip http secure-server

! Specify how you want users who connect via HTTPS to be authenticated
R1(config)# ip http authentication ?
  aaa     Use AAA access control methods
  enable  Use enable passwords
  local   Use local username and passwords

R1(config)# ip http authentication local

! If you are using the local database, make sure you have at least one user
! configured in the running-config so that you can login.  To test, open
! a browser to HTTPS://a.b.c.d where a.b.c.d is the IP address on the
! router.
```

Implementing Logging Features

Logging is important as a tool for discovering events that are happening in the network and for troubleshooting. Correctly configuring logging so that you can collect and correlate events across multiple network devices is a critical component for a secure network.

Configuring Syslog Support

Example 6-11 shows a typical syslog message and how to control what information is included with the message.

Example 6-11 *Using Service Time Stamps with Syslog Events*

```
R4(config)# interface fa0/0
R4(config-if)# shut
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to administra-
 tively down
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
 state to down
R4(config-if)#

! If we add timestamps to the syslog messages, those timestamps can assist it
! correlating events that occurred on multiple devices
```

```
R4(config)# service timestamps log datetime
R4(config)# int fa0/0
R4(config-if)# no shutdown

! These syslog messages have the date of the event, the event (just after
! the %) a description, and also the level of the event.  The first is 3,
! the second is 5 in the example shown
*Nov 22 12:08:13: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state
   to up
*Nov 22 12:08:14: %LINEPROTO-5-UPDOWN: Line protocol on Interface
   FastEthernet0/0, changed state to up
```

To configure logging, you just tell CCP what the IP address of your syslog server is and which level of logging you want to do to that IP address. As a reminder, level 7, also known as debug level, sends all syslog alerts at level 7 and lower. To configure logging, navigate to **Configure > Router > Logging**, as shown in Figure 6-5.

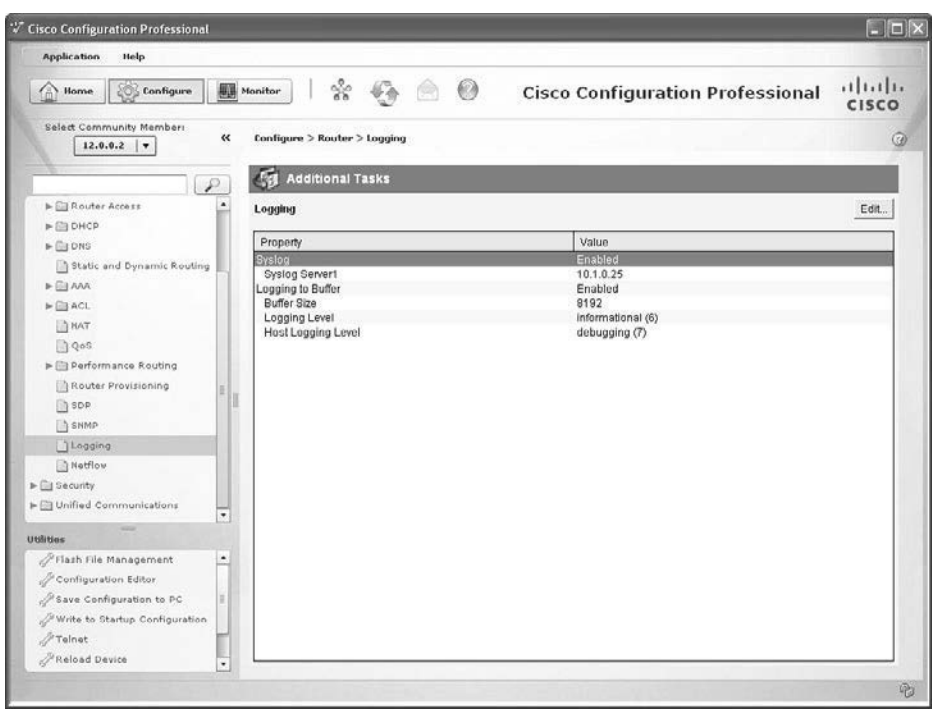

Figure 6-5 *Viewing the Logging Configuration*

To modify any of the logging settings, click the **Edit** button, as shown in Figure 6-6.

Figure 6-6 *Using CCP to Edit the Logging Settings*

In Figure 6-6, we have configured level 7 logging (debugging level) to a syslog server at the IP address of 10.1.0.25, and we have specified that the logging level to the buffer on the router is level 6 (informational level). The memory buffer to hold syslog messages is 8192 bytes. Beyond the 8192 bytes worth of messages in memory, any new messages will replace the oldest messages in a *first in, first out (FIFO)* manner. An example of a syslog server is syslog software running on a PC or dedicated server in your network.

The CCP (for the preceding scenario) creates the equivalent output at the CLI, as shown in Example 6-12.

Example 6-12 *CLI Equivalent Generated by CCP*

```
logging 10.1.0.25
logging trap debugging
logging buffered 8192 informational
```

Figure 6-7 shows the syslog output from the router being collected on the syslog server computer.

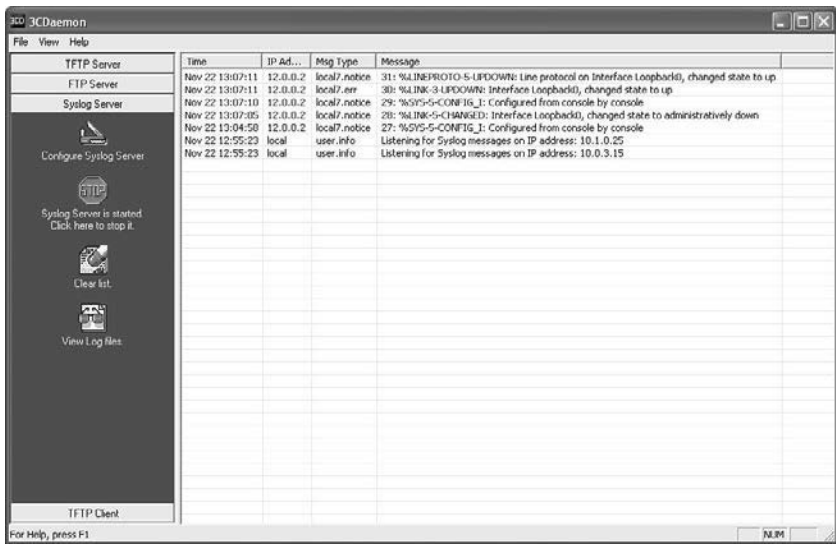

Figure 6-7 *Sample Output Viewed on a Syslog Server*

SNMP Features

Key Topic

Simple Network Management Protocol (SNMP) has become a de facto standard for network management protocols. The intent of SNMP is to manage network nodes, such as network servers, routers, switches, and so on. SNMP versions range from version 1 to 3, with some intermediate steps in between. The later the version, the more security features it has. Table 6-5 describes some of the components of SNMP.

Table 6-5 *Components of SNMPv1 and SNMPv2c Network Management Solutions*

Component	Description
SNMP manager	An SNMP manager runs a network management application. This SNMP manager is sometimes called a *Network Management Server (NMS)*.
SNMP agent	An SNMP agent is a piece of software that runs on a managed device (such as a server, router, or switch).
Management Information Base	Information about a managed device's resources and activity is defined by a series of *objects*. The structure of these management objects is defined by a managed device's *Management Information Base (MIB)*. This can be thought of as a collection of unique numbers associated with each of the individual components of a router.

An SNMP manager can send information to, receive request information from, or receive unsolicited information (called a trap) from a managed device (a router). The managed device runs an SNMP agent and contains the MIB.

Even though multiple SNMP messages might be sent between an SNMP manager and a managed device, consider the three broad categories of SNMP message types:

- **GET:** An SNMP GET message is used to retrieve information from a managed device.
- **SET:** An SNMP SET message is used to set a variable in a managed device or to trigger an action on a managed device.
- **Trap:** An SNMP trap message is an unsolicited message sent from a managed device to an SNMP manager. It can be used to notify the SNMP manager about a significant event that occurred on the managed device.

Unfortunately, the ability to get information from or send configuration information to a managed device poses a potential security vulnerability. Specifically, if an attacker introduces a rogue NMS into the network, the attacker's NMS might be able to gather information about network resources by polling the MIBs of managed devices. In addition, the attacker might launch an attack against the network by manipulating the configuration of managed devices by sending a series of SNMP SET messages.

Although SNMP does offer some security against such an attack, the security integrated with SNMPv1 and SNMPv2c is considered weak. Specifically, SNMPv1 and SNMPv2c use *community strings* to gain read-only access/read-write access to a managed device. You can think of a community string much like a password. Also, be aware that multiple SNMP-compliant devices on the market today have a default read-only community string of "public" and a default read-write community string of "private."

The security weaknesses of SNMPv1 and SNMPv2c are addressed in SNMPv3. SNMPv3 uses the concept of a security model and a security level:

- **Security model:** A security model defines an approach for user and group authentications.
- **Security level:** A security level defines the type of security algorithm performed on SNMP packets. Three security levels are discussed here:
  - **noAuthNoPriv:** The noAuthNoPriv (no authentication, no privacy) security level uses community strings for authentication and does not use encryption to provide privacy.
  - **authNoPriv:** The authNoPriv (authentication, no privacy) security level provides authentication using *Hashed Message Authentication Code (HMAC)* with *message digest algorithm 5 (MD5)* or *Secure Hash Algorithm (SHA)*. However, no encryption is used.
  - **authPriv:** The authPriv (authentication, privacy) security level offers HMAC MD5, or SHA authentication and also provides privacy through encryption. Specifically, the encryption uses the Ci*pher Block Chaining (CBC) Data Encryption Standard (DES) (DES-56)* algorithm.

As summarized in Table 6-6, SNMPv3 supports all three of the previously described security levels. Notice that SNMPv1 and SNMPv2 support only the noAuthNoPriv security level.

Key Topic

Table 6-6 *Security Models and Security Levels Supported by Cisco IOS*

Security Model	Security Level	Authentication Strategy	Encryption Type
SNMPv1	noAuthNoPriv	Community string	None
SNMPv2c	noAuthNoPriv	Community string	None
SNMPv3	noAuthNoPriv	Username	None
	authNoPriv	MD5 or SHA	None
	authPriv	MD5 or SHA	CBC-DES (DES-56)

Through the use of the security algorithms, as shown in Table 6-6, SNMPv3 dramatically increases the security of network management traffic as compared to SNMPv1 and SNMPv2c. Specifically, SNMPv3 offers three primary security enhancements:

- **Integrity:** Using hashing algorithms, SNMPv3 can ensure that an SNMP message was not modified in transit.
- **Authentication:** Hashing allows SNMPv3 to validate the source of an SNMP message.
- **Encryption:** Using the CBC-DES (DES-56) encryption algorithm, SNMPv3 provides privacy for SNMP messages, making them unreadable by an attacker who might capture an SNMP packet.

To configure SNMP on the router is simple, especially with CCP. If you know the community strings to use, and the IP address of the SNMP manager, you can configure it on the router by navigating to **Configure > Router > SNMP** and from there use the **Edit** button to add, change, or remove any of the SNMP-related settings. CCP enables command-line editing through the Utilities menu, but currently the SNMP Properties window does not support the configuration of SNMPv3. You can configure the basic SNMPv1 information, as shown in Figure 6-8.

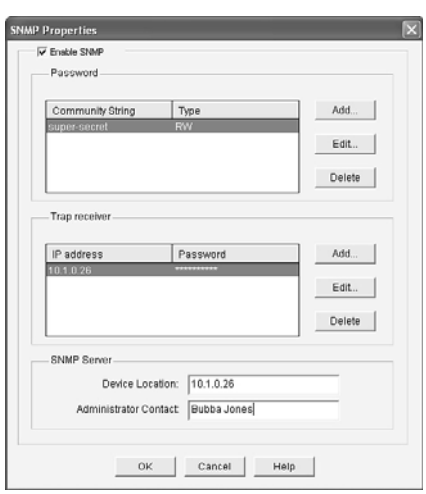

Figure 6-8 *Using CCP to Configure SNMPv1 Information*

The command-line output for this GUI would look similar to that shown in Example 6-13.

Example 6-13 *Output Created by CCP for Implementing SNMPv1*

```
snmp-server location 10.1.0.26
snmp-server contact Bubba Jones
snmp-server community super-secret RW
snmp-server host 10.1.0.26 trap cisK0tRap^
```

Configuring NTP

Because time is such an important factor, you should use *Network Time Protocol (NTP)* to synchronize the time in the network so that events that generate messages and time stamps can be correlated. You can use CCP to implement the NTP in addition to using the CLI. Let's take a look at both right now.

To configure the NTP, you first need to know what the IP address is of the NTP server you will be working with, and you also want to know what the authentication key is and the key ID. NTP authentication is not required to function, but is a good idea to ensure that the time is not modified because of a rogue NTP server sending inaccurate NTP messages using a spoofed source IP address.

Armed with the NTP server information, in CCP you go to **Configure > Router > Time > NTP and SNTP** and click **Add** and put in the information about the server you will be getting the time from. When done, you click **OK** to close the dialog box. It may take anywhere between 5 and 15 minutes for the router to synchronize its clock. In Figure 6-9, this router is being told that the NTP server is at 55.1.2.3, that it should source the NTP requests from its IP address on its local Fast Ethernet 0/0 interface, and that it should use key number 1, and the password associated with that key. If multiple NTP servers were configured, the Prefer option is used to identify the preference of which NTP server to use.

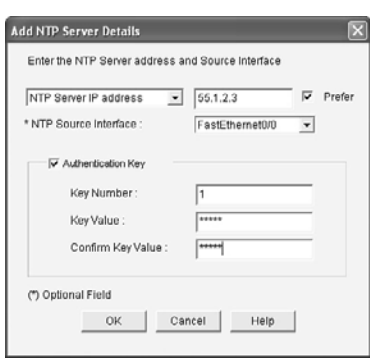

Figure 6-9 *Configuring a Router to Use an NTP Server*

NTP supports authentication on a Cisco router because the router supports NTPv3. Example 6-14 shows the effective equivalent syntax that is created and delivered to the router.

Example 6-14 *Using Authentication via Keys with NTPv3*

```
ntp update-calendar
ntp authentication-key 1 md5 pAs5w0rd!3@
ntp authenticate
ntp trusted-key 1
ntp server 55.1.2.3 key 1 source FastEthernet0/0 prefer
```

To verify the status on this router acting as a NTP client, you could use the commands from the CLI as shown in Example 6-15.

Example 6-15 *Verifying Synchronization from the NTP Client*

```
R2# show ntp status
Clock is synchronized, stratum 4, reference is 55.1.2.3
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**24
reference time is D27619E3.7317ACB3 (12:53:55.449 UTC Tue Nov 22 2011)
clock offset is 0.0140 msec, root delay is 0.00 msec
root dispersion is 0.97 msec, peer dispersion is 0.43 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000000053 s/s
system poll interval is 64, last update was 130 sec ago.

R2# show ntp association
  address    ref clock     st  when  poll  reach  delay  offset  disp
*~55.1.2.3   127.127.1.1   3   4     64    77     0.000  14.090  190.28
 * sys.peer, # selected, + candidate, - outlyer, x falseticker,
   ~ configured
R2#
```

Note NTP uses UDP port 123. If NTP does not synchronize within 15 minutes, you may want to verify that connectivity exists between this router and the NTP server that it is communicating to. You also want to verify that the key ID and password for NTP authentication are correct

Securing the Cisco IOS Image and Configuration Files

If a router has been compromised, and the flash file system and NVRAM have been deleted, there could be significant downtime as the files are put back in place before restoring normal router functionality. The Cisco Resilient Configuration feature is intended to improve the recovery time by making a secure working copy of the IOS image and startup configuration files (which are referred to as the *primary bootset*) that cannot be deleted by a remote user.

To enable and save the primary bootset to a secure archive in persistent storage, follow Example 6-16.

Example 6-16 *Creating a Secure Bootset*


```
! Secure the IOS image
R6(config)# secure boot-image
%IOS_RESILIENCE-5-IMAGE_RESIL_ACTIVE: Successfully secured running image

! Secure the startup-config
R6(config)# secure boot-config
%IOS_RESILIENCE-5-CONFIG_RESIL_ACTIVE: Successfully secured config archive
 [flash:.runcfg-20111222-230018.ar]

! Verify the bootset
R6(config)# do show secure bootset
IOS resilience router id FTX1036A13J

IOS image resilience version 12.4 activated at 23:00:10 UTC Thu Dec 22 2011
Secure archive flash:c3825-advipservicesk9-mz.124-24.T.bin type is image
 (elf) []
  file size is 60303612 bytes, run size is 60469256 bytes
  Runnable image, entry point 0x80010000, run from ram

IOS configuration resilience version 12.4 activated at 23:00:18 UTC Thu Dec
 22 2011
Secure archive flash:.runcfg-20111222-230018.ar type is config
configuration archive size 1740 bytes

! Note: to undo this feature, (using the "no" option in front of the command)
! you must be connected via the console.  This prevents remote users from
! disabling the feature.
```

Exam Preparation Tasks

Review All the Key Topics

Review the most important topics from this chapter, denoted with a Key Topic icon. Table 6-7 lists these key topics.

Table 6-7 *Key Topics*

Key Topic Element	Description	Page Number
Text	Management plane best practices	95
Text	AAA components	98
Text	Storing usernames, passwords, and access rules	98
Text	Router access authentication	100
Table 6-2	AAA components to secure administrative and remote LAN access	101
Text	The AAA method list	101
Table 6-3	Method list options	101
Text	Limiting the administrator by assigning a view	103
Text	Encrypted management protocols	103
Text	Using logging files	104
Text	User authentication in AAA	108
Text	Using the CLI to troubleshoot AAA for Cisco routers	113
Example 6-4	Using **debug** commands	113
Example 6-5	Creating and assigning commands to custom privilege levels	118
Text	Implementing parser views	120
Example 6-7	Creating and working with parser views	121
Example 6-9	Preparing for SSH	123
Text	SNMP features	128
Table 6-6	Security models and security levels supported by Cisco IOS	130
Example 6-16	Creating a secure bootset	133

Complete the 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 Tables Answer Key," also on the CD, includes completed tables and lists so that you can check your work.

Define Key Terms

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

AAA, method list, custom privilege level, parser view, SSH, syslog, SNMP, NTP, secure bootset

Command Reference to Check Your Memory

This section includes the most important configuration and EXEC commands covered in this chapter. To see how well you have memorized the commands as a side effect of your other studies, cover the left side of Table 6-8 with a piece of paper, read the descriptions on the right side, and see whether you remember the commands.

Table 6-8 *Command Reference*

Command	Description
service password-encryption	Encrypt most plaintext passwords in the configuration.
aaa new-model	Enable AAA features.
aaa authentication login default local	Create a default method list for character mode login that will use the local database (running config) on the router or switch.
enable view	Enter the root parser view, from where you can create additional views. This requires that **aaa new-model** already be in place in the configuration.
privilege exec level *8 show startup-config*	Assign a **show startup-config** command to a custom privilege level 8.
crypto key generate rsa	Create the public/private key pair required for SSH.
secure boot-image	Secure the IOS image on flash
aaa authentication *bubba* **local enable**	Create an authentication method list called bubba that will use the local database first, and if the username does not exist, will require the enable secret to allow login.
line console 0 **login authentication bubba**	Apply the method list named bubba to the console port.

This page intentionally left blank

Index

A

B

C

D

E

F

G

H

I

J-K

L

M

N

O

P

Q

R

S

T

U

V

W

X - Y

Z