The Complete Cisco VPN Configuration Guide

By Richard Deal

.....
Publisher: **Cisco Press**
Pub Date: **December 15, 2005**
ISBN: **1-58705-204-0**
Pages: **1032**

Table of Contents | Index

Overview

The definitive guide to building a complete VPN solution with Cisco routers, PIX Firewalls, concentrators, and remote access clients

- A complete resource for understanding VPN components and VPN design issues
- Understand state-of-the-art VPN connection types like IPSec, PPTP, and L2TP
- Real-world case studies detail implementation of complex VPN configurations on Cisco devices including routers, PIX Firewalls, concentrators, and software and hardware clients

Virtual Private Networks (VPNs) are the most popular component in a company's remote access solution. With increased use of Internet connectivity and less reliance on private WAN networks, VPNs provide a much-needed secure method of transferring critical information. Vendors like Cisco Systems continually upgrade products to provide features that take advantage of advances in standards and protocols like IPSec and L2TP (Layer 2 Tunneling Protocol). Cisco VPN equipment is at the center of this access revolution; nearly every enterprise network contains Cisco gear and most of this equipment supports VPN functionality. As Cisco integrates security and access features into routers, firewalls, clients, and concentrators, its solutions become ever more accessible to companies with networks of all sizes. Engineers need to know how to set up various VPN deployments using Cisco equipment. Currently, there is no single book that covers how to deploy VPNs using all of Cisco's VPN-capable products, including IOS routers, PIX Firewalls, 3000 series Concentrators, and the Cisco software and hardware clients. *The Complete Cisco VPN Configuration Guide* contains detailed explanations of all Cisco VPN products, describing the details of setting up IPSec and SSL connections on any type of Cisco device, including concentrators, clients, routers, or the PIX Firewall. With copious configuration examples and troubleshooting scenarios, it offers clear information on VPN design.

NEXT

The Complete Cisco VPN Configuration Guide

By Richard Deal

Publisher: **Cisco Press**
Pub Date: **December 15, 2005**
ISBN: **1-58705-204-0**
Pages: **1032**

Table of Contents | Index

PREV NEXT

Copyright

The Complete Cisco VPN Configuration Guide

Richard Deal

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

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

First Printing December 2005

Library of Congress Cataloging-in-Publication Number: 2004105575

ISBN: 1-58705-204-0

Trademark Acknowledgments

Warning and Disclaimer

This book is designed to provide information about setting up VPNs, on Cisco products, including the IOS routers, VPN 3000 concentrators, PIX and ASA security appliances, VPN Client software, 3002 hardware client, SSL Client, and Microsoft's Windows Client. 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 author, 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.

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 the U.S. please contact: **International Sales** international@pearsoned.com

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value.

PREV NEXT


ABC Amber CHM Converter Trial version
Please register to remove this banner.
http://www.processtext.com/abcchm.html

About the Author

Richard A. Deal has 20 years of experience in the computing and networking industry including networking, training, systems administration, and programming. In addition to a B.S. in mathematics and computer science from Grove City College; Richard holds many certifications from Cisco. For the past eight years, Richard has operated his own company, The Deal Group Inc.

Besides teaching various Cisco certification courses, Richard has also published many books. Richard recently published *Cisco Router Firewall Security* with Cisco Press, which is on the Cisco CCIE Security reading list. Richard is also the author of the following books: *CCNA Cisco Certified Network Associate Study Guide* (Exam 640-801), *Cisco PIX Firewalls*, and *CCNP BCMSN Exam Cram 2* (642-811).

Richard is actively writing Cisco certification self-preparation tests for Boson (http://www.boson.com) and has written many self-preparation tests for the CCSP certification. Richard currently lives with his wife, Natalya, and his daughter, Emily Alina, in Oviedo, FL, just outside of Orlando.

About the Technical Reviewers

Pete Davis has been working with computers and networks since he was able to walk. By age 15, he was one of the youngest professional network engineers and one of the first employees at an Internet service provider. Pete implemented and maintained the systems and networks behind New England's largest consumer Internet service provider, TIAC (The Internet Access Company). In 1997, Pete joined Shiva Corporation as product specialist. Since 1998, Pete has been with Altiga Networks, a VPN Concentrator manufacturer in Franklin, MA, that was acquired by Cisco Systems on March 29, 2000. As product line manager, Pete is responsible for driving new VPN-related products and features.

Stephen Marcinek, CCIE No. 7225, is a technical trainer for Boson Training, a Cisco Learning Partner. He develops course content and delivers numerous classes in Cisco Networking and Security, from Introductory to CCIE level. Stephen also consults for numerous large organizations. He holds a bachelor's degree from Rutgers University and is a member of Mensa.

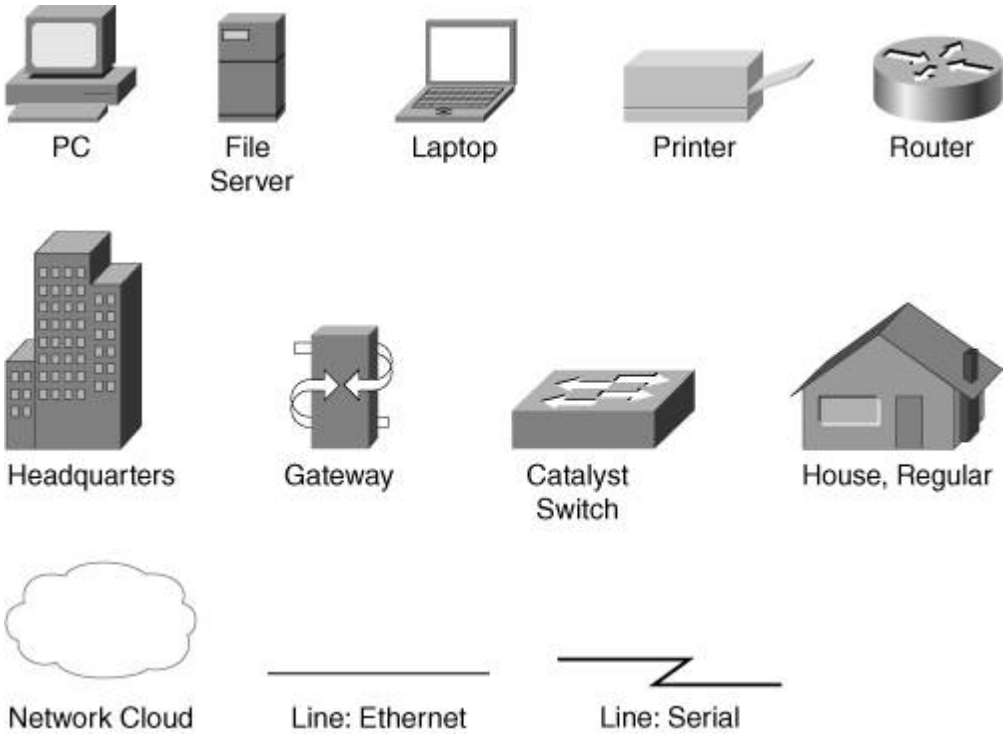
Mark Newcomb, CCNP, CCDP, is a retired network security engineer. Mark has more than 20 years of experience in the networking industry, focusing on the financial and medical industries. Mark is a frequent contributor and reviewer for Cisco Press books.

Acknowledgments

A real special thanks goes out to the technical editors of this book: Pete Davis, Mark Newcomb, and especially Steve Marcinek. I have personally known Steve for quite some time and can always count on him as a coworker and friend.

A big thank you also goes out to the production team for this book. Raina Han, Brett Bartow, Sheri Cain, Mary Beth Ray, Jim Schachterle, Michelle Grandlin, Jill Batistick, and Rozi Harris have been incredibly professional and a pleasure to work with. I couldn't have asked for a finer team.

Icons Used in This Book



PREV NEXT

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:

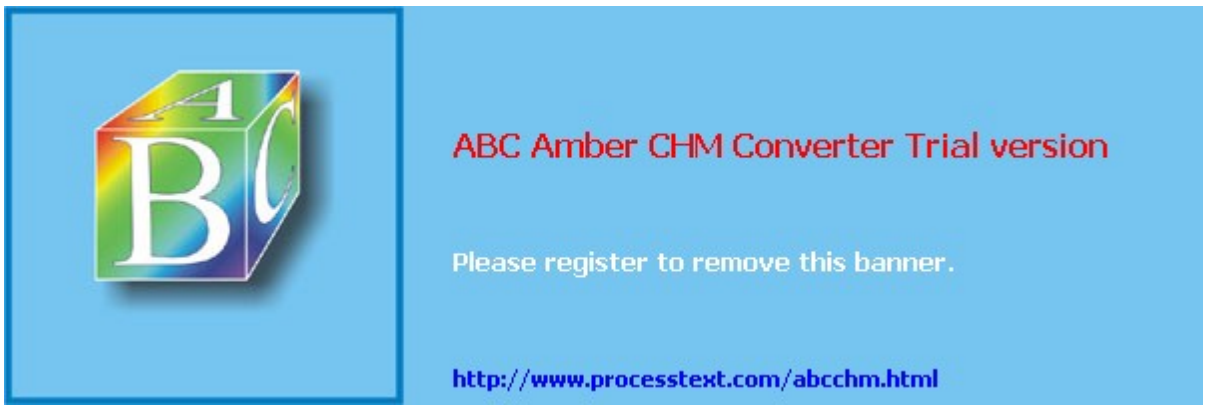
- **Boldface** indicates commands and keywords that are entered literally as shown. In actual configuration examples and output (not general command syntax), boldface indicates commands that are input manually by the user (such as a **show** command).
- *Italics* indicate arguments for which you supply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

PREV NEXT

Introduction

Cisco has been an important part of the networking industry for many years and will continue to play a key role in company networks. The first router product I worked on, back in 1993, was a Cisco AGS+. I have seen many flavors of the IOS, including the introduction of most of the security features you see today in the Cisco IOS operating system, such as IPsec. Over the past several years, I have seen security becoming a key component in network design. And with more and more companies using the Internet as a business tool today, security is more important than ever, especially the use of VPNs.

PREV NEXT



Goals and Methods

Four years ago, I realized that there were many certification books to help people pass Cisco security certification exams; however, I found that there were no books of any substance that brought Cisco security features together to be applied in a real-life situation. I continually monitor various Cisco newsgroups and constantly see questions related to how to implement various Cisco security features. This was the foundation of my first security book, *Cisco PIX Firewalls*, with McGraw-Hill/Osborne.

As a security professional, I'm constantly asked questions about designing, implementing, and tuning VPNs that involve Cisco products. Because I constantly see many of the same questions over and over again, I've decided to pull this information together into a guide that encompasses building VPNs with Cisco products. The purpose of this book is to show you how to implement VPNs using the Cisco line of VPN-capable products, including the following:

- IOS Routers
- VPN 3000 Concentrators
- PIX and ASA Security Appliances
- VPN 3002 Hardware Client
- Cisco VPN Client Software
- Cisco SSL VPN Client Software
- Microsoft Client Software

Because this is not a certification book, but a "how to" book, I've included the following methods to help you with the "how to" process:

- Explaining what VPNs are and the technologies commonly used by VPNs
- Discussing the VPN implementation types, including IPsec, L2TP and PPTP, and SSL
- Discussing Cisco VPN-capable products and their features and capabilities
- Configuring the VPN 3000 concentrators for site-to-site, or LAN-to-LAN (L2L), and remote access VPNs
- Configuring the IOS routers for site-to-site and remote access VPNs
- Configuring the PIX and ASA security appliances for site-to-site and remote access VPNs
- Configuring the Cisco VPN Client, Cisco VPN 3002 Hardware Client, Microsoft VPN Windows Client, and the Cisco SSL Client for remote access VPNs
- Troubleshooting common VPN problems on Cisco products
- Supplying many examples, including a detailed case study at the end of the book, to show you how Cisco security features should be implemented
- Introducing you to real-life situations I've had to deal with in implementing and troubleshooting VPNsevery chapter includes a sidebar illustrating my own experiences

Who Should Read This Book?

This book is intended to provide the necessary framework for implementing VPNs on Cisco products. With this goal in mind, this is a "how-to" book. Although other objectives can be achieved from using this book, including preparation for the Cisco CCSP SNRS, SNPA, CSVPN, and SND exams, this book is written with one main goal in mind: implementing VPNs using Cisco VPN-capable products.

This book assumes that you have a basic understanding of the following:

- Cisco routers and the IOS operating system and command-line interface (CLI)
- Cisco PIX and ASA security appliances and the Finesse Operating System (FOS) and its CLI

For the other Cisco products I discuss in this book, such as the VPN 3000 concentrators, I assume you have little or no product knowledge. However, I assume that you do have an intermediate-to-advanced level of knowledge of Cisco products, and at a minimum, you should have the Cisco CCNA certification to understand and make best use of the material in this book.

Because this book focuses on using VPNs to provide secure connectivity between devices or networks, this book will be very useful for any network administrator or engineer who currently must implement VPNs using Cisco products.

How This Book Is Organized

Although this book can be read cover-to-cover, it is designed to be flexible and allow you to move easily between chapters and sections of chapters to cover just the material that you are interested in. However, each part and each chapter in each part builds upon the others. There are six parts to this book, excluding the book's front matter. Each part deals with an important component of VPNs on Cisco products. The following are the topics covered in the chapters of this book:

- **Chapter 1, "Overview of VPNs"** This chapter contains a brief overview of the kinds of threats that you'll face in moving traffic across unprotected networks, and how VPNs can be used to secure your traffic. The chapter begins with a discussion of three common problems with moving traffic across an unprotected network: eavesdropping, masquerading, and man-in-the-middle attacks. The chapter then covers how a VPN can be used to defeat these attacks. I define what a VPN is, its components, designs, types of VPN implementations, and criteria you should consider when choosing a particular VPN solution.

- **Chapter 2, "VPN Technologies"** This chapter contains an introduction to the technologies used to implement VPNs. More commonly than not, I'm surprised to see that many network administrators and engineers don't understand the technologies they are using to protect their traffic. Therefore, this chapter will give you an overview of what technologies VPNs commonly use and how these technologies work. This chapter covers keys, encryption, packet authentication, exchanging keys, and authentication methods.

- **Chapter 3, "IPsec"** This chapter provides an overview of one of the most common VPN implementations: IPsec. Even though this book focuses on choosing VPN devices, configuring them, and troubleshooting them, understanding how a particular VPN is implemented is important to complete the previously mentioned tasks. Of all the VPN implementations I've worked with, IPsec is the most complex; and I've yet to find a book that brings the topic together in an easy-to-read and understandable fashion. Therefore, this chapter will discuss the standards that make up IPsec, the two phases involved in building a secure session, how the connections are built between IPsec peers, and common problems that can cause IPsec sessions to break, including address translation and firewalls, and solutions to these problems.

- **Chapter 4, "PPTP and L2TP"** This chapter discusses two popular VPN implementations used in Microsoft-centric shops: PPTP and L2TP. I discuss how each of these VPNs are implemented and then compare the two solutions.

- **Chapter 5, "SSL VPNs"** This chapter covers the use of SSL to implement a VPN solution. I discuss what an SSL VPN is and the three basic methods of implementation: clientless, thin client, and network. I also discuss when SSL VPNs are commonly used and the Cisco SSL solution: WebVPN.

- **Chapter 6, "Concentrator Product Information"** This chapter introduces the VPN 3000 concentrators, which are commonly used to implement remote access VPNs. The chapter discusses the concentrator models, the modules that can be inserted into their chassis, the features available in various software versions, and an introduction to the CLI and GUI of the concentrators.

- **Chapter 7, "Concentrator Remote Access Connections with IPsec"** This chapter focuses on terminating IPsec remote access sessions on VPN 3000 concentrators. I begin the chapter by discussing the two methods of controlling remote access: groups and users. I then discuss how to terminate IPsec remote access sessions on the concentrators. I conclude the chapter by discussing a new feature on the concentrators, Network Access Control (NAC), which can be used to force remote access clients to meet certain criteria before allowing them to establish a VPN session.

- **Chapter 8, "Concentrator Remote Access Connections with PPTP, L2TP, and WebVPN"** This chapter focuses on terminating remote access sessions on VPN 3000 concentrators using PPTP, L2TP, and WebVPN. I begin the chapter by discussing the configuration of PPTP and L2TP on the concentrators. The second half of the chapter discusses the WebVPN (SSL VPN) features of the concentrators, where I cover topics

Additional Information

Many of the features I discuss in this book are only supported on various software versions on the particular products. To find whether or not a feature is supported on a specific product platform or software version, use the Cisco Feature Navigator at http://www.cisco.com/go/fn. You'll need a CCO account to use this feature.

For a list of product security advisories and notices for Cisco products and software version releases, visit http://www.cisco.com/en/US/products/products_security_advisories_listing.html.

Tip

I highly recommend that you *carefully* view this list before loading a specific software version on your VPN product.

PREV NEXT

Part I: VPNs

Chapter 1. Overview of VPNs

This chapter introduces the concepts of virtual private networks (VPNs) and why they are used. I examine issues with sending traffic across public networks and what VPNs can do to protect this traffic. I introduce connection methods for VPNs, types of VPNs, things to consider when using VPNs, VPN components, VPN designs and issues, examples of VPN implementations, and some issues to consider when choosing a VPN implementation. Further chapters in this book expand on the topics introduced here.

PREV NEXT

Traffic Issues

VPNs were developed initially to deal with security issues of transmitting clear text data across a network. Clear text data is information that can examined and understood by any person, including the source, destination, and anyone in between. Examples of applications that send traffic in a clear text format are Telnet, file transfers via FTP or TFTP, e-mail using the Post Office Protocol (POP) or Simple Mail Transfer Protocol (SMTP), and many others. Unethical individuals, such as hackers, can take advantage of applications that send clear text data to execute the following types of attacks:

- Eavesdropping
- Masquerading
- Man-in-the-middle

Each type of attack exposes your data and company assets to various risks. The following three sections discuss these attacks in more depth.

Eavesdropping Attacks

The most common type of attack with clear text data is *eavesdropping*. In an eavesdropping attack, a person examines the contents of packets as they are transmitted between two devices. Some types of applications and protocols are susceptible to eavesdropping attacks, including Telnet, POP, HTTP, TFTP, FTP, Simple Management Network Protocol (SNMP), and others.

With all of these applications and protocols, authentication information, like usernames and passwords, is passed, in clear text, between two devices. A hacker can use this information to execute access and other types of attacks.

Note

Even though some protocols might send information in clear text, they, in many cases, have a minimal authentication method to verify an individual's identity before allowing that person access to a resource. For example, applications such as Telnet, POP, and SMTP allow for authentication, even though the authentication information is sent in clear text. Actually, these protocols initially were not designed for security, but to solve specific connectivity problems. However, things have changed since these applications were developed in the 1970s, 1980s, and early 1990s, especially with the explosion of Internet usage.

Eavesdropping Tools

Typically, a protocol analyzer is used to examine (sniff) packets. The analyzer can be a hardware-based solution or a PC with a promiscuous network interface card (NIC) and appropriate software. For this kind of attack to work, the attacker must have access to a connection between the actual source and destination devices.

There are two main categories of protocol analyzers: general and attack. A general protocol analyzer captures all packets it sees and typically is used as a diagnostic tool to troubleshoot problems. There are many freeware, software-based protocol analyzers available that perform this function.

An attack protocol analyzer, on the other hand, is an enhanced form of a general protocol analyzer. Attack protocol analyzers look at certain types of applications and protocols for authentication, financial, and security information. An attacker will use this specific information to execute other types of attacks.

Eavesdropping Solutions

Sensitive information includes credit card information, personal information, Social Security numbers, telephone numbers and addresses, usernames and passwords, and proprietary information. Because many protocols and applications are insecure when transmitting sensitive information (they send their information in clear text), protection becomes necessary. One solution is to use one-time passwords (OTP) with token cards. This prevents someone from

VPN Definition

I've mentioned, in the last few sections, that VPNs can be used to deal with certain kinds of attacks. Therefore, the question is: What is a virtual private network (VPN)? Having worked in the computer field for almost twenty years, I am constantly asked to explain various technologies and what they are used for. In security, the most common question I'm asked is "What's a VPN?" I've probably seen all possible explanations of what a VPN is, such as the following:

- It's an encrypted tunnel.
- It uses IPsec, GRE, PPTP, SSL, L2TP, or MPLS (described later in the chapter).
- It encrypts data.
- It protects traffic across the Internet.
- It protects your data from hackers and attacks.

As you can see, many people have different views or perceptions of what a VPN is. For example, if you would search the Internet for the term VPN, you would easily find dozens of different, sometimes similar, sometimes conflicting definitions. For instance, at www.webopedia.com, the definition of a VPN is: "a network that is constructed by using public wires to connect nodes. For example, there are a number of systems that enable you to create networks using the Internet as the medium for transporting data. These systems use encryption and other security mechanisms to ensure that only authorized users can access the network and that the data cannot be intercepted."

VPN Description

To me, webopedia's definition makes it even more confusing as to what a VPN is. To clear up the confusion, I'll discuss what a VPN actually is and the different types of categories a VPN falls under. I'll then discuss some different types of VPN implementations. In its simplest form, a VPN is a connection, typically protected, between two entities that are not necessarily directly connected. The two entities could be directly connected via a point-to-point link, but it is more common to see them separated by more than one hop or network. The term "entities" could refer to either a specific device or a particular network (multiple devices). The connection, in many instances, crosses a public network; however, VPNs easily can be used for internal purposes. And the word "protected" in my definition of a VPN is somewhat open to interpretation. Most people assume that this means encryption (protecting traffic from eavesdropping attacks), or that packets haven't been tampered with (by a man-in-the-middle attack). And these assumptions are typically correct; however, a good VPN solution will deal with most, if not all, of the following issues:

- Protecting data from eavesdropping by using encryption technologies, such as RC-4, DES, 3DES, and AES
- Protecting packets from tampering by using packet integrity hashing functions such as MD5 and SHA
- Protecting against man-in-the-middle attacks by using identity authentication mechanisms, such as pre-shared keys or digital certificates
- Protecting against replay attacks by using sequence numbers when transmitting protected data
- Defining the mechanics of how data is encapsulated and protected, and how protected traffic is transmitted between devices
- Defining what traffic actually needs to be protected

As you can see, a VPN is responsible for all kinds of functions.

Note

Of course, not every VPN implementation will include all of these components or will not

VPN Components

Now that you have a basic understanding of what a VPN is, let's discuss the components that make up a traditional VPN. Not every VPN implementation will include any or all of these components. Plus, based on the requirements listed in your security policy, you might not need all of these components. Therefore, you need to examine your security policy to determine which VPN implementation (or implementations) has the necessary components to meet your security policy's requirements.

The following sections will discuss some of the more important components that are typically part of a VPN implementation. Specifically, they will cover the following:

- Authentication
- Encapsulation Method
- Data Encryption
- Packet Integrity
- Key Management
- Non-Repudiation
- Application and Protocol Support
- Address Management

Authentication

One concern you might have is to somehow verify a device's or user's identity before allowing it to establish a VPN connection to your network. There are two general categories of authentication:

- Device
- User

Device Authentication

Device authentication allows you to restrict VPN access to your network based on authentication information that a remote VPN device provides. Typically this is one of the following two types of authentication:

- Pre-shared key or keys
- Digital signature or certificate

Pre-shared keys are typically used in smaller VPN environments. One or more keys is configured and used to authenticate a device's identity. Setting up pre-shared key authentication is very simple. Many administrators prefer its use instead of digital signatures or certificates, which require a lot more work to set up. Pre-shared keys requires you to manually configure a key or keys on each device that will participate with VPN connectivity.

Given the amount of configuration, though, pre-shared keys have one main disadvantage: they don't scale well. For example, assume that you currently have nine sites with a router at each site, where pre-shared keys are used for device authentication and the VPN L2L design is fully meshed between the sites. You add an additional site. This requires you to add nine keys to the router at the new site and set up keying information on the routers at the other nine sites. So adding more sites makes addition and management of authentication keys very complex.

Note

The original idea behind the use of pre-shared keys was that a key would be associated with a specific static source IP address; however, this concept did not work for remote access

VPN Designs

After you have weighed the VPN components necessary for your VPN implementation, you'll need to take a look at the layout of your VPN design: what devices need VPN functionality (servers, routers, firewalls, PCs, and so on), what type of VPN functionality they need, and what connectivity is required. This information will greatly assist you in picking a suitable VPN implementation, and will bring up any design issues and problems you should consider before making a final choice.

The following sections discuss some VPN designs, and some issues to consider when using VPNs in your network. Specifically, the section covers the following:

- Connection types
- Considerations
- Redundancy

Connection Types

From a design perspective, this section will cover the various types of basic connections that VPNs use. These connections include point-to-point, hub-and-spoke, and fully meshed.

Point-to-Point

There are two basic types of VPN point-to-point connections:

- Device-to-device
- Network-to-network

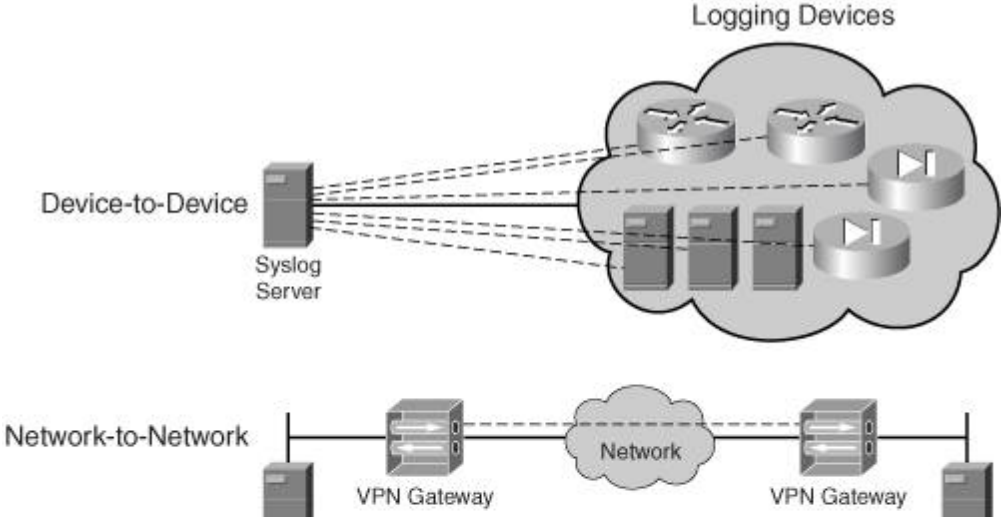
Device-to-Device Connection

A device-to-device VPN connection is a user-to-user VPN type, where only two devices are involved in the VPN. This connection type is usually deployed where only a specific type of traffic between two devices needs to be protected. An example of device-to-device connection includes backing up the configuration file on a Cisco router to a TFTP server, sending SNMPv2 traffic from a managed device, like a Catalyst switch, to an SNMP management server, or sending logging traffic from a PIX security appliance to a syslog server.

One concern of device-to-device connections is that they place an extra burden on the VPN endpoint device. For example, imagine that you have a syslog server that has to handle logging information from 200 devices. In this situation, the syslog server would have to terminate 200 VPN device-to-device connections, which might place an undue burden on it. This is shown in the top part of Figure 1-8.

Figure 1-8. Device-to-Device and Network-to-Network Connections

[View full size image]



VPN Implementations

Throughout the chapter up to this point, I have repeatedly referred to "VPN implementations," but haven't defined what a VPN implementation actually is. First, let me discuss what a VPN is and what it can encompass. Specifically, the following sections will discuss these popular VPN implementation methods, including how they are implemented, and their basic advantages and disadvantages:

- GRE
- IPsec
- PPTP
- L2TP
- MPLS
- SSL

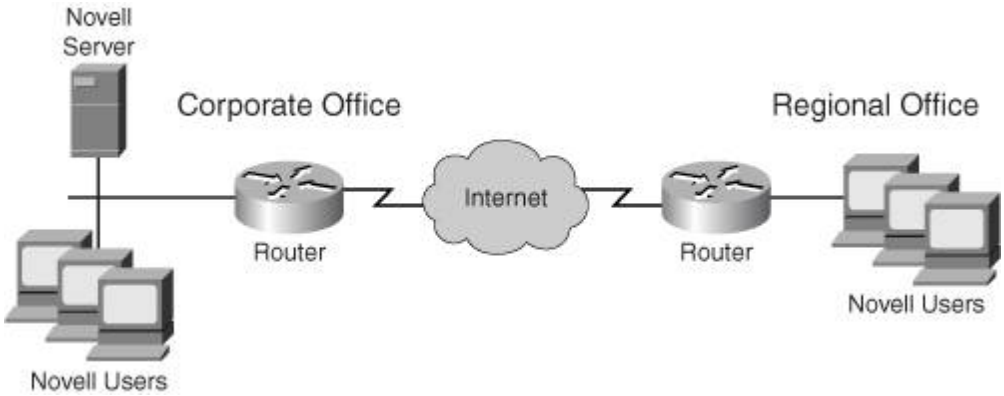
Other chapters in the book will expand on many of these VPN implementations, so the material you see in the following sections is a basic overview.

GRE

Generic Route Encapsulation (GRE) is a VPN technology originally developed by Cisco and later written up under two Internet Engineering Task Force (IETF) RFCs1701 and 2784. Cisco developed GRE as an encapsulation method to take a packet from one protocol, encapsulate it in an IP packet, and transport the encapsulated packet across an IP backbone.

Figure 1-11 to illustrates the usefulness of GRE tunneling. In many earlier networks, companies had problems moving traffic between various networking locations where the backbone of the network, or the WAN connection, didn't speak the same protocol. In Figure 1-11, two Novell IPX networks need to communicate with each other. In this example, they are connected via the Internet, which is IP-based. Without GRE, the Novell devices would have to speak both IP and IPX: IPX for access internal services and IP for the remote site's services. Cisco developed GRE specifically for this problem. GRE allows you to take a packet from one protocol, such as IPX, and encapsulate it in an IP packet (where the IP protocol number is 47, representing GRE-encapsulated information).

Figure 1-11. GRE Example



In Figure 1-11, the perimeter routers at the two sites perform this encapsulation/de-encapsulation process. From the Internet's perspective, it only sees IP packets; from the corporate and regional offices' perspectives, they see only IPX packetswhich makes everyone happy.

GRE was, and still is, used to connect small pockets of a particular protocol across an IP-based backbone without having to configure the backbone to also run additional protocols. GRE, which is a Layer-3 IP protocol, can encapsulate the following protocols:

VPNs: Choosing a Solution

As you saw in the section, "VPN Implementations," there are actually quite a few VPN solutions to choose from. You should use several criteria in selecting the correct VPN solution for your company's network. It might involve more than one solution, like IPsec and SSL, or IPsec and L2TP/IPsec.

To simplify this process, I evaluate the following criteria when choosing a VPN solution:

- Security
- Implementation, management, and support
- High availability
- Scalability and flexibility
- Cost

The following sections will cover these criteria in more depth.

Security

One of the first things I consider with a VPN solution is security. Toward that end, I ask the following questions:

- What do I need to protect?
- What kind of protection is required?
- How much protection is needed?

As shown in the preceding list, I first need to determine what is to be protected. Do I need to protect traffic for specific applications, such as e-mail, database access, file transfers, and others? Do I need to protect traffic for specific hosts? Do I need to protect traffic for specific network segments? If I only need to protect traffic for specific applications, I would probably first examine SSL VPNs to see if there is a solution available for the particular application or applications that need to be protected. Otherwise, I would look at other VPN solutions.

Second, what kind of protection is necessary? Does the traffic need to be encrypted? Do I need to perform packet integrity checking? How important is it to verify a device's identity? Once I've answered these questions, I can narrow in on a more specific VPN solution. For instance, if I need encryption, I can immediately rule out GRE.

And third, how much protection is needed? For example, if I require encryption to provide data confidentiality, how strong does the encryption process need to be? Can I use DES or must I use a much stronger encryption algorithm, like 3DES? For device authentication, can I use pre-shared keys or should I use digital certificates? Again, I use these questions to narrow my pick to the most appropriate VPN solution.

Implementation, Management, and Support

Most often network administrators forget to factor in implementation, management, and support when choosing a VPN solution. For example, if I compared SSL VPNs to IPsec, I would find that setting up, managing, and troubleshooting SSL VPNs is *much* easier when compared to IPsec.

As an example, you might only need to protect HTTP traffic. Both SSL and IPsec VPNs can do this; however, you'll have to perform a lot more work to implement, manage, and support an IPsec solution than you would an SSL solution, making the SSL a lower-cost, more scalable solution. This difference will become more apparent as you go through the IPsec and SSL components of this book on concentrators, routers, and PIX and ASA security appliances.

High Availability

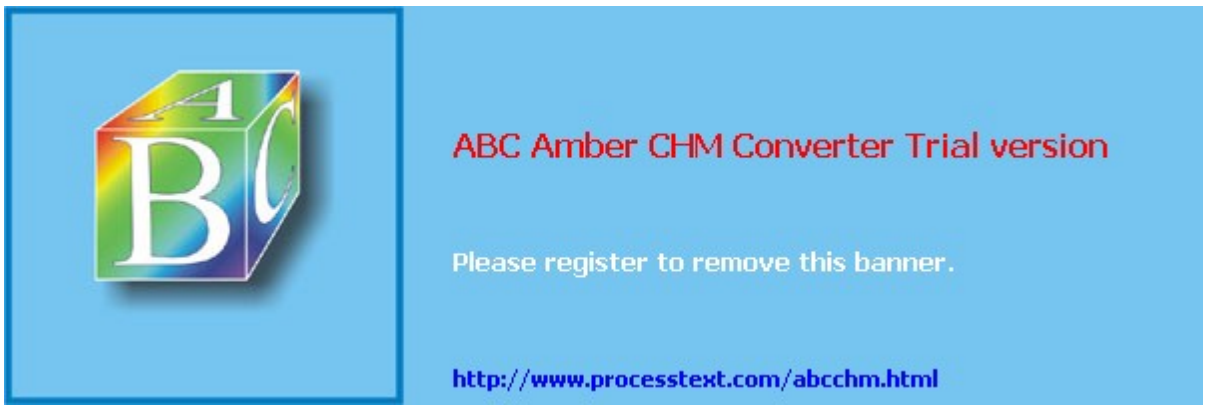
As I mentioned earlier in the "Redundancy" section of this chapter, redundancy might be a

Summary

This chapter introduced you to the concept of virtual private networks. As I mentioned in the chapter introduction, I'll expand on many of the topics and issues discussed here in future chapters.

Next up is Chapter 2, "VPN Technologies," where I delve further into technologies used by VPNs to provide protection for data, such as keys, encryption algorithms, hashing functions, and authentication methods.

PREV NEXT



Chapter 2. VPN Technologies

Before I can begin discussing VPN implementations such as Internet Protocol Security (IPsec), Layer 2 Tunnel Protocol (L2TP), Point-to-Point Tunneling Protocol (PPTP), and Secure Socket Layer (SSL), you first need an understanding of the technologies that VPNs can use to provide protection for traffic. I'm sure you have already heard of terms such as keys, DES, 3DES, MD5, pre-shared keys, and the like; however, an in-depth understanding of these protocols, algorithms, functions, and processes will help you determine the pros and cons of VPN technologies. Use this information to pick the optimal VPN implementation based on the type of technologies you'll need to protect your traffic.

I've broken this chapter into five sections: keys, encryption, packet authentication, key exchange, and authentication methods. This chapter will discuss these technologies and how they are related to VPNs, including the advantages and disadvantages of the technologies for a particular category, for example, using a pre-shared key or keys for authentication versus digital certificates.

Keys

We commonly use the term "key" in day-to-day life. One definition of a key is a tool to open a locked door, where something is kept hidden from prying eyes. In the data world, the term "key" has a similar meaning. A key is used to protect information in various ways. For instance, a data key performs a similar function as a password used to protect a user account or a PIN (personal identification number) used with your ATM card to access your bank account. Normally, the longer the key, the more secure the protection it can provide; however, this is not always the case. The following three sections will discuss how keys are used and the two types of keying algorithms: symmetric and asymmetric.

Key Usage

In network security, keys serve a multi-functional process. For example, keys are used for all of these three critical VPN functions:

- Encryption
- Packet integrity checking
- Authentication

There are two basic types of keying solutions:

- Symmetric
- Asymmetric

The following sections will talk more about these two basic types of keying implementations.

Symmetric Keys

Symmetric keys use the same single key to provide a security function to protect information. For example, an encryption algorithm that uses symmetric keys uses the same key to encrypt and decrypt information. Because the same key is used to create and verify the security protection, the algorithm used tends to be fairly simple and thus very efficient. Therefore, symmetric algorithms, like symmetric encryption algorithms, tend to work very quickly.

Because symmetric keying is very efficient and fast, it typically is used in encryption and packet integrity checking. Some encryption algorithms and standards that use symmetric keying are: DES, 3DES, CAST, IDEA, RC-4, RC-6, Skipjack, and AES. MD5 and SHA are examples of hashing functions that use symmetric keying.

One problem, however, with symmetric keying is that the two devices performing protection of data somehow have to get the same key value. For example, if two devices, RouterA and RouterB, are performing DES encryption, and RouterA generates the symmetric key for DES, RouterB also will need this same key to decrypt information that RouterA sends it. There are two basic ways to accomplish this:

- **Pre-sharing keys** You can pre-share the keys, out-of-band between the two devices.
- **Using a secure connection** You can use either an existing secure, protected connection to send keys across, or create a new protected connection to send keys across.

This last option is a "catch-22" situation, because to have a secure connection, you need keys; and to share keys, you need a secure connection. Pre-sharing keys doesn't scale very well. Later in the chapter in the "Key Exchange" section I'll discuss ways of sharing keys, dynamically, in a secure fashion between two devices without having to resort to a manual-based pre-sharing method.

Asymmetric Keys

Unlike symmetric keying, where the same key is used to create and verify the protection information, asymmetric keying uses two keys:

Encryption

Encryption is the process of transforming data in a form which is impossible to decipher without the knowledge of the key or keys used to encrypt it. Depending on the encryption algorithm, either symmetric or asymmetric keys are used. These keys were discussed in the preceding sections of this chapter.

This part of the chapter will cover the following:

- Encryption process
- Encryption algorithms

Encryption Process

The type of keying used affects how encryption is performed. For example, if you use a symmetric keying algorithm, data is encrypted and decrypted with the same key. However, with an asymmetric keying encryption algorithm, a public key is used to encrypt the data and the corresponding private key is used to decrypt it. We have seen the advantage of asymmetric key algorithms for encryption: you can easily share the public key across a public network and have a remote device use this key to encrypt data sent to you. Even if an attacker sees the public key, it won't do him any good because only the corresponding private key can decrypt the data.

However, because the complexity of the encryption/decryption algorithm makes asymmetric keying with encryption a very slow process, asymmetric keying typically is reserved for identity authentication and key sharing, and symmetric keying is used for data encryption. Because of this, the following sections will focus only on encryption algorithms that use symmetric keys.

Note

VPN devices, especially VPN gateways, commonly offload encryption processes to a hardware module to speed up the encryption and decryption of packets.

One main problem with symmetric encryption algorithms, though, is that the same key must be used on the source and destination. Sharing the key can be problematic. If you sent the key across the network to a peer, an eavesdropping attacker could see the key and be able to decrypt your messages. You could pre-share the key, but managing the periodic changing of the key for increased security causes management headaches. The "Key Exchange" section later in the chapter will examine this issue in more depth.

Encryption Algorithms

Many encryption algorithms have been developed that use symmetric keys. These algorithms include the following:

- **Data Encryption Standard (DES)** DES was developed by the National Institute of Standards and Technology (NIST). It uses a 56-bit key structure, which is the most common implementation, but weak by today's symmetric keying standards.
- **Triple DES (3DES)** 3DES is an enhanced implementation of DES, which basically uses DES three times, with three different keys, on the data that need to be protected. Because it uses three 56-bit keys, DES is commonly referred to as using 168-bit keying structure. 3DES is much stronger than DES, but is slower.
- **Advanced Encryption Standard (AES)** AES was designed to replace 3DES, providing faster and more secure encryption.
- **CAST** CAST is similar to DES and uses a 128- or 256-bit key structure. It is less secure than 3DES, but is faster.
- **International Data Encryption Algorithm (IDEA)** IDEA was developed by the Swiss Institute of Technology. It uses a 128-bit key structure; it falls somewhere between

Packet Authentication

Packet authentication is used for two purposes:

- To provide data origin authentication
- To detect packets that have been tampered with

The following sections will discuss how packet authentication works (implementation), examples of functions that are used with packet authentication, the uses of packet authentication, and issues with performing packet authentication.

Packet Authentication Implementation

Hashing functions are used to create a digital signature by taking a variable-length input, such as user data or a packet, along with a key and feeding it into a hashing function. The output is a fixed-length result. If the same input is fed into the hashing function, it will always result in the same output.

Hashing Message Authentication Codes (HMAC) functions are a subset of hashing functions. HMAC functions were developed specifically to deal with authentication issues with data and packets. HMACs use a shared secret symmetric key to create the fixed output, called a digital signature or fingerprint. Generic hashing functions have a drawback: If an eavesdropper can intercept the sent data, he can easily generate his own signature of the data and send this "doctored" information to you. HMAC overcomes this problem by using a shared secret key to create the digital signature; therefore, only the parties that know the key can create and verify the signature for sent data.

Note

With HMAC functions, if the same data and same secret key are used to generate a signature, they will always generate the same signature; if you change the data or the key value, the resulting signature changes.

Two examples of HMAC functions are MD5 and SHA, which I'll discuss in the following two sub-sections. For now, know that the basic mechanics for HMAC functions are shown in Figure 2-2. Here is an explanation of the steps shown in Figure 2-2:

1. The source takes data that needs to be protected and a key, which is shared with the destination, and runs it through an HMAC function.
2. The output of the HMAC function is a digital signature or fingerprint. In IPsec VPNs, it is also called an Integrity Checksum Value (ICV).
3. The source then takes the data that was originally fed into the HMAC function and sends the data along with the digital signature, to the destination.
4. The destination will use the same process to verify the signature; it takes the data sent by the source, along with the same shared key, and inputs this into the same HMAC function, resulting in a second signature.
5. The destination then compares the source's sent signature to the just-computed signature. If they are the same, the destination recognizes that the only device that could have created the signature was a device with the same key. The HMAC function is a one-way process: it is impossible to reverse-engineer the process by taking the fingerprint and data and coming up with the symmetric key used to create the fingerprint. If the two fingerprints are different, the destination assumes that the data was tampered with (purposefully, like by an attacker, or by accident, like data corruption or address translation) and discards the data.

Figure 2-2. HMAC Signature Creation and Verification

Key Exchange

So far I've talked about all different kinds of keys: symmetric and asymmetric keys, and encryption and HMAC keys. As I mentioned in the last two sections on encryption and packet authentication, which typically use symmetrical keys, the sharing of the protection keys is a security issue. This next section will explore the key-sharing process in more depth and talk about some possible solutions, such as the following:

- Key Sharing Dilemma
- Diffie-Hellman
- Key Refreshing
- Limitations of Key Exchange Methods

Key Sharing Dilemma

A simple example illustrates the issues of sharing keys for symmetric keying algorithms and functions. You have decided to protect financial data between two devices, PeerA and PeerB, and want to encrypt this information using a special symmetric encryption algorithm. PeerA has decided that the encryption key should be "If you can guess this key, you win a lollipop!" Now the issue is to get this key to PeerB so that it can decrypt your financial data successfully. I'll discuss some possible solutions in the following subsections.

Pre-Share the Key

One solution might be to take the key on PeerA and share it, out-of-band, with PeerB. For instance, you could copy the key to a floppy disk or write it down on a piece of paper and then mail this to PeerB. Or you could just call up PeerB and tell him what the key is over the phone. Both of these are referred to "out-of-band" sharing, since the actual key is not shared across the data network.

One downside to pre-sharing keys is that it doesn't scale very well. For example, if you had 100 peers you needed to share keys with, it would take quite a while to perform this process. At a minimum, you would want to change keys every time an employee left the company who knew the pre-shared keys. Also, you would want to change the keys periodically to make your encryption process more secure, just in the off chance your encryption key was compromised. If your security policy stated that you needed to change the encryption key every hour, this solution would not be feasible because of the out-of-band delay involved in getting the keys between the two peers.

Use an Already Encrypted Connection

If you would attempt to share the encryption key in-band with PeerB, like using Telnet to access PeerB, the key would be susceptible to an eavesdropping attack. One solution is to use an already encrypted connection. You could use a secure program to do this, such as SSH; however, SSH requires keys that would have to be exchanged in a secure fashiona catch-22 situation, because you would need a secure connection to share the SSH keys in the first place.

Encrypt the Key with an Asymmetric Keying Algorithm

I've already (indirectly) discussed one way to solve the symmetric key-sharing issue: use asymmetric encryption. If you recall from the "Asymmetric keys" section, asymmetric keying uses two keys: a public and private key. These keys have a symbiotic relationship with each other and a special algorithm is used to generate these keys. When used for encryption purposes, each peer would generate a public/private key pair. Then each peer would share their public keys with each other. When PeerA would want to send traffic to PeerB, PeerA would use PeerB's public key to encrypt data and then send it to PeerB. PeerB would then use its own private key to decrypt the data.

In our example of financial data, I'll assume that each transfer is 100 MB in size, and this happens every 15 or 20 minutes. This is a large amount of data; asymmetric encryption algorithms are very slow and process-intensive; so they are not desirable when large amounts of data need to be transferred.

Authentication Methods

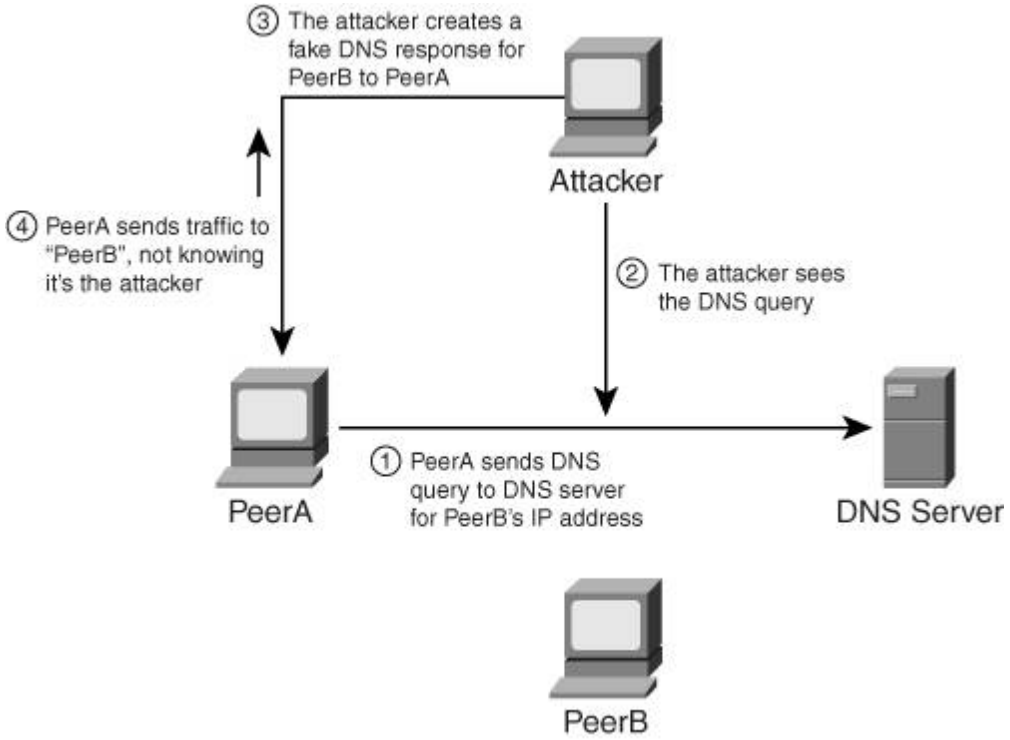
Authentication is implemented using digital signatures. Digital signatures are most commonly created by taking some message text, such as information unique to a device or person, along with a key, through a hashing function. The digital signature is like the signature that you would use to sign a check, your fingerprint, or a retinal scan of your eye: it's something unique to you and no one else. Digital signatures are used to implement non-repudiation in VPNs: being able to prove, with certainty, the identity of a device.

The last part of this chapter on VPN technologies will cover authentication methods: how two peers can recognize that when they establish a connection to each other, they are really connecting to the associated peer and not someone pretending to be that peer. In this section I'll explore further how man-in-the-middle attacks occur and the types of authentication you can use to discover and prevent man-in-the-middle attacks.

Man-in-the-Middle Attacks

So that you have a better understanding of a man-in-the-middle attack, I'll use Figure 2-5 to illustrate how this attack occurs. In this example, PeerA wants to send data to PeerB. PeerA does a DNS lookup for PeerB's address, shown in Step 1. However, the attacker also sees the DNS request and sends a reply back to PeerA before the DNS server has a chance, shown in Steps 2 and 3. The IP address that the attacker sends is the attacker's own IP address. PeerA knows no better and assumes that when it uses the IP address in the DNS reply that it is sending traffic to PeerB; however, as shown in Step 4, the traffic actually is directed to the attacker.

Figure 2-5. Man-in-the-Middle Attack Example



This is a simple example of using spoofing of DNS replies. If the DNS server's reply was received before the attacker's, PeerA would connect to PeerB; however, a sophisticated hacker could use a session hijacking/re-routing attack to redirect traffic sent from PeerA to PeerB to the attacker himself, still pulling off the man-in-the-middle attack. Given this security problem, some type of authentication is required to allow PeerA and PeerB to verify their identities when communicating with each other. The next section will discuss different types of authentication that might solve this problem.

Authentication Solutions

With VPN implementations, you can use two types of authentication to verify a peer's

Summary

This chapter showed you the basics of VPN technologies. Understanding the technologies that VPNs use to protect information is useful when attempting to understand how a VPN implements its protection methods. A VPN implementation will use many of the technologies discussed in this chapter.

Next is Chapter 3, "IPsec," where I introduce the interworkings of the IPsec standard for VPNs. Many of the things I discussed in this chapter are used by IPsec to implement protected connections between networks or devices.

PREV NEXT

Chapter 3. IPsec

IP Security, or IPsec for short, is a framework of standards that provides the following key security features at the network layer between two peer devices:

- Data confidentiality
- Data integrity
- Data authentication
- Anti-replay detection
- Peer authentication

The Internet Engineering Task Force (IETF) defines the standards for IPsec in various RFCs. Because it provides network layer protection between devices or networks, and because it is an open standard, it is commonly used in today's networks that use IPv4 and IPv6.

This chapter will explore many of the standards that IPsec uses to provide a secure transport for communication. I'll first cover the standards used, and then discuss how these standards are implemented by IPsec in the "ISAKMP/IKE Phase 1" and "ISAKMP/IKE Phase 2" sections. As you will see in the chapter, vendors (such as Cisco), have a tendency to *enhance* the standards to overcome problems that IPsec can experience in data networks. Cisco, for example, has added many features to enhance both LAN-to-LAN (L2L) and remote access sessions. I'll discuss many of these features at the end of this chapter.

IPsec Standards

There are many different ways of implementing VPNs, as discussed in Chapter 1. For example, you could use an SSL VPN solution to protect data between two devices; however, the main limitation of SSL VPNs is that they primarily provide application layer protection, which typically is limited to web browser-based connections, and specific applications the vendor has written code for, to tunnel through the SSL VPN. Their advantage, however, is that they can use an existing web browser on the user's desktop.

IPsec, on the other hand, provides protection at the network layer; therefore, any IP traffic can be protected between peer devices. But IPsec is intrusive on the client (remote access) side; typically you need additional software installed on your device to provide the network-layer protection.

Note

Depending on the vendor's gateway product, you might be able use the Microsoft L2TP/IPsec VPN client, which is pre-installed on Microsoft Windows 2000 and XP desktops.

IPsec's framework is defined in RFC 2401; however, the implementation of IPsec is defined across quite a few different RFCs. As I mentioned in the chapter introduction, IPsec provides the following services:

- **Data confidentiality** This is done via encryption to protect data from eavesdropping attacks; supported encryption algorithms include DES, 3DES, and AES.
- **Data integrity and authentication** This is done via HMAC functions to verify that packets haven't been tampered with and are being received from a valid peer; in other words, to prevent a man-in-the-middle or session hijacking attack. Supported HMAC functions include MD5 and SHA-1.
- **Anti-replay detection** This is done by including encrypted sequence numbers in data packets to ensure that a replay attack doesn't occur from a man-in-the-middle device.
- **Peer authentication** This is done to ensure that before data is transmitted between peers, the peers are "who they say they are." Device authentication is supported with symmetric pre-shared keys, asymmetric pre-shared keys, and digital certificates. remote access connections also support user authentication using XAUTH, short for extended authentication.

The two main groupings of standards that IPsec uses are:

- **ISAKMP/IKE/Oakley/SKEME** These standards are used to set up a secure *management* connection, determine keying information for encryption, and use signatures for authentication of the management connection. This connection is used so the two IPsec peers can share IPsec messages with each other.
- **AH and ESP** These standards are used to provide protection for user data. They can provide for confidentiality (only ESP), data integrity, data origin authentication, and anti-replay services. I like to refer to these connections as *data* connections.

The following sections will provide an overview of the RFCs used by IPsec, including the standards mentioned in the above bullets. Specifically, they cover the following:

- IETF RFCs
- IPsec Connections
- Basic Process of Building Connections

IETF RFCs

IPsec is a framework of standards defined in IETF RFCs. This section provides a quick overview of these RFCs:

ISAKMP/IKE Phase 1

As you saw in the list in the preceding section, some of the same steps are performed in both the site-to-site and remote access IPsec setup; however, remote access has quite a few additional steps. In this current section, I'll expand on the IPsec setup steps and cover them in more depth.

ISAKMP and IKE work together to establish secure connectivity between two devices. ISAKMP defines the message format, the mechanics for a key exchange protocol, and the negotiation process to build connections. ISAKMP, however (as already mentioned), doesn't define how keys are created, shared, or managed for protecting the secure connections; IKE is responsible for this.

To help you understand the actual details of how an ISAKMP/IKE Phase 1 management connection is established, this part of the chapter covers the following:

- The Management Connection
- Key Exchange Protocol: Diffie-Hellman
- Device Authentication
- Remote Access Additional Steps

The Management Connection

The management connection is established in Phase 1. This connection uses UDP on port 500 for communication. It is a bidirectional connection, and both peers can use it to share IPsec messages with each other.

Note

The ISAKMP/IKE connection uses UDP. The source and destination port are 500; however, I have seen some vendors use a random source port number greater than 1,023 instead of 500.

No matter if the session is a site-to-site or remote access session, three things will occur during ISAKMP/IKE Phase 1:

1. The peers will negotiate how the management connection will be protected.
2. The peers will use Diffie-Hellman to share keying information to protect the management connection.
3. The peers will authenticate each other before ISAKMP/IKE Phase 2 can proceed.

ISAKMP/IKE Phase 1 is basically responsible for setting up the secure management connection. However, there are two modes for performing these three steps:

- Main
- Aggressive

The following two sections will cover these two modes, followed by a section that discusses how you can specify the policies that will be used to protect the management connection.

Main Mode

Main mode performs three two-way exchanges totaling six packets. The three exchanges are the three steps listed in the last section: negotiate the security policy to use for the management connection, use DH to encrypt the keys for the encryption algorithm and HMAC function negotiated in Step 1, and perform device authentication using either pre-shared keys, RSA encrypted nonces, or RSA signatures (digital certificates).

Main mode has one advantage: the device authentication step occurs across the secure

ISAKMP/IKE Phase 2

All of the things discussed in the last section only cover the setup of the management connection. No user data actually traverses this management connection; only ISAKMP/ IKE messages traverse this management connection. This section will discuss how the protected user data connections are built by covering the following:

- ISAKMP/IKE Phase 2 Components
- Phase 2 Security Protocols
- Phase 2 Connection Modes
- Phase 2 Transforms
- Data Connections

ISAKMP/IKE Phase 2 Components

ISAKMP/IKE Phase 2 only has one mode: Quick mode. Quick mode defines how protected data connections are built between two IPsec peers. Quick mode has two main functions:

- Negotiate the security parameters to protect the data connections.
- Periodically renew the keying information for the data connections (basically rebuilding the connections).

ISAKMP/IKE Phase 2 has one unique characteristic: there are actually *two* unidirectional data connections built between the two peers. For example, PeerA would have a data connection to PeerB and PeerB would have a separate data connection to PeerA. Because these connections are separate connections, the security parameters negotiated could be different between the two peers. For example, the PeerA-to-PeerB connection could use 3DES for encryption, but the PeerB-to-PeerA connection could use DES. However, this is commonly not done: the same security parameters typically are used for both data connections.

The following are policies that need to be determined to configure your devices to build ISAKMP/IKE Phase 2 connections:

- Which data traffic should be protected between the two peers? With site-to-site connections, this is either defined statically or learned dynamically; with remote access connections, this is determined by the split tunneling policy defined on the VPN gateway.
- What security protocol(s) should be used to protect the traffic? The two protocols defined by IPsec are AH and ESP.
- Based on the security protocol(s) selected, how should the data traffic be protected? For example, what HMAC function or encryption algorithm should be used?
- What mode of operation should the security protocols use? The two operation modes are tunnel and transport.
- When refreshing keying information, should the ISAKMP/IKE Phase 1 management be used to share the new keys or should perfect forward secrecy be used instead?
- What's the lifetime of the data connections? This can be based on time expired or amount of data transmitted across the connections.

The following sections will discuss this information in more depth.

Phase 2 Security Protocols

IPsec can use one or two security protocols to protect the data transmitted across the data connections built in ISAKMP/IKE Phase 2:

- AH

IPsec Traffic and Networks

There are two important issues you might have to deal with concerning the sending and receiving of IPsec traffic:

- Address translation
- Firewalls
- Other IPsec issues

The following three sections will discuss these issues in more depth.

IPsec and Address Translation

Address translation translates addressing and, possibly, port information inside IP, TCP, or UDP headers. My goal here is to not discuss what address translation is; I'm assuming you are already familiar with it. My other Cisco Press book, *Cisco Router Firewall Security*, covers the mechanics of address translation on Cisco routers, if you're not that familiar with the process.

As you already know, there are three IPsec connections between two IPsec peers:

- A management connection, which uses UDP port 500
- Two data connections, which use AH (protocol 51) or ESP (protocol 50)

The following two sections will discuss issues with address translation and IPsec, and possible solutions.

Address Translation Issues

Devices performing address translation typically don't create any issues for the management connection, but can cause problems with the data connections. However, some earlier PAT devices would force all management connections to use UDP port 500 for both the source and destination, which could cause problems for multiple simultaneous connections to the same endpoint through the PAT device; this shouldn't be a problem with most translation devices today, though.

As I mentioned in the "Phase 2 Security Protocols" section, AH breaks completely when any type of address translation is performed on an AH packet. PAT isn't supported because PAT requires an outer TCP or UDP headerand AH is a Layer-3 protocol that lacks this. AH also won't work with NAT because AH includes most of the fields in the entire IP packet in the input of the HMAC function. Therefore, if the source or destination address is changed, this would invalidate the ICV signature by the destination IPsec device.

ESP actually works with NAT, since the outer IP header isn't included in the ICV computation for the digital signature; only the ESP and user data parts are included. However, like AH, ESP doesn't work with PAT because PAT requires the use of TCP or UDP in the outer header and ESP is a Layer-3 protocol that lacks this functionality.

Address Translation Solutions

Given the above issues with address translation, there are some solutions available to you assuming you are willing to forgo the use of AH and stick with only ESP. Remember that ESP will work with NAT, but breaks with PAT. To have ESP interoperate with devices performing PAT, you could insert a TCP or UDP header *between* the outer IP header and the ESP part, as shown in Figure 3-8. By inserting either a TCP or UDP header, you make the address translation device happy and it can perform PAT; plus, because this information is in the outer header, it is not included in the HMAC process to create the digital signature (only the ESP header and user data are included).

Figure 3-8. ESP Address Translation Solutions

Summary

This chapter introduced you to an overview of IPsec, including its components and how it implements a secure connectivity solution. Of all of the VPN implementations, IPsec is probably the most complicated one.

Setting up and troubleshooting an IPsec connection is not necessarily a simple process. Therefore, understanding how IPsec works and the types of connections that are built is important when it comes time to troubleshoot connections that won't come up. Whereas this chapter gives you an overview of the IPsec standard and implementation, other chapters will spend more time on the actual configuration and troubleshooting process, because this is different on a product-by-product basis.

Next up is Chapter 4, "PPTP and L2TP," where I discuss an overview of PPTP and L2TP VPN implementations so that you have an understanding as to which VPN implementation, compared to IPsec, makes more sense for your particular network infrastructure. Following this chapter, Chapter 5 will cover the last type of VPN implementation I'll cover in this book: "SSL VPNs."

PREV NEXT

Chapter 4. PPTP and L2TP

Even though IPsec is very popular in the marketplace as far as VPN implementations go, there are many other VPN implementations to choose from. For example, if you come from a Microsoft shop, you probably will be exposed to Microsoft's VPN implementations: the Point-to-Point Tunneling Protocol (PPTP) and the Layer-2 Tunneling Protocol (L2TP). Like IPsec, both of these protocols have been standardized by IETF and ratified into RFCs.

PPTP is a simple-to-implement VPN solution defined in IETF's RFC 2637. It allows for the use of user authentication (before IPsec included this with XAUTH) and the ability to be interoperable with NAT, making it, originally, a preferred remote access solution over IPsec, which lacked these features. IPsec was not designed as a remote access solution, originally lacking features such as user authentication, address assignment, and enforcement of user policies; nor does IPsec support the transport of multiple protocols (only IP is supported) or multicast transmissions. Therefore, PPTP had a lot of support from the user community where secure remote access connectivity was needed.

L2TP is, rather, a merging of two standards: PPTP and the Cisco Layer-2 Forwarding (L2F) protocol. L2TP is defined in IETF's RFCs 2661 and 3438. L2TP can work in conjunction with IPsec to leverage the remote access advantages that L2TP provides, but also the security that IPsec offers. In today's world, you'll typically see Microsoft shops using L2TP over IPsec (L2TP/IPsec) for remote access connectivity. This chapter will be devoted to these protocols, providing an overview of how they work.

PPTP

PPTP originally was developed by Microsoft to provide a secure remote access solution where traffic needed to be transported from a client, across a public network, to a Microsoft server (VPN gateway). One of the interesting items about PPTP's implementation is that it is an extension of the Point-to-Point Protocol (PPP). Because PPTP uses PPP, PPTP can leverage PPP's features. For example, PPTP allows the encapsulation of multiple protocols, such as IP, IPX, and NetBEUI, via the VPN tunnel. Also, PPP supports the use of authentication via PAP, CHAP, and MS-CHAP. PPTP can use this to authenticate devices.

PPTP recently added support for the Extensible Authentication Protocol (EAP) to authenticate users. EAP was designed for wireless networks, but has been incorporated into PPTP. Even though PPP is used in dialup scenarios, PPTP doesn't require the use of dialup to establish remote access connections; you can use dialup or your local LAN connection.

Note

If you're in a pure Microsoft shop and using a Windows remote access server (RAS) to terminate remote access connections, you'll typically be using PPTP or L2TP/IPsec for your client connections. PPTP is typically used with the older Microsoft Windows platforms, Windows 95, 98, or ME. For the newer Microsoft platforms, 2000, XP, and 2003, L2TP/ IPsec is more commonly used.

Here is a quick review of some of PPTP's features:

- **Compression** Compression of data is handled by Microsoft's Point-to-Point Compression (MPPC) protocol within the PPP payload. This is supported by both PPTP and L2TP and normally enabled for dialup clients.
- **Encryption** Encryption of data is handled by Microsoft's Point-to-Point Encryption (MPPE) protocol within the PPP payload. The encryption uses RSA's RC4 encryption algorithm. PPTP uses this method, whereas L2TP uses IPsec, which is more secure. With MPPE, the initial key created during user authentication is used for the encryption algorithm and is regenerated periodically.
- **User authentication** User authentication is achieved using PPP's authentication methods, such as PAP or CHAP, and others, such as EAP. MPPE support requires the use of MS-CHAPv1 or v2. If you use EAP, you can choose from a wide range of authentication methods, including static passwords and one-time passwords (through the use of token cards).
- **Data delivery** Data is packetized using PPP, which is then encapsulated in a PPTP/ L2TP packet. By using PPP, PPTP can support multiple transport protocols, such as IP, IPX, NetBEUI, and others.
- **Client addressing** PPTP and L2TP support dynamic addressing of the client using PPP's Network Control Protocol (NCP). As mentioned in the last chapter, the Cisco IKE Mode Config supports a similar function.

To help you understand the inner workings of PPTP, the following sections cover these topics:

- Review of PPP
- PPTP Components
- How PPTP Works
- Issues with the Use of PPTP

PPP Review

Because PPTP leverages the use of PPP, I'll briefly review PPP in this section. PPP originally was designed to transmit data across dialup or point-to-point links. It supports the encapsulation of many Layer-2 and Layer-3 protocols as a transport.

L2TP

L2TP is a combination of PPTP and L2F. It is defined in RFCs 2661 and 3438. L2TP took the best of both PPTP and L2F and integrated them into a single protocol. Like PPTP, L2TP uses PPP to encapsulate user data, allowing the multiple protocols to be sent across a tunnel. L2TP, like PPTP, extends the PPP protocol. As an additional security enhancement, L2TP can be placed in the payload of an IPsec packet, combining the security advantages of IPsec and the benefits of user authentication, tunnel address assignment and configuration, and multiple protocol support with PPP. This combination is commonly referred to as L2TP over IPsec or L2TP/IPsec. The remainder of this chapter is devoted to an overview of L2TP, how it is implemented, and the advantages it has over PPTP.

L2TP Overview

L2TP, like PPTP, encapsulates data in PPP frames and transmits these frames across an IP backbone. Unlike PPTP, L2TP uses UDP as an encapsulation method for both tunnel maintenance and user data. Whereas PPTP uses MPPE for encryption (this is negotiated via PPP), L2TP relies on a more secure solution: L2TP packets are protected by IPsec's ESP using transport mode. Even though you can use L2TP without IPsec, the main issue with this approach is that L2TP itself doesn't perform encryption and therefore needs to rely on something else. Therefore, most L2TP implementation will include the use of IPsec.

Tip

The only time I would use L2TP without IPsec was if I was trying to troubleshoot a connection where I didn't know if the problem was IPsec-related or L2TP-related; by removing IPsec from the equation, I could focus on just the L2TP VPN component.

L2TP is a remote access solution. It consists of two devices: a client and a server. Tunnel maintenance and the tunnel data used between these two devices use the same packet structure, simplifying L2TP's implementation.

Until IPsec introduced XAUTH for user authentication, the only standards-based method of performing user authentication was PPTP and then L2TP. And even now, XAUTH is still in an IETF-RFC draft state, so one vendor's implementation might not be compatible with another's. One of the main concerns with remote access VPNs is who actually is using the device to obtain connectivity. Certificates and pre-shared key information typically are stored on the client; therefore, if someone gains access to the client device, or steals the client device, authentication is defeated. Authenticating the user who is actually using the device is important, which is why many VPN implementations authenticate the users themselves. L2TP allows for many types of user authentication including legacy methods such as PAP, CHAP or MS-CHAP, and current authentication services such as EAP (and its derivatives).

As I mentioned in the last chapter, Cisco uses a proprietary method of assigning addressing and policy information to remote access clients during IKE Mode Config in IPsec. L2TP has standardized this process by using DHCP to handle address assignment. DHCP supports standardized features such as address assignment, failover, address pool management, and even address authentication.

Note

If you work in a Microsoft-only shop, your VPN implementation of choice will be L2TP/ IPsec. Microsoft pushes this standard heavily over PPTP or IPsec. As mentioned in the last chapter, many vendors enhance IPsec for remote access connectivity, including Cisco. This can create vendor interoperability problems if you want to use one vendor's solution for the client side and another for the server side. By using L2TP/IPsec for remote access VPNs, you can feel fairly safe that no matter what vendor you're using for the client or server, they should be compatible with each other.

L2TP Operation

This section provides an overview of the components of L2TP and how it works. I'll quickly

Summary

This chapter showed you the basics of L2TP and PPTP VPNs. Both have their roots in Microsoft's VPN initiative. PPTP was Microsoft's first VPN implementation, and mostly was an enhanced version of PPP. However, PPTP lacked many security features and thus was combined with the Cisco L2F into L2TP.

Because L2TP doesn't provide for security services such as confidentiality and packet authentication and validation, IPsec is used as a transport. I'll discuss more about how to implement L2TP/IPsec and PPTP remote access VPNs in Chapter 8, "Concentrator Remote Access Connections with PPTP, L2TP, and WebVPN" and Chapter 13, "Windows VPN Client."

Next up is Chapter 5, "SSL VPNs," in which I discuss SSL and its use in implementing VPNs.

PREV NEXT

Chapter 5. SSL VPNs

Chapter 3 discussed IPsec VPNs, and Chapter 4 discussed PPTP and L2TP VPNs. All three of these VPN implementations provide network layer protection; they can protect traffic from the network layer and higher. However, one of their downsides is that they require special software to be installed on the client device, and possible user training on how to use the software.

Some companies want a solution that is more simple to use and more easy to maintain than the three I just mentioned. Secure Socket Layer (SSL) began as a protocol to protect web (HTTP) traffic between an end-user device and a web server. Normally, it is used to provide protection for online purchases and identity information at e-commerce sites such as Cisco Press and Cisco. However, many network vendors are leveraging SSL's capabilities and using SSL to implement VPN solutions. One main advantage that SSL VPNs have over the other three is that SSL VPNs require no VPN software, by default, to be installed on the user's desktop; a currently installed web browser is used. Using a web browser allows a user to access a central site securely from both corporate and non-corporate PCs. The remainder of this chapter will focus on the use of SSL for VPN implementations.

SSL Overview

SSL is an emerging VPN implementation in the marketplace. It was designed for remote access solutions and does not provide site-to-site connections. SSL VPNs provide secure access primarily to web-based applications. Because SSL uses a web browser, users typically do not have to load any special client software on their desktops.

SSL VPNs operate at the session layer of the OSI Reference Model. And because the client is a web browser, only those applications that support a web browser, by default, will work with a VPN solution. Therefore, applications such as Telnet, FTP, SMTP, POP3, multimedia, IP telephony, remote desktop control, and others don't work with SSL VPNs because they don't use a web browser for their front-end user interface. Of course, many vendors use either Java or ActiveX to enhance SSL VPNs by supporting non-HTTP applications, POP3 and SMTP e-mail clients, and Microsoft Windows file and print sharing. For example, the Cisco SSL VPN implementation supports non-web-based applications such as Citrix, Windows Terminal Services, and many others. Plus, some vendors use Java or ActiveX to deliver other SSL VPN components, such as additional security functions for removing any traces from the PC of a user's activity after the SSL VPN has been terminated, in addition to others. Cisco refers to their SSL VPN implementation as *WebVPN*.

Note

Be aware that each networking vendor has a list of non-HTTP applications that it supports. Therefore, you should scrutinize a vendor's SSL VPN offering carefully and compare this to your users' needs before choosing between different SSL VPN vendors. In some cases, a vendor might even offer full tunneling of network-level traffic via an SSL VPN, making the solution more similar to the capabilities of a Layer-3 VPN implementation such as IPsec.

SSL Client Implementations

A main reason that network administrators like the appeal that SSL VPN implementations offer is that SSL VPNs don't require any special kind of VPN client software to be pre-installed on the user's desktop. Of course, the user does need some software, like an SSL-enabled web browser, typically with either Java or ActiveX enabled; and the user probably already has these things installed from an initial desktop installation.

There are three general types of SSL client implementations:

- Clientless
- Thin client
- Network client

Because only a web browser is required on the user's desktop, the SSL VPN client is commonly referred to as "clientless" (not quite true, of course) or "webified." Therefore, SSL VPNs are sometimes called clientless VPNs or Web VPNs when only a web browser is used for the SSL VPN. The main drawback of a clientless VPN is that only web-based traffic can be protected.

A thin client typically is Java or ActiveX software downloaded via the SSL VPN to the user's desktop. It allows a small subset of non-web applications to be transported across the SSL VPN. The process of transporting non-web applications across an SSL VPN is sometimes referred to as "port forwarding." The initial Cisco WebVPN implementation supported this feature.

For network-based access, an SSL client is required to be installed on the user's desktop; however, this typically is downloaded dynamically to the user's desktop when he establishes the initial SSL VPN to the central site. With network-based access, most network-layer traffic can be protected by the SSL VPN, which is similar to what other network-based VPN implementations do. Because an SSL client needs to be installed to provide network-level protection, the user must have administrative rights on the PC he is using; without administrative rights, the user will be restricted to a thin client or clientless connection, depending on the vendor's SSL VPN implementation.

When to Use SSL VPNs

Now that you have a basic understanding of the operation and use of SSL VPNs, I'll discuss when SSL VPNs make sense. Normally, I would consider using clientless SSL VPNs when some or all of the following are true:

- Users use a web browser to access and interact with their applications.
- Users need to access only a very limited number of company resources and not a wide range of hosts and services.
- As an administrator, you have very little or no control over the user's desktop and the software that is or isn't installed on it.
- Users need only occasional access to the network, where sometimes this access is not from their PC but a public device such as an unmanaged PC (PC at an Internet café or an airport or library kiosk).

If you have a small number of non-web applications, in addition to web applications, that you want to protect to a central site, using a thin client SSL VPN would be a good solution; however, you'll need to ensure that for the non-web applications you want to tunnel, the list of vendors you're examining support these.

If you're interested in reducing the amount of management you'll need to perform on a user's desktop but want to be able to protect network-level traffic to the corporate site, I might consider a network SSL VPN. One main disadvantage of this solution, though, is that you might not be able to implement any type of split tunneling policies, and thus the user might be able to access resources from his PC in clear text.

The following two sections discuss the advantages and disadvantages of SSL VPNs, followed by a quick comparison between SSL VPNs and IPsec VPNs.

Advantages of SSL VPNs

SSL VPNs are ideal for those users who use web browsers on a daily basis for interacting with a company's applications. Here is a brief list of the advantages of SSL VPNs:

- No additional software typically needs to be installed on the users' desktops.
- You can access applications securely from anywhere; you only need a device with a web browser.
- A wide variety of web browsers are usually supported.
- Little training is required for your users.
- Users typically can be authenticated by several methods, including static passwords, certificates, directory services, and token card solutions. With directory services, a single login process authenticates the user to the SSL gateway in addition to authenticating to the directory service.
- SSL VPNs work with address translation because SSL/TLS uses TCP, and only the TCP payload is protected. Because SSL provides protection for the payload in the TCP segment, and not the outer TCP segment header, SSL VPN traffic can traverse NAT or PAT devices. This is one headache of most Layer-3 VPN implementations; their data tunnels typically are Layer-3 tunnels. Sometimes NAT works for Layer-3 VPNs, but PAT breaks the tunnel connection. Of course, as I mentioned in Chapters 3 and 4, there are ways of getting around this problem for Layer-3 VPN solutions; however, these add more overhead and make the Layer-3 VPN solution less efficient than a clientless SSL VPN. However, if you'll be using a thin or network client, the amount of overhead to tunnel traffic is comparable to IPsec.

Disadvantages of SSL VPNs

Given their advantages, SSL VPNs do have their limitations and disadvantages. This section will explore a few of them. Web applications use TCP port 80 for their connections. Encrypted

Cisco WebVPN Solution

Because the implementation of an SSL VPN typically is unique on a vendor-by-vendor basis, I won't go into the particulars as to how each different SSL VPN vendor implements SSL VPNs. However, since this book focuses on VPNs and Cisco products, this is a good chapter to introduce how Cisco implements SSL VPNs in their *WebVPN* solution. The remainder of this chapter will focus on the Cisco WebVPN solution. This part of the chapter will provide a brief overview, whereas Chapter 8, "Concentrator Remote Access Connections, with PPTP, L2TP, and WebVPN" will discuss how to set them up on Cisco VPN 3000 series concentrators. Chapter 18, "Router remote Access Connections," discusses setup on Cisco routers.

VPN 3000 Series Concentrators

The VPN 3000 concentrators were originally the only Cisco product that supported WebVPN functionality for remote access; Cisco has added WebVPN to the ASA security appliances and the IOS-based routers. The VPN 3000 concentrators also support IPsec, PPTP, and L2TP/IPsec VPNs for site-to-site and remote access connections. As you will see in the next chapter, Cisco has a wide range of 3000 series hardware platforms to choose from. The 3000 series concentrators also support many types of authentication for remote access users (like WebVPN users), including static usernames and passwords, AAA RADIUS, Microsoft NT Domain, Microsoft Active Directory, RSA SDI (token cards), and X.509 digital certificates.

The Cisco WebVPN implementation was introduced in Version 4.1 of the 3000 series software. This implementation was enhanced with the acquisition of Twingo in March of 2004 and added to the 4.7 software, where this feature is referred to as Cisco Secure Desktop (CSD). Cisco integrated Twingo's technology into the 4.7 software to provide for enhanced services, such as technology to reliably eliminate all traces of sensitive data from the user's device after the SSL VPN terminates. This includes clearing out history and temporary files, caches, cookies, e-mail attachments, and other downloaded data. This feature is important where a user is using a public device for the secure access.

WebVPN Operation

The operation of WebVPN is straightforward. The user initiates an SSL connection to the public interface of the 3000 series concentrator: in the web browser address field, the URL would look something like this:

`https://IP_address_of_concentrator`

The digital certificate of the concentrator must be accepted by the client if the WebVPN server's certificate is self-signed; otherwise, if a CA is used and the WebVPN server obtains its certificate from a CA, the user's web browser will accept the server's certificate automatically. If digital certificate authentication is enabled for user devices, the WebVPN gateway validates the user's certificate. At this point, a system integrity scan can be accomplished using CSD. CSD can determine if the remote system is a corporate-owned asset based on information included on a user's certificate, a file/hash value, or a particular register entry. After this, CSD can then verify that the user's desktop meets any personal firewall or anti-virus policy defined on the WebVPN server. Once the SSL VPN is established and the virtual desktop starts (assuming it's a network-based SSL VPN), CSD ensures that all data for the duration of the session is encrypted and protected from being left on the system once the SSL VPN is terminated or times out.

Note

The use of CSD assumes that the user is initiating the SSL VPN from a corporate-owned PC; otherwise, a clientless SSL VPN solution normally will be used where the use of CSD will probably not work (CSD requires many administrative rights the user will probably not have on a non-corporate PC).

If using a clientless SSL VPN, an initial login page is then presented to the user. At this point the user is actually using the WebVPN tunnel; however, users first must authenticate their identities by supplying a username and password. This information is either validated on the concentrator itself or on an external security server (these were discussed in the "VPN 3000

Summary

This chapter introduced you to SSL VPNs. As you have seen, clientless, thin client, and network client SSL VPNs don't provide a complete VPN solution, but solve specific problems related to secure connectivity. However, network client solutions come close to providing a solution like IPsec remote access solutions. SSL VPNs are great when most of a user's access to the corporate Internet is via a web browser, but when many non-web browser-based applications are being used by the user, a Layer-3 VPN implementation, like IPsec, is probably a better choice for a solution.

Next up is Part II, "Concentrators," where I show you how to use the Cisco 3000 series concentrators as VPN gateway solutions. I'll discuss how to use them for both remote access and site-to-site connectivity, and how to troubleshoot VPN connections that terminate on them.

PREV NEXT

Part II: Concentrators

PREV NEXT

Chapter 6. Concentrator Product Information

Cisco offers a wide range of VPN products, including routers, PIX and ASA security appliances, VPN hardware appliances (concentrators), and software and hardware clients. With this wide range of choices, Cisco provides a VPN solution that can fit any network or market. VPN implementations supported by these products include IPsec, L2TP over IPsec, PPTP, and WebVPN.

A few years back Cisco introduced a concept called "Easy VPN." Easy VPN's premise was to make it easy to set up remote access connections. When I examine the client functionality of Cisco client products throughout this book, you'll see that setting them up as clients is a very simple processthus the term "easy" in Easy VPN. Easy VPN allows administrators to deploy complicated VPN technologies without the configuration headaches associated with them.

Two components of Easy VPN are as follows:

- **Easy VPN Server** Products that support the server functions include the VPN 3000 concentrators, IOS routers, and PIX and ASA security appliances.
- **Easy VPN Remote** (sometimes referred to as Client) Products that support client functions include SOHO IOS routers, SOHO PIX firewalls, the Cisco VPN 3002 hardware client, and the Cisco VPN software client.

This chapter will focus on one set of Easy VPN components: Cisco 3000 series concentrators. Specifically, it will cover:

- Concentrator Models
- Concentrator Modules
- Concentrator Features
- Concentrator Access Introduction

Concentrator Models

The Cisco VPN 3000 series concentrators, commonly referred to as VPN hardware appliances, originally were built by Altiga. Cisco acquired Altiga in 2000. These concentrators were built primarily to handle large numbers of remote access sessions, but they also support site-to-site connectivity.

Of all of the Cisco VPN offerings, the Cisco VPN 3000 series concentrators provide the most flexible and scalable remote access solution: IPsec, L2TP over IPsec, PPTP, and WebVPN VPN implementations are supported. And Cisco has added many enhanced features to their concentrators to meet their customers' remote access needs. Cisco currently sells six different models of the 3000 series:

- **3005** small branch office
- **3015** small branch office
- **3020** medium branch office or small company
- **3030** small company or medium corporation
- **3060** medium or enterprise corporation
- **3080** enterprise corporation or ISP

The 3005 can perform VPN functions only in software, whereas the other concentrators support Scalable Encryption Process (SEP) modules that can perform VPN functions in hardware. SEP modules are upgradeable and can be added easily to increase capacity and throughput. All of the concentrators are software-upgradeable and have a Motorola PowerPC processor, NVRAM (this is where critical system parameters are stored, such as management passwords), and Flash memory for files. The following sections will cover the different concentrator models.

Note

Cisco doesn't charge their customers for using the Cisco VPN client software; instead, limits are placed on the Easy VPN server side. In other words, the Easy VPN server product you buy will affect how many simultaneous clients (or users) you can terminate on it.

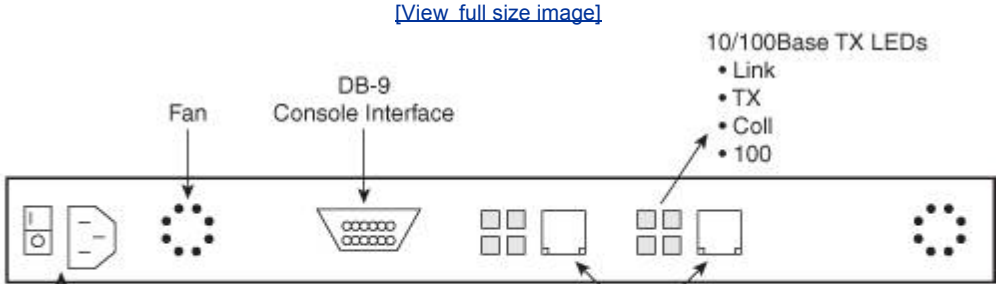
3005 Concentrator

The 3005 is for small businesses with a small-bandwidth Internet connection. The 3005 supports up to 4 Mbps VPN performance, so it's ideal for sites that have a T1, cable modem, or DSL connection. You can have up to 200 IPsec remote access sessions terminated on the 3005 or 50 WebVPN sessions with Version 4.7 of the operating system installed.

The 3005 is not hardware-upgradeable, but you can upgrade the software. The 3005 does only software-based encryption and supports a single power supply. Figure 6-1 shows the rear of the 3005 chassis. It has two autosensing 10/100BaseTX Ethernet interfaces. The left-hand interface is a private interface, connected to the internal network, and the right-hand interface is the public interface, connected to the external network. The only item of interest on the front of the 3005 chassis is a system LED; hence, the front of the chassis is not shown in the diagram.

Figure 6-1. 3005 Chassis

[View full size image]



Concentrator Modules

The 3015, 3020, 3030, 3060, and 3080 support modular slots for additional cards. Currently, the only two cards that you can put into these slots are SEP-2 and SEP-E modules. SEP modules perform VPN functions, such as encryption, in hardware.

When Cisco acquired Altiga, there were three cards you could put in these slots: an SEP module (Version 1), a T1 module, or an E1 module. Cisco no longer sells these cards: only the SEP-2 and SEP-E modules are available.

SEP Modules

The SEP-2 modules will perform encryption for DES and 3DES only. The SEP-E module has replaced the SEP-2 module. It allows the concentrator to perform DES, 3DES, *and* AES encryption. To perform AES encryption in hardware, the concentrator also needs to be running at least Version 4.0 of the software.

You cannot use both SEP-2 and SEP-E modules in the same chassis. If you have a concentrator that has both, the SEP-2 modules are disabled automatically and only the SEP-E module(s) will be active.

To determine the kind of SEP module you have installed, you can either log in to the concentrator to see the type of module (the **Monitor > System Status** screen) or you can examine the module itself. In the lower right corner of the SEP card's cover plate will be a label with one of these pieces of information:

- SEP 200U indicates an SEP-2 module
- SEP-E indicates an SEP-E module

Caution

The SEP modules are *not* hot-swappable; failing to turn off and unplug the concentrator when inserting or removing an SEP module can destroy the box and possibly cause electrocution.

SEP Operation

Each SEP module supports between 1,500 and 5,000 simultaneous remote access sessions, depending on the 3000 series model the module is plugged into. Placement of the SEP modules in the chassis of the concentrator is important. Referring back to Figure 6-2, the top two slots, by default, are the active slots. They process VPN sessions. The slot beneath a top slot provides redundancy for the slot above it. Redundancy is top-down, as follows:

- If a top SEP module fails and there is an SEP module installed beneath it, no VPN sessions are lost because the bottom module has a replication of all VPN information of the module above it.
- If you have only two SEP modules in the chassis and they are installed in the top two slots, sessions will be split between the two modules. If one of the modules fails all VPN sessions are dropped. Site-to-site sessions will be rebuilt to the other SEP module automatically; however, remote access users will have to reinitiate their VPN session manually (unless their client supports the auto-initiation feature).

Concentrator Features

Cisco 3000 series concentrators support features that provide high performance, scalability, enhanced security, high availability, and many other benefits. Here's a brief list of features:

- High performance is provided by SEP modules for hardware-based encryption.
- Scalability is provided by the Cisco Virtual Cluster Agent (VCA) load balancing technology and a modular design with four SEP slots.
- Enhanced security is provided by internal and external (AAA RADIUS, Microsoft's NT Domain and Active Directory, and RSA's SDI) user authentication, firewall policies, user and group management features, and detailed logging.
- High availability is provided by redundant SEPs, redundant chassis failover with VCA or VRRP, and SNMP management and monitoring.
- VPN implementations include WebVPN, IPsec, PPTP, L2TP, L2TP over IPsec, and these features: NAT-T, IPsec over UDP, and IPsec over TCP.
- VPN remote access policy features include (by group or user) filtering, idle and maximum session timeouts, time of day access control, authorization profiles, firewall policies, split tunneling, client and network extension modes, addressing pools, and different authentication methods per group.
- VPN technologies supported include ESP; GRE (for PPTP only); DES, 3DES, and AES; MD5 and SHA-1; MPPE with 40- or 128-bit RC4; ISAKMP and IKE; DH groups 1, 2, 5, and 7; SCEP; and X.509 digital certificates.
- Routing protocols supported include RIP v1 and v2, OSPF, RRI, static routing, and network auto discovery (NAD).
- The concentrators are compatible with the following remote access clients: WebVPN via a web browser or the Cisco SSL VPN Client; Cisco VPN Client for IPsec on Windows 98, ME, NT 4.0, 2000, XP, Linux for Intel, Solaris for UltraSparc, and MAC OS X 10.x; Microsoft's PPTP/MPPE/MPPC client with MS-CHAP or EAP; Microsoft's L2TP over IPsec for Windows 98, ME, NT 4.0, 2000, and XP.
- Management features include access via the console port, Telnet, SSHv1, HTTP, and HTTPS; authentication, authorization, and accounting of administrators through AAA TACACS+; access control of management sessions, logging via the console, a logging buffer, syslog, SNMP, and e-mail; automatic backup of logs via FTP; address translation with NAT and PAT; and packet filtering.

The following sections will discuss some important features that were introduced in newer versions of the concentrator's software. The Cisco 3.0 code release is the first major update of the software since Cisco acquired Altiga. Because this was a handful of years ago, I'll start with 3.5 and work my way up with the new features.

Why I prefer Cisco concentrators

I have worked with many VPN gateway products in my time, including all of the Cisco productsrouters, PIXs and ASAs, and concentrators. And out of all the VPN gateway products I've ever dealt with, Cisco VPN 3000 series concentrators are my favorite for remote access solutions. From a feature perspective, there's nothing else in the marketplace in a single chassis that offers all of the features that Cisco has bundled with their concentrators. When it comes to configuration, the GUI interface is very intuitive and *much* easier to use than a CLI like Cisco routers and PIXs (the concentrator does support a menu-based CLI). And when it comes to troubleshooting VPN connections, the logging functions of the concentrator far surpass anything else I've used. The logging output is customizable to 13 different levels and the output is written, for the most part, in layman's terms, making troubleshooting a simple process.

Introduction to Accessing a Concentrator

Now that I've covered the different 3000 models and provided a brief overview of the features of the concentrators, I'll give a quick overview on how to access a concentrator and navigate through its screens. The remaining chapters in Parts II and III will cover the specifics of how to set up remote access (IPsec, PPTP, L2TP/IPsec, and WebVPN) and site-to-site sessions that terminate on the VPN 3000 concentrators.

You can access the concentrator out-of-band by using its console port. The console port is a DB-9 interface. When using a terminal package, such as HyperTerminal or TeraTerm, set its communications properties to the following:

- 9600 baud
- 8 data bits
- 1 stop bit
- No parity
- Hardware flow control or no flow control

In-band management is supported with the following protocols:

- Telnet
- SSH
- HTTP or HTTPS
- FTP
- TFTP
- SCP

The concentrators support two types of management interfaces: character-based interface, commonly called a command-line interface (CLI), and graphical user interface (GUI). The following sections will discuss these two interfaces.

Command-Line Interface

When you receive a concentrator from the factory, it has no configuration on it; therefore, you need to use the menu-driven CLI to put a basic configuration on it. Once you have, at the minimum, an IP address on the private interface of the concentrator, administrators typically use a web browser from there on out to manage it.

To help you understand the initial access to the CLI, and additional tasks you can perform from the CLI, the following sections cover these topics:

- Bootup Process
- Initial Configuration
- CLI Menu Access
- Password Recovery

Bootup Process

To see the bootup process of the VPN 3000 concentrator, you need to use its console, as shown in Example 6-1.

Example 6-1. Concentrator Bootup Process

```
Boot-ROM Initializing...
Boot configured 32Mb of RAM.
```

Summary

This chapter introduced you to Cisco VPN 3000 series concentrators. I discussed and compared the various models that Cisco sells, and introduced you to the two configuration interfaces the concentrator offers: CLI and GUI. Now that you have a basic understanding of how to interact with the concentrator, the next set of chapters in this part will focus on configuring remote access and site-to-site sessions, managing the concentrator, and troubleshooting connections.

Next up is Chapter 7, "Concentrator Remote Access Connections with IPsec," where I show you how to set up a concentrator to accept remote access connections that use IPsec. Remote acccess solutions using PPTP, L2TP, and WebVPN will be discussed in Chapter 8, "Concentrator Remote Access Connections with PPTP, L2TP, and WebVPN."

PREV NEXT

Chapter 7. Concentrator Remote Access Connections with IPsec

The last chapter introduced the VPN 3000 concentrators, including a basic overview of their CLI and GUI interfaces. This chapter will focus solely on using a concentrator to terminate remote access sessions. Cisco concentrators support these remote access VPN implementations:

- IPsec
- PPTP
- L2TP over IPsec
- WebVPN

The remainder of this chapter will focus on the configuration of the concentrator to support IPsec remote access VPN implementations, including basic monitoring of connections and controlling remote access connectivity to the concentrator through groups, users, and network access control (NAC). Chapter 8 will discuss the latter VPN implementations on the VPN 3000 concentrators. I'll discuss detailed troubleshooting of VPN connections in Chapter 11, "Verifying and Troubleshooting Concentrator Connections," and the setup of the remote access clients in Part III, "Clients."

Controlling Remote Access Sessions to the Concentrator

Before I begin discussing how to terminate remote access sessions on a concentrator, I first need to discuss two basic means for controlling remote access to the concentrator:

- Groups
- Users

The following two sections will discuss these two concepts.

Group Configuration

Groups are used to simplify the application of remote access policies to your remote access users. If users have similar policies, you can create a single group with the associated policies and then place the right users into that group. With this process, you only have to create the policies once, but you can apply the policies to many users.

Cisco supports two types of groups:

- Base or Global
- Specific

The following two sections will discuss these two group types.

Base Group

The base group, commonly called the global group, is used for one primary purpose: to define remote access policies that are common to all specific groups and users. In other words, users typically are not associated with the global group. As an example, perhaps you have a policy that says all remote access users can use IPsec only to establish a tunnel to your concentrator. You have 20 specific groups this policy needs to be applied to. This policy is defined at the group level. However, it makes no sense to have to apply this policy to each of the groups individually. This is the main purpose of the base group: to apply a single policy that will affect all groups on the concentrator.

To access the base group's configuration, go to **Configuration > User Management > Base Group**. From here, you'll see the screen shown in Figure 7-1. At the top of the screen are tabs that take you into different configuration areas for the group policies. In Figure 7-1, the default tab, General, is in the foreground. You can click a tab to take you to different areas. I'll discuss each of these tabs, and their configuration parameters, throughout the rest of this chapter and the next. The one exception to this is the HW Client tab, which I'll discuss in Chapter 14, "3002 Hardware Client."

Figure 7-1. Global Group Screen

[View full size image]

IPsec Remote Access

Now that you have a basic understanding of some of the global tasks you'll perform no matter what type of remote access sessions will be terminating on your concentrator, the remainder of this chapter will focus on the setup of IPsec remote access sessions. If you recall from Chapter 3, "IPsec," IPsec is one standard that defines how to protect traffic between devices. It supports site-to-site and remote access connectivity. For remote access sessions, the following steps are performed to build a tunnel:

1. Negotiate the ISAKMP/IKE Phase 1 transform set.

2. Perform Diffie-Hellman (DH) to come up with the shared secret key, and use this to encrypt the encryption key and HMAC function key to share between the peers.

3. Perform device authenticationconcentrators only support pre-shared keys and digital certificates with a CA.

4. Authenticate the user using XAUTH.

5. Apply access policies to the user, such as assigning the user an internal IP address, defining split tunneling and split DNS usage, enforcing a firewall policy, and so on. Cisco commonly calls this step either IKE Mode Config or IKE Client Config.

6. You can use reverse route injection (RRI) to advertise the client's internal addressing information; this is necessary if the following are true:

   - The client could connect to more than one concentrator at the corporate site and local address pools are not being used to assign addressing information to the client.

   - The client is a hardware client in network extension mode and needs to advertise the address of its private or inside interface to the corporate office.

One of the issues of IPsec is that for remote access connections, how policies are applied to a client are not defined by the IPsec standardeach IPsec vendor does this differently. Therefore, when I explain how to configure Step 5 later in this chapter, it is specific to the Cisco implementation. Also, I'll cover Step 6 in Chapter 14, "3002 Hardware Client," Chapter 18, "Router Remote Access Connections," and Chapter 22, "PIX and ASA Remote Access Connections." This chapter will focus on the configuration of remote access software clients on the concentrator.

This part of the chapter will discuss:

- Configuring ISAKMP/IKE Phase 1 transform sets (proposals) for the management connection of IPsec clients.
- Defining device authentication.
- Configuring the group's general IPsec policies.
- Configuring the group's IKE Mode Config policies.
- Setting up IPsec client firewall policies.
- Configuring ISAKMP/IKE Phase 2 data transform sets.

ISAKMP/IKE Phase 1: IKE Proposals

The first configuration items I'll explain are how to configure Steps 1 and 2: negotiating the Phase 1 transforms and the use of DH to build the secure management connection. On the concentrator, this is referred to as setting up your "IKE Proposals or Policies." The following two sections will discuss how to set up your IKE proposals.

IKE Proposal Screen

Network Access Control (NAC) for IPsec and L2TP/IPsec Users

In version 4.7, Cisco introduced the Network Access Control (NAC) feature for IPsec and L2TP/IPsec clients Secure Desktop (CSD) feature for WebVPN (discussed in the next chapter), NAC for IPsec and L2TP/IPsec validating a user's access based on their operating system version and applied service packs, the anti-virus updates, the personal firewall software and applied updates, and the intrusion protection software and app

With CSD for WebVPN, the concentrator validates a user's access. With NAC, the concentrator serves as a Trust Agent (CTA) software is installed on a user's PC and sends the required NAC information, using the Ex Authentication Protocol (EAP), to the concentrator. The concentrator then forwards this information to an Secure ACS (CSACS), using EAP over RADIUS. The AAA server validates the user's access and sends the r concentrator, along with any other policy access information, where the concentrator enforces the downlo for WebVPN, you have to define all the policies locally on the concentrator, whereas with NAC, these polici AAA RADIUS server; because of this, setting up NAC on the concentrator is simpler.

Note

The configuration of NAC on CSACS is beyond the scope of this book. Information on configuration NAC on at http://www.cisco.com/en/US/partner/products/sw/secursw/ps2086/products_user_guide_chapter09186

Global Configuration of NAC for IPsec

To set up the global configuration of NAC, go to **Configuration > Policy Management > Network Admis** are two options from this screen: *Global Parameters* and *Exception List*. The next two sections will talk abo options.

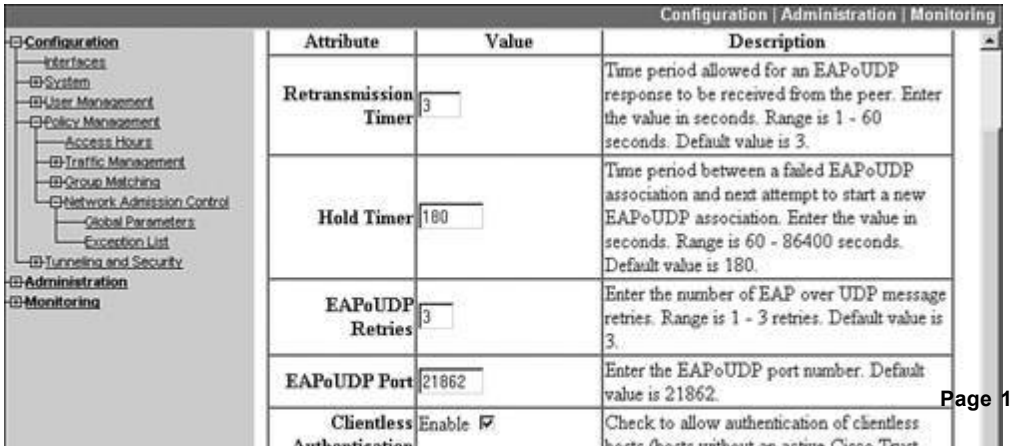
NAC Global Parameters

When you click the **Global Parameters** hyperlink option from the **Configuration > Policy Management > Control** screen, you are taken to the screen shown in Figure 7-52. Here are the parameters you can config

- **Retransmission Timer** Defines how long the concentrator will wait for a NAC response from a devic request. The default is 3 seconds but can range from 160.
- **Hold Timer** Defines how long the concentrator waits before attempting to establish a new associati failed NAC credential validation or the configured number of EAP over UDP (EAPoUDP) retries has be default is 180 seconds but can range from 601,440.
- **EAPoUDP Retries** Defines the number of times the concentrator will retransmit EAP over UDP messa NAC association as failed, thereby starting the hold timer. The default is 3 times, but this can range
- **EAPoUDP Port** Defines the EAP over UDP port used for NAC communications; this defaults to 21,862
- **Clientless Authentication: Enable** Allows authentication of user devices that aren't using Cisco Tr because you're in a migration process and slowly adding CTA to your user's desktops). Once you've to define a username and password for the *Clientless Authentication: Username* and *Clientless Auth* parameters. The AAA RADIUS server will use these authentication credentials to validate network a

Figure 7-52. NAC Global Parameters Screen

[View full size image]

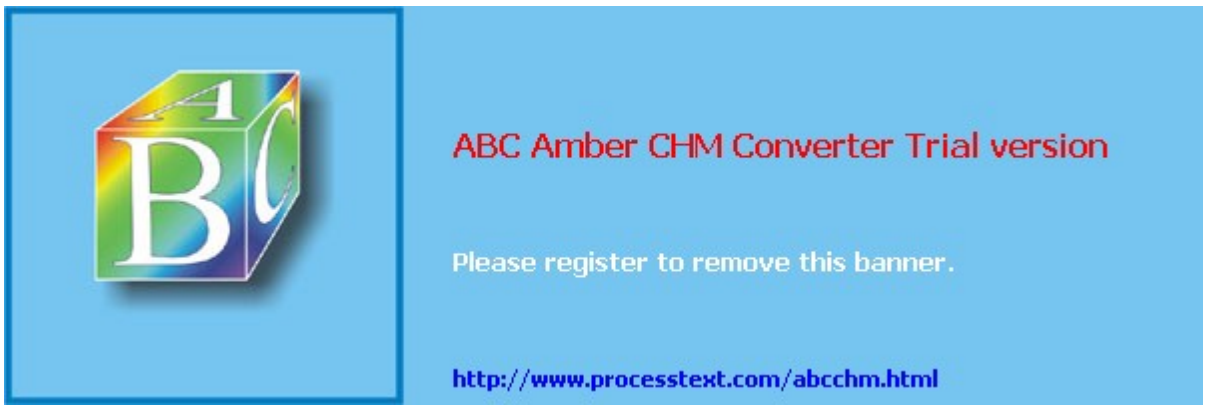


Summary

This chapter introduced you to configuring remote access sessions using IPsec. In regard to IPsec, I'll spend more time on how to implement and troubleshoot it on the concentrator when I discuss troubleshooting in Chapter 11, "Verifying and Troubleshooting Concentrator Connections," and the configuration of remote access clients in Part III.

Next up is Chapter 8, "Concentrator Remote Access Connections with PPTP, L2TP, and WebVPN," where I show you how to use your concentrator to terminate remote access sessions from PPTP, L2TP, L2TP/IPsec, and WebVPN clients.

PREV NEXT



Chapter 8. Concentrator Remote Access Connections with PPTP, L2TP, and WebVPN

The last chapter introduced how to configure the VPN 3000 concentrators to accept IPsec remote access sessions. This chapter will focus on using a concentrator to terminate other types of remote access sessions, including:

- PPTP
- L2TP over IPsec
- WebVPN

The remainder of this chapter will focus on the configuration of the concentrator to support these types of remote access VPN implementations. I'll spend more time on implementing PPTP/L2TP connectivity in Chapter 13, "Windows Software Client." Most of this chapter is dedicated to the newest CiscoVPN implementation: SSL. Cisco refers to their SSL VPN implementation as WebVPN. I'll focus on setting up the concentrator to accept clientless connections (just a web browser), thin client connections (port forwarding), and network client connections (SSL VPN Client software).

PREV NEXT

PPTP and L2TP Remote Access

The configuration of PPTP and L2TP remote access is much simpler than that of IPsec, even if you'll be using L2TP over IPsec; this is because if you're using IPsec with Cisco remote access clients, you have many more IKE Mode Config features than what L2TP over IPsec clients support. The configuration of both of these protocols occurs in two locations on the VPN 3000 concentrator:

- User management
- Global configuration

Most of your configuration is done at the group level; very rarely will you have to change the global (system-wide) properties for PPTP or L2TP. The following three sections will discuss the configuration of these two protocols: group configurations for PPTP and L2TP, global configurations for PPTP, and global configurations for L2TP.

PPTP and L2TP Group Configuration

Most of your configuration for PPTP and L2TP will be done under a group's configuration: **Configuration > User Management > Groups**. In a group within the General tab, (discussed in the last chapter), the *Tunneling Protocols* parameter allows you to specify which tunneling protocols, if any, a group is allowed to use. If you want the group to be able to use PPTP or L2TP, you must select these.

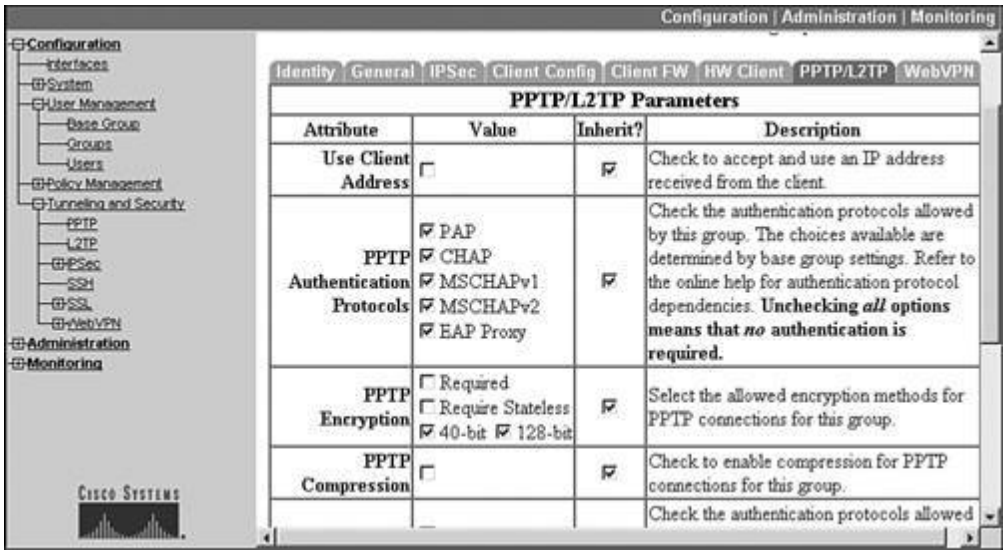
Note

If you want to use L2TP over IPsec, the group no longer will be able to perform general IPsec connections to the concentrator.

The remainder of the group's configuration is done under the PPTP/L2TP tab; the top part of this screen is shown in Figure 8-1. The first parameter is the *Use Client Address* parameter, which is disabled. If you want the client to use its own address, make sure you've enabled this option in the **Configuration > System > Address Management > Address** section; this was discussed in the "Address Assignment" section earlier in the last chapter. However, because of various reasons, including security, it is recommended to have the concentrator assign an address to the client.

Figure 8-1. *Group Configuration:* PPTP/L2TP Tab

[View full size image]



Below this there are two sets of parameters: one set applies to PPTP, which you can see in Figure 8-1, and one set applies to L2TP (you can't see this in Figure 8-1). Here are the parameters:

- **PPTP Authentication Protocols** These check boxes specify the PPP authentication

WebVPN Remote Access

The third type of remote access connectivity I will discuss is how to set up the concentrator to accept WebVPN sessions. I discussed SSL VPNs in Chapter 5, along with the Cisco implementation of SSL VPNs: WebVPN.

SSL uses digital certificates for authentication (in most cases this is server-side authentication). By default, the concentrator will create a self-signed SSL server certificate when it boots up. Or, you can install a third-party SSL identity certificate on the concentrator; however, if you choose this option, you also must install certificates on your web browser clients.

If you choose the latter option, you'll need to obtain a certificate for your concentrator, which I explained previously in Chapter 7 in the "Digital Certificates" section.

The remainder of this section will focus on the five areas concerning the setup of WebVPN on the VPN 3000 concentrators:

- HTTPS access
- System-wide WebVPN parameters
- Group WebVPN configuration
- SSL VPN Client (SVC)
- Cisco Secure Desktop (CSD) for WebVPN access

HTTPS Access

One of the first steps you'll need to perform is to make sure that HTTPS access and WebVPN access are allowed to the concentrator. The following two sections will discuss how you allow HTTPS and WebVPN access.

First, you need to make sure that HTTPS access is allowed by your concentrator. There are two areas that you'll need to examine and possibly change:

- HTTPS properties
- WebVPN interface configuration

HTTPS Properties

To access the concentrator's HTTPS properties, go to **Configuration > Tunneling and Security > SSL**. There are two options on this page:

- *HTTPS*
- *Protocols*

If you click the **HTTPS** hyperlink, you'll find three options:

- **Enable HTTPS** This check box enables or disables HTTPS access; by default it is enabled.
- **HTTPS Port** This text box allows you to change the TCP port number that HTTPS access will use; this defaults to 443, but you can change it to another number to enhance your security. Please note that IPsec over TCP cannot use a port that WebVPN will be using.
- **Client Authentication** This check box, when checked, allows the concentrator to verify the client's digital certificate. If you choose this option, you must install certificates manually on all of your client's web browsers as well as on your concentrator. You also must configure an authorization server using RADIUS or LDAP. I discussed this in the last chapter in the "Authorization Servers" section. Next, in the group configuration's IPsec tab, set *Authentication* to "None" and the *Authorization Type* to either "RADIUS" or "LDAP." Remember that the concentrator will send the

Summary

This chapter introduced you to configuring remote access sessions: PPTP L2TP/IPsec, and WebVPN. In regard to PPTP and L2TP/IPsec, I'll spend more time on how these are configured on the concentrator when I discuss the Microsoft software client in Chapter 13, "Windows Software Client."

Next up is Chapter 9, "Concentrator Site-to-Site Connections," where I show you how to use your VPN 3000 concentrator to terminate site-to-site, or LAN-to-LAN (L2L), sessions.

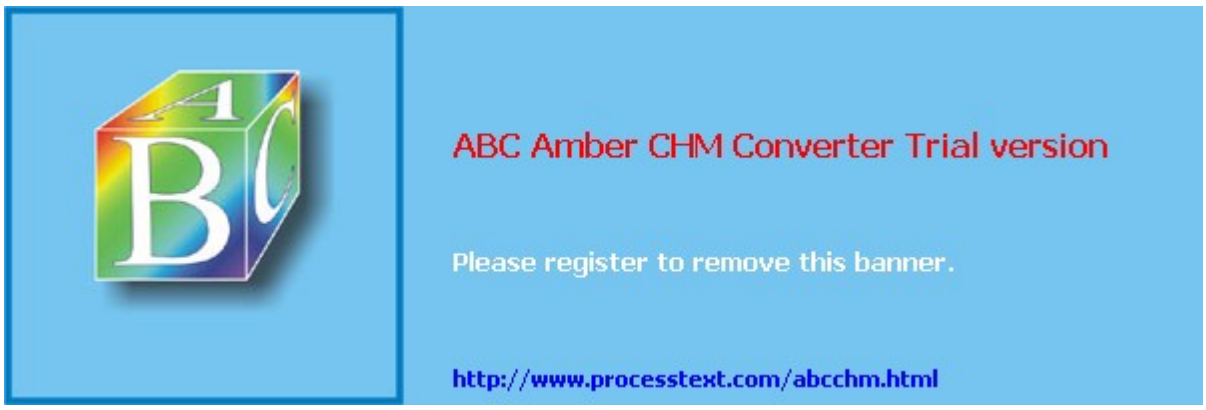
PREV NEXT

Chapter 9. Concentrator Site-to-Site Connections

Chapters 7 and 8 focused on using the concentrator to terminate remote access sessions: IPsec, PPTP, L2TP/IPsec, and WebVPN. Most people assume that the concentrator's primary purpose is for remote access connectivity; however, you can also use the concentrator to set up site-to-site connections. With Cisco concentrators, site-to-site sessions commonly are referred to as LAN-to-LAN (L2L) connections. Because Cisco concentrators are fully compliant with IPsec, it is easy to use a concentrator to terminate L2L sessions with other VPN gateway devices such as Cisco routers, PIX and ASA security appliances, other VPN 3000 concentrators, and other vendors' VPN gateway products.

This chapter will focus on using a VPN 3000 concentrator to terminate L2L sessions by covering the following items:

- L2L Connectivity Example
- ISAKMP/IKE Phase 1 Preparation
- Adding Site-to-Site Connections
- Address Translation and L2L Connections

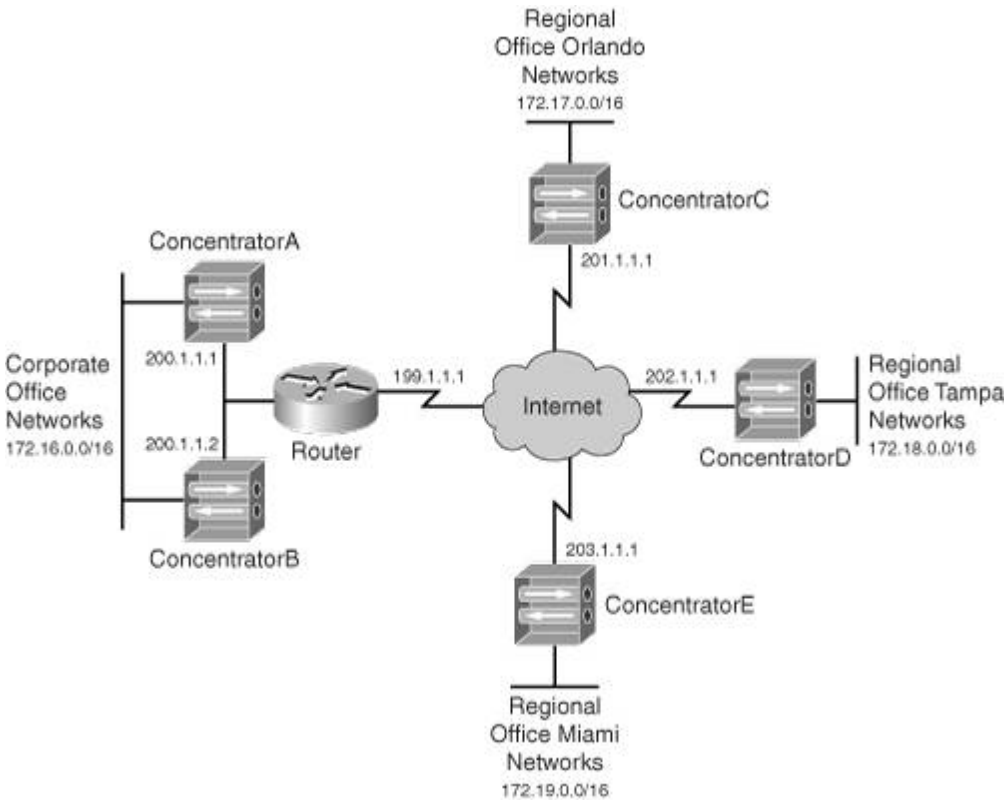


L2L Connectivity Example

To understand the components involved in an L2L session, I've created the diagram shown in Figure 9-1. This figure shows a simple example of a network using L2L sessions. In this example, a corporation has two redundant 3060 concentrators at the corporate site: ConcentratorA and ConcentratorB. These concentrators handle L2L sessions and many remote access sessions. Redundancy is set up between the concentrators. This chapter discusses L2L redundancy and in Chapter 10, "Concentrator Management," I'll discuss remote access redundancy.

Figure 9-1. L2L Example

[View full size image]



The corporate network is using 172.16.0.0/16 for a network number, where this has been subnetted into many subnets. The regional offices in Orlando, Tampa, and Miami each have a 3030 concentrator. These concentrators each have an L2L session back to the redundant configuration at the corporate office. These concentrators also handle local remote access users. Because very little traffic flows between the regional offices, the network administrators decided to send all traffic through the corporate site; however, if traffic patterns change, an L2L session can easily be added between two regional sites.

All of the VPN 3000 concentrators support IPsec L2L sessions; however, not every concentrator has the same capabilities. Table 9-1 compares the number of simultaneous L2L sessions that each of the concentrators support. Remember from Chapter 6, "Concentrator Product Information," that L2L sessions count as a session against the total number of concurrent (L2L and remote access) sessions that a concentrator supports. For example, the 3080 supports 10,000 total sessions, of which no more than 1,000 of those can be L2L sessions.

Table 9-1. Concentrator L2L Session Restrictions

Models	Maximum L2L Sessions
3005	100
3015	100

ISAKMP/IKE Phase 1 Preparation

The remainder of this chapter will discuss how to set up and modify L2L connections, and will examine the kinds of issues you'll deal with when using these connections. Before you begin adding an L2L session, you'll first need to create an ISAKMP/IKE Phase 1 transform set that you'll use for the L2L session. This section will discuss the ISAKMP/IKE Phase 1 transforms that you can use or create for your L2L connection.

Existing IKE Policies

Cisco already has some predefined Phase 1 transforms that you can use for your L2L sessions. If you recall from Chapter 7, "Concentrator Remote Access Connections with IPsec," to access the concentrator's existing Phase 1 transforms, you go to the **Configuration > Tunneling and Security > IPsec > IKE Proposals** screen.

Table 9-2 lists the L2L Phase 1 transforms that exist and are activated, by default, on the concentrators. Of course, there are other predefined transforms that are not activated by default. You can use the ones Cisco has predefined, modify these, or create your own.

Table 9-2. Concentrator Predefined Active ISAKMP/IKE Phase 1 Transforms

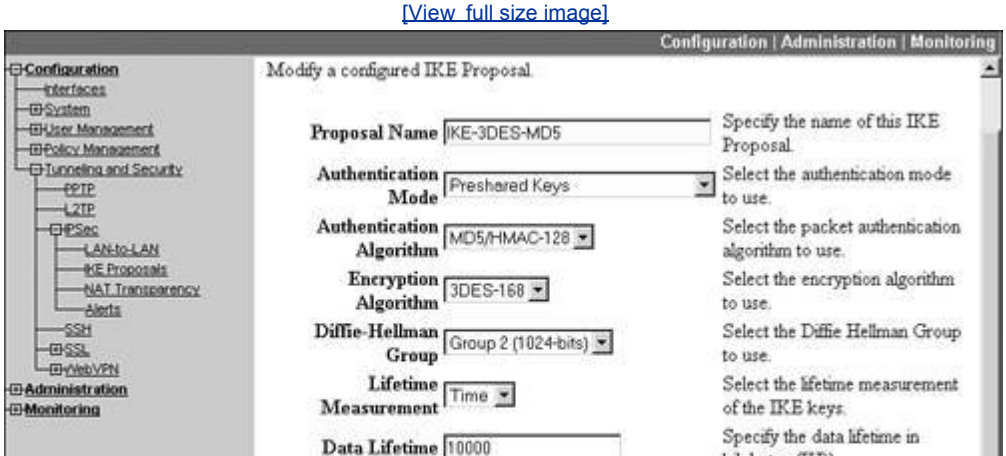
Proposal Name	Encryption Algorithm	HMAC Function	DH Key Group	Device Authentication
IKE-3DES-MD5	3DES	MD5	2	Pre-shared keys
IKE-3DES-MD5-DH1	3DES	MD5	1	Pre-shared keys
IKE-DES-MD5	DES	MD5	1	Pre-shared keys
IKE-3DES-MD5-RSA	3DES	MD5	2	RSA signatures
IKE-AES128-SHA	AES-128	SHA	2	Pre-shared keys

IKE Policy Screen

From the IKE policy screen, click the *Add* button to add a new proposal or select an existing proposal by clicking its name and click the *Modify* button to change it. This takes you to the IKE policies configuration screen shown in Figure 9-2.

Figure 9-2. IKE Policies Screen

[View full size image]



Adding Site-to-Site Connections

Once you have set up your IKE policies for your management connection, you are ready to create your L2L session. To add (or modify) an L2L session, go to **Configuration > Tunneling and Security > IPsec > LAN-to-LAN**. The resulting screen lists the existing L2L connections.

Note

Chapter 6, "Concentrator Product Information," discussed the use of Quick Configuration to put a minimal configuration on your concentrator. This method only allows you to set up remote access connectivity. Before you can add an L2L session, you must configure the concentrator's public interface by, at minimum, assigning an IP address to it. This can be done using Quick Configuration or going to **Configuration > Interfaces**.

To assist you with building and maintaining L2L sessions, the following subsections cover these topics:

- Adding L2L Sessions
- Completing L2L Sessions
- Modifying L2L Sessions

Adding L2L Sessions

The following subsections cover these topics:

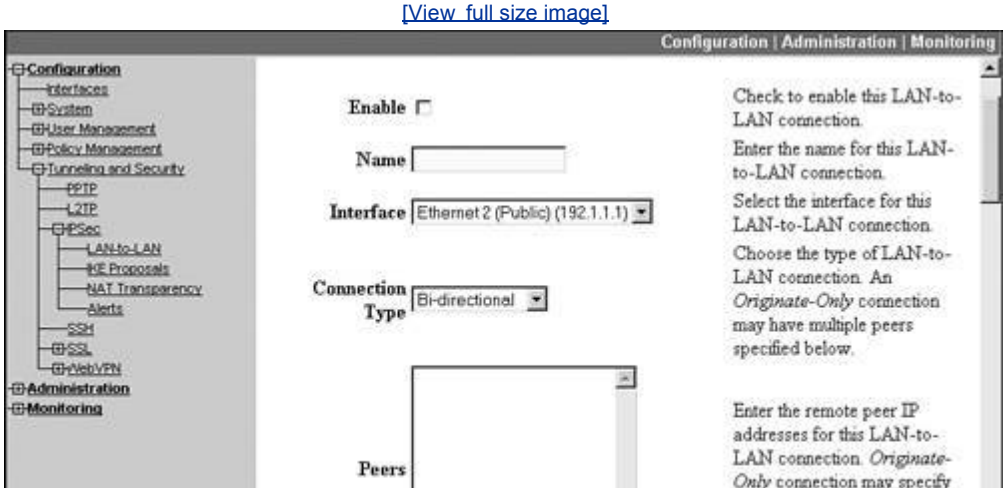
- Basic L2L Configuration Parameters
- Peer Connectivity
- Device Authentication Information
- Connection Policies
- Routing Options
- Local and Remote Networks

Basic L2L Configuration Parameters

To add a new session, click the *Add* button, where you'll be taken to the screen shown in Figures 9-3, 9-4, 9-5, and 9-6 (I've broken this screen into multiple figures). The first parameter at the top of the screen in Figure 9-3 is the *Enable* parameterclicking this check box enables this specific L2L session. This option allows you to disable a session without having to delete the session, which might be useful when you are trying to troubleshoot a problem.

Figure 9-3. L2L Screen: Part 1

[View full size image]



Address Translation and L2L Sessions

Because IPsec L2L sessions create a logical extension between two networks, the assumption is that the two locations have unique network numbers. Of course, if you are connecting two of your own sites together, you should have designed your network to have unique numbers. However, if you are connecting to a different company, such as a business partner, you might have overlapping addresses between the two companies or networks. In this situation, you would have to implement address translation to solve the problem.

The VPN 3000 concentrators support basic address translation abilities that you can apply to an interface and affect all traffic entering or leaving it or for L2L sessions. In this book I'll discuss only using address translation for L2L sessions because there are better solutions, such as the PIX/ASA or IOS routers, that are more flexible in creating address translation policies for traffic entering and leaving interfaces.

To help you understand address translation and the concentrator's configuration of address translation, the following sections will discuss these topics:

- Introducing concentrator address translation abilities
- Example needing L2L address translation
- Creating L2L address translation rules
- Enabling L2L address translation

Introducing Concentrator Address Translation Abilities

The address translation feature for L2L connections is commonly called LAN-to-LAN NAT (network address translation) by Cisco. L2L NAT is only necessary when you have overlapping addresses at both sides of an L2L tunnel, or you want to control what networks appear as on the two sides of an L2L tunnel. Before I begin discussing how to set up L2L NAT, I first need to discuss the address translation capabilities of the concentrators.

There are three types of address translation supported by the concentrator:

- **Static NAT** This can be used to translate local source addresses statically before sending packets across the tunnel or remote destination addresses as packets come off of the tunnel; static NAT is most commonly used for services that must always be presented by the same address.
- **Dynamic NAT** This can be used to translate local source addresses dynamically to unique global addresses (using a pool of addresses) before sending packets across the tunnel; dynamic NAT is typically used when dynamic PAT won't work (dynamic PAT only works with TCP and UDP connections). You cannot use dynamic NAT for packets coming from a remote L2L peer across the IPsec tunnel; the remote L2L peer has to perform this process.
- **Dynamic PAT** This can be used to translate local source addresses dynamically to the *same* global address, differentiating each local device by ensuring that the source port number in the TCP or UDP segment is unique for each connection. You cannot use dynamic PAT for packets coming from a remote L2L peer across the IPsec tunnel; the remote L2L peer has to perform this process.

Note

Even though Cisco calls the address translation feature L2L NAT, this is misleading because the concentrator supports both static and dynamic NAT *and* PAT.

Example Needing L2L Address Translation

I'll now take a look at a simple example that needs address translation for an L2L connection, shown in Figure 9-13. In this example, both sites are using 10.0.0.0/8 (shown at the top of the figure). SiteA needs to access Web2 and SiteB needs to access Web1, where both

Summary

This chapter showed you the basics of setting up L2L sessions on a VPN 3000 concentrator, including routing and discovery issues and address translation problems for L2L sessions. Next up is Chapter 10, "Concentrator Management," where I show you how to perform some additional configuration tasks such as routing, remote access redundancy, and bandwidth management, and some management features on the concentrator.

PREV NEXT

Chapter 10. Concentrator Management

This chapter is unique when compared to the last three chapters on the VPN 3000 concentrator; those chapters focused on remote access and site-to-site connections. This chapter, instead, will focus on some of the additional configurations and features you might enable on your concentrator. Some important items I'll address in this chapter include bandwidth management, routing, redundancy, and administrative components, including administrative access, software upgrades, and file management.

I won't be discussing all of the screens on the concentrator in this book, because that would be a book in itself. However, what I've covered up to this point focuses on the important aspects of the concentrator. Chapter 11 will focus on using tools on the concentrator to troubleshoot VPN problems.

Bandwidth Management

The first management feature I'll focus on in this chapter is the concentrator's bandwidth management feature. Bandwidth management can be used to define bandwidth policies that affect site-to-site or remote access sessions, such as IPsec, L2TP, and PPTP connections.

Bandwidth management was first introduced in Version 3.6. One concern you might have with VPN sessions is that one group of users or a particular L2L session will use up all of your available bandwidth, leaving no bandwidth for other VPN sessions. Fortunately, bandwidth management allows you to create policies and apply them to your VPN sessions, so that you can prevent this problem.

Using bandwidth management is a two-step process:

1. Creating your bandwidth management policies
2. Activating your bandwidth management policies

The following two sections will discuss this two-step process.

Creating Bandwidth Policies

Cisco supports two types of bandwidth policies:

- Reservation
- Policing

Bandwidth reservation reserves a minimum amount of bandwidth for an L2L or remote access session. This policy type typically is used to ensure that certain users or L2L sessions can get a fair share of the available bandwidth, especially L2L sessions. In other words, your L2L sessions might need a certain amount of bandwidth and you don't want heavy-bandwidth remote access users, such as those with broadband access, to affect your L2L sessions adversely.

Bandwidth policing places limits on the maximum rate for tunneled traffic; traffic that exceeds the configured rates is dropped by the concentrator. This policy type typically is used when you want to ensure that a particular L2L session or group of users (like those with broadband connections) don't hog all of the bandwidth on a particular interface on which their VPN session terminates.

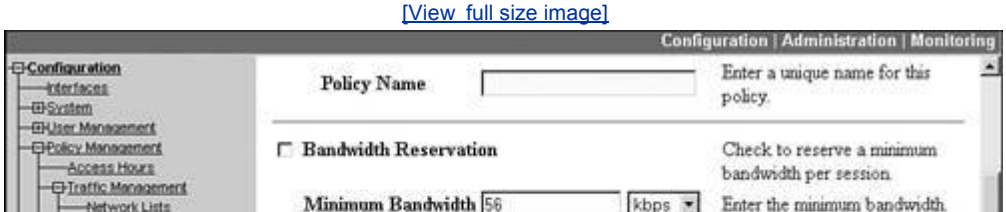
One of your first tasks in setting up bandwidth management is to create your bandwidth policies. Typically, you'll create three types of bandwidth policies:

- A general policy that will affect all VPN traffic entering or leaving your concentrator's public interface
- Specific policies for your remote access groups (these policies override the general interface policy)
- Specific policies for your L2L sessions (these policies override the general interface policy)

To create a bandwidth policy, go to **Configuration > Policy Management > Traffic Management > Bandwidth Polices.** This screen lists your existing policies. To add a policy, click the *Add* button, taking you to the screen in Figure 10-1.

Figure 10-1. Bandwidth Policy Creation

[View full size image]



Routing on the Concentrator

Because concentrators are used in medium-to-large networks and these networks are composed of many subnets, you'll need to set up routing on your concentrator. The concentrator supports two types of routing: static and dynamic. You can create specific static routes or default routes. In addition, the concentrator supports two dynamic routing protocols: RIP and OSPF. The remaining subsections will discuss the configuration of routing on the concentrator.

Static Routing

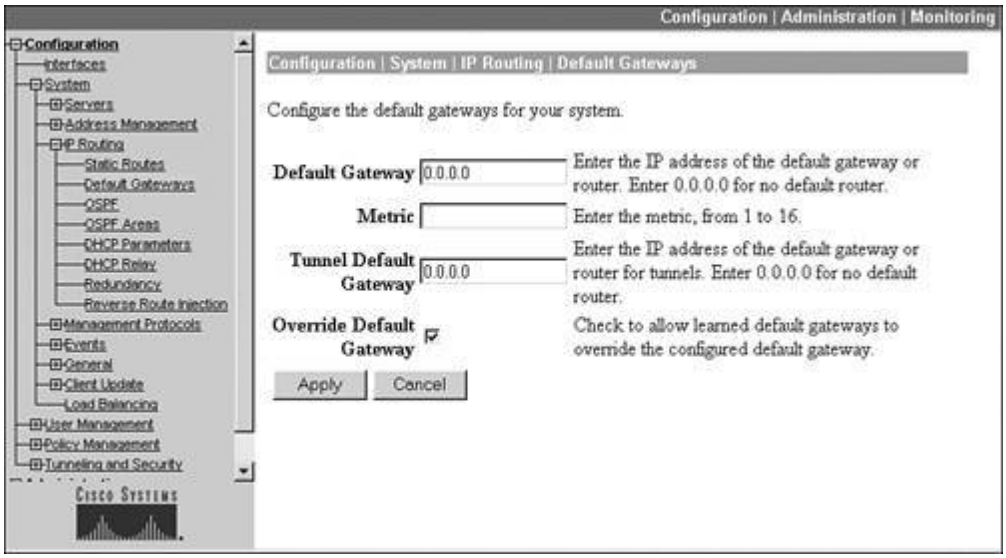
Global routing configurations are performed on the IP Routing screen, which is accessed by clicking **Configuration > System > IP Routing**. The concentrator supports two types of static routes: default and static. Each is discussed in turn in the following sections.

Default Route

A default route is a gateway of last resort: if a specific route is not found in the concentrator's routing table to reach a destination, the default route is used. To define a default route, click the *Default Gateways* hyperlink, which takes you to the screen in Figure 10-4.

Figure 10-4. Default Route

[View full size image]



Within the screen, there are several parameters, as follows:

- The *Default Gateway* parameter allows you to assign an IP address of the next hop to reach the destination network. This is typically a router or firewall.
- The *Metric parameter* allows you to specify a metric for the default gateway. This is used when you need to create a primary and backup default route, where a metric of 1 specifies the primary default route.
- The *Tunnel Default Gateway* parameter allows you to specify a different IP address to use for the default gateway of VPN sessionsnon-VPN traffic would use the default gateway specified in the *Default Gateway* parameter.
- The *Override Default Gateway* check box allows you to use a dynamically learned default gateway instead of the one hard-coded on this screen: this could be one learned via DHCP if the concentrator is a DHCP client or one learned via RIP or OSPF.

You would click the *Apply* button to accept your configuration.

Static Routes

Static routes typically are used when you aren't using a dynamic routing protocol to reach

Chassis Redundancy

The concentrators support two general types of redundancy: intra- and inter-chassis redundancy. Intra-chassis redundancy is available on the 3015 concentrators and higher, where dual power supplies and redundant SEP cards are supported. All of the Cisco concentrators, however, support inter-chassis redundancy through one of two solutions:

- Virtual Router Redundancy Protocol (VRRP)
- Virtual Cluster Agent (VCA), sometimes referred to as Remote Access Load Balancing

Both are mutually exclusive. You can use one *or* the other; both have advantages and disadvantages. For example, VRRP, because it is a standard, can be used to provide redundancy using concentrator and nonconcentrator products (such as Cisco routers); however, VRRP does not support load balancingonly redundancy. VCA, on the other hand, allows for load balancing of incoming remote access sessions; however, VCA is supported only by the VPN 3000 concentrators and the ASA security appliances.

The following two sections will discuss how these inter-chassis redundancy features are implemented on the concentrators.

VRRP

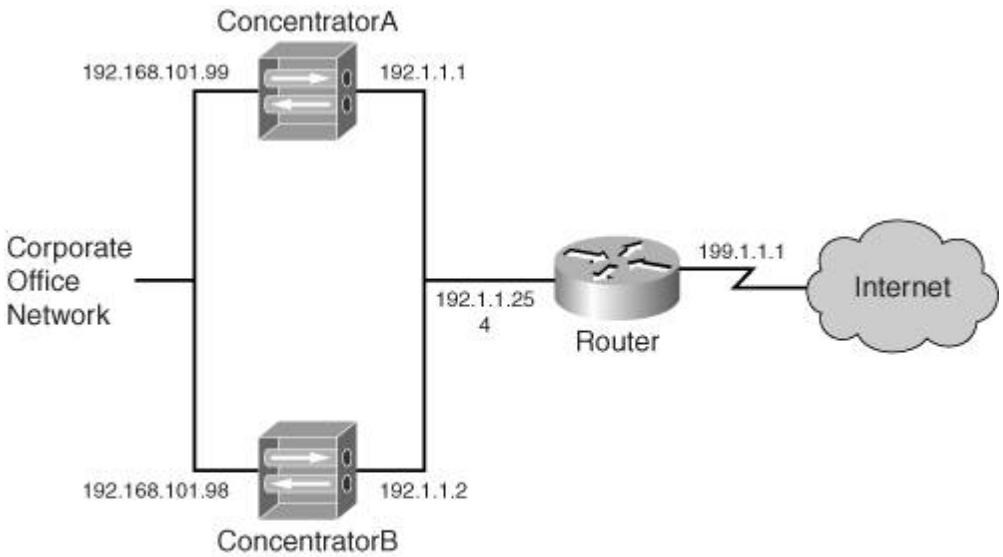
VRRP provides inter-chassis redundancy and is an open standard; even Cisco routers support it. VRRP is typically used to provide default gateway redundancy; however, it can also be used to provide redundancy for VPN connectivity.

With VRRP, an election process is used to elect a master device responsible for handling traffic, including VPN traffic. A virtual address is then assigned by the master, and remote devices connect to this virtual address. The virtual addresses are most commonly the configured IP addresses on the interfaces of the master, but can be a virtual address. The master will process packets sent to the virtual address. One or more backup VRRP peers monitor the status of the master; if the master fails, one of the backups will be promoted to the master role and will begin processing packets sent to the virtual address.

VRRP Example

Figure 10-10 illustrates how VRRP works. In this example, there are two concentrators at the corporate site. One VRRP group is used off of each interface: Group 1. ConcentratorA is the master for each group (you can influence the election process by changing the priority of the master). Note that the virtual addresses used are those of the master concentrator.

Figure 10-10. VRRP Example



Administration Screens

The goal of this book is to focus on VPN sessions; however, because not as many people are familiar with Cisco VPN 3000 concentrators compared to Cisco routers and security appliances, I want to spend some time focusing on some important administrative functions on the concentrator. Therefore, the last topic I'll focus on is the administrative screens (the monitoring screens I'll discuss in Chapter 11, "Verifying and Troubleshooting Concentrator Connections").

All of the administrative functions of the concentrator are found by clicking the *Administration* hyperlink at the top or left-hand side of the web browser window. You have the following options from the main Administration screen:

- **Administer Sessions** Provides statistics for all administrative, remote access, and L2L sessions, including the ability to disconnect sessions.
- **Software Update** Can upgrade the concentrator and remote access client software.
- **System Reboot** Can reboot or power down the concentrator, and schedule these actions.
- **Reboot Status** Can view any scheduled reboots of the concentrator.
- **Ping** Tests connectivity with ICMP echoes.
- **Traceroute** Displays the router path to reach the remote destination.
- **Monitoring Refresh** Defines automatic screen updates for the Monitoring screens.
- **Access Rights** Takes you to a sub-menu screen where you can maintain the administrator accounts on the concentrator, restrict management access to the concentrator, set session timeouts and limits, and specify the use of an AAA TACACS+ server for administrative authentication functions.
- **File Management** Allows you to view, save, swap, delete, and transfer files to the concentrator's Flash memory.
- **Certificate Management** Installs and manages your concentrator's root, identity, and SSL certificates.

The following sections will discuss some of the more important administrative items in more depth.

Administrator Access

This section will focus on managing management access to the concentrator. This can be accomplished with administrative accounts, access control lists (ACLs), access settings, centralized authentication with AAA (authentication, authorization, and accounting), and management protocols and access. The following sections will discuss these control mechanisms briefly.

Administrator Accounts

If you won't be using AAA to centralize management of your concentrators, you'll need to secure the management accounts on each concentrator. To manage your concentrator's accounts, go to the **Administration > Access Rights > Administrators** screen. This screen will list the five management accounts that the concentrators support. Depending on the concentrator software version, from 13 accounts will be enabled: *admin, isp,* and/or *user.* Therefore, you'll want to enable or disable the ones you want to use, and for the ones you want to use, possibly change their level of security access.

Table 10-2 lists the default accounts and their access levels. The *Authentication* parameter determines rights to access or change authentication functions, such as those in these screens: **Configuration > User Management, Configuration > Policy Management > Access Hours,** and so on. The *General* parameter defines access/change rights for most concentrator functions except for those of the other parameters. The *SNMP* parameter limits

Summary

This chapter showed you the basics of setting up some of the additional features of the concentrator, such as bandwidth management, routing, redundancy with VRRP and VCA, management access, and other important features for using the concentrator for VPN connectivity.

Next up is Chapter 11, "Verifying and Troubleshooting Concentrator Connections," where I show you how to use various concentrator features to troubleshoot problems such as VPN connectivity issues. I'll focus on using the concentrator's monitoring screens for this function, including the Live Event Log and the Filterable Event Log (through the use of event classes).

Chapter 11. Verifying and Troubleshooting Concentrator Connections

This chapter will discuss how to use various VPN 3000 concentrator tools to troubleshoot VPN connection problems. In the first half of the chapter I'll discuss some of the basic tools you'll use on the concentrator and in the second half I'll take a look at using these tools to solve common connectivity problems.

I won't be able to discuss every type of problem you'll see for terminating VPN sessions on the concentrator, but I'll discuss a few of the more common ones in this chapter. In Part III, "Clients," I'll discuss more problems and how to troubleshoot these problems from both the client and the concentrator ends of the connection. In this chapter, though, I'll focus on common problems and how to use various tools on the concentrator to pinpoint the cause of these problems.

PREV NEXT

Concentrator Tools

In the first half of the chapter I'll focus on the Monitoring screens of the concentrator, which up to this point haven't really been discussed. The Monitoring screens allow you to view status and statistical information about what is currently happening on the concentrator, in addition to what has happened previously.

When you click **Monitoring**, you can view the concentrator's status, the status of VPN and administrative sessions, statistics, and event logs. Each of these screens, with the exception of the Live Event Log, display read-only statistics. By default, the statistics on the screen are not updated automatically. Instead, you must click the **Refresh** icon in the top right corner of the window to update the contents on these other screens.

Note

You can change your screen update options by going to the **Administration > Monitoring Refresh** screen. This was discussed in the last chapter, "Concentrator Management."

When you go to the Monitoring screen, you can access the following monitoring options, which take you to different monitoring screens:

- **Routing Table** View the current routing table entries (static and dynamic).
- **Dynamic Filters** View the filters applied to VPN remote access user connections that were retrieved from an AAA RADIUS server during IKE Mode Config.
- **Filterable Event Log** View the log file of events stored in the concentrator's Flash memory; these events are buffered up by the concentrator over a period of time, similar to a Cisco router's or PIX's logging buffered feature.
- **Live Event Log** View the current events as they occur on the concentrator; these events are similar to using the debug feature on a Cisco router or PIX or ASA security appliance.
- **System Status** View the installed software revision, the uptime, and status of the concentrator, including its front-panel LEDs.
- **Sessions** View the current sessions on the concentrator, including VPN and administrative sessions; these can be sorted by type, amount of data transmitted, duration of the connections, or throughput of the connections.
- **Statistics** View various statistics on the concentrator. These include VPN, administrative, Layer-2, Layer-3, and service statistics.

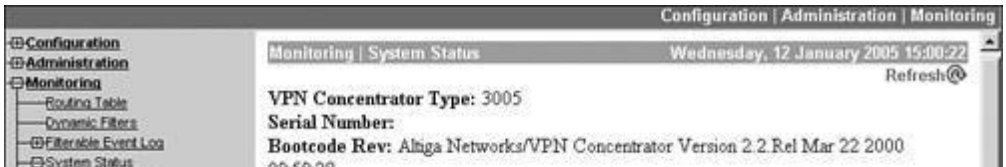
The following sections will discuss some of the more important monitoring screens, listed above, in more depth.

System Status

To view the status of the concentrator, go to **Monitoring > System Status**, where you'll see the screen shown in Figure 11-1. At the top of the screen, you can see the concentrator type (3005), the boot code revision (2.2), the software version (4.1.7A), how long the concentrator has been up (53 minutes, 28 seconds), the date and time the concentrator was either powered up or restarted (01/12/2005 14:06:54), and the amount of RAM (32 MB). For concentrators shipped in the last 45 years, you should also see the concentrator's serial number.

Figure 11-1. System Status Screen

[View full size image]



Troubleshooting Problems

The second half of this chapter will focus on using tools on the concentrator to troubleshoot common problems, including:

- ISAKMP/IKE Phase 1 problems, such as

  - Policy mismatches

  - Authentication problems, including device pre-shared key authentication mismatches, device certificate authentication issues, and user authentication issues

- ISAKMP/IKE Phase 2 transform set mismatches

Note

Following chapters will discuss many more topics on troubleshooting; for example, Chapter 12, "Cisco VPN Software Client," discusses problems with MTU and address translation, and Chapter 19, "Troubleshooting Router Connections," discusses problems with fragmentation.

ISAKMP/IKE Phase 1 Problems

In this section I'll cover some common experiences with ISAKMP/IKE Phase 1 problems and how to use the concentrator's event log to troubleshoot these problems. A good number of remote access and L2L session problems I've dealt with involve either policy mismatches or authentication issues.

IKE Policy Mismatch

One of the more common problems with establishing IPsec sessions is a mismatch in the ISAKMP/IKE Phase 1 policy, what the Cisco VPN concentrators refer to as the "IKE proposals or policies." I discussed the configuration of IKE policies in Chapters 7, "Concentrator Remote Access Connections with IPsec," and 9, "Concentrator Site-to-Site Connections," under the **Configuration > Tunneling and Security > IPsec > IKE Proposals** screen.

In this example problem, I'll use an L2L session between a Cisco router and a concentrator. I've purposely set up the configuration on the concentrator so that there is *not* a matching Phase 1 policy between the two devices. Initially, the logging level was set to 5 for the event log, and the log information from the **Monitoring > Filterable Event Log** screen is shown in Example 11-2. In this example, the remote peer (192.1.1.2) is trying to establish a connection to the concentrator; however, there is an "Error processing payload" error being displayed in event 5, which doesn't illuminate the problem much.

Example 11-2. Initial log of a Phase 1 policy mismatch

```
5 01/16/2005 17:15:34.280 SEV=4 IKE/48 RPT=1 192.1.1.2
Error processing payload: Payload ID: 1

6 01/16/2005 17:15:44.280 SEV=4 IKE/48 RPT=2 192.1.1.2
Error processing payload: Payload ID: 1

7 01/16/2005 17:15:54.280 SEV=4 IKE/48 RPT=3 192.1.1.2
Error processing payload: Payload ID: 1

8 01/16/2005 17:16:04.280 SEV=4 IKE/48 RPT=4 192.1.1.2
Error processing payload: Payload ID: 1
```

Because the logging output in Example 11-2 didn't give me enough detailed information to troubleshoot the problem, I went to the **Configuration > System > Events > Classes** screen on the concentrator and added an event logging class, IKEDBG, and set the logging level to 9 for this event for the event log destination. I then tried to bring the IPsec session up again; this time, the events displayed are more verbose, as shown in Example 11-3. In this example, I can see the actual transforms being negotiated, such as event ID 6, which is the first proposal being negotiated. Event ID 47 states that no compatible proposals were found

Summary

This chapter showed you the basics of using the concentrator's features to monitor and troubleshoot problems. There are four main tools you'll commonly use to monitor sessions and troubleshoot problems:

- Use the **Administration > Administer Sessions** or **Monitoring > Sessions** screen to monitor VPN and administrative sessions terminated on the concentrator.
- Use the Live Event Log to get a quick overview of a problem.
- Use the Filterable Event Log to get more details about a problem.
- If the Filterable Event Log doesn't illuminate enough information to troubleshoot a problem, configure an Event Class with a higher logging level and then use the Filterable Event Log again to pinpoint the problem.

Next up is Part III, "Clients," where I show you how to use various software and hardware clients for remote access sessions. I'll also go into more depth on troubleshooting various components of VPN sessions with remote access users.

Part III: Clients

PREV NEXT

Chapter 12. Cisco VPN Software Client

Part II of this book discussed how to terminate VPN sessionsL2L and remote accesson Cisco VPN 3000 concentrators. In Part III, I'll discuss three remote access clients commonly used in Cisco shops: Cisco VPN Client software, Microsoft Windows' VPN client, and Cisco VPN 3002 hardware client. In Chapter 18, "Router Remote Access Connections," I'll discuss how to use a low-end Cisco router as a client and in Chapter 22, "PIX and ASA Remote Access Connections," I'll discuss how to use a PIX 501 or 506E as one.

To start off Part III, in this chapter I'll discuss the use of the Cisco VPN Client software for Windows to terminate IPsec remote access sessions, sometimes referred to as the "Unity" client. Cisco has moved away from using the term "Unity" to describe the client, though, because it conflicts with a product in their voice product line. Even though the software client can be used to terminate IPsec VPNs on any of the Cisco Easy VPN Server productsconcentrator, router, PIX, or ASAI'll focus on terminating client sessions on the VPN 3000 concentrators. The chapter is broken into six parts:

- An introduction to the software client, including installation of the client and its files and programs
- The GUI interface of the client
- IPsec remote access sessions to an Easy VPN Server, including the use of pre-shared keys and digital certificates
- Additional client components, including Application Launcher, Windows login, auto-initiation, and the stateful firewall features
- Software updates of the client
- Troubleshooting problems with the client's included tools

Cisco VPN Client Overview

The Cisco VPN Client is a VPN remote access client that runs on Microsoft Windows PCs, Linux PCs (Intel-based), Macintoshes (Mac OS X), and Sun UltraSPARC workstations (Solaris). Of the four, the Microsoft and Macintosh clients support a graphical user interface (GUI); the other two use a command-line interface (CLI). The Cisco VPN Client uses IPsec to establish a remote access VPN session to an Easy VPN Server using Cisco Easy VPN technology. Supported servers include the VPN 3000 series concentrators, IOS-based routers, and PIX and ASA security appliances. Other VPN connection methods, such as PPTP, L2TP/IPsec, and WebVPN, are not supported with Cisco VPN Client software. The following sections will discuss the features and installation of the VPN Client for Microsoft Windows.

Note

There are two basic versions of the Cisco software client for Windows: Versions 3.x and 4.x. The GUI interfaces between the two are different; however, how you perform tasks within each client is very similar. Therefore, this book will focus on using the 4.6 Windows client and I'll point out differences between the 3.x and 4.x clients as I proceed through the chapter. Because of space constraints, I will not be covering the non-Windows versions of the client. Also, Cisco used to offer two other clients, but these have been discontinued: the Cisco Secure VPN Client (discontinued in 2003) and the Cisco VPN 5000 Client (discontinued in 2002).

Cisco VPN Client Features

The Cisco VPN Client for Windows (Version 4.6) supports Windows 98, Windows NT 4.0, Windows ME, Windows 2000, and Windows XP platforms. It can be used to establish a secure IPsec session using either a dialup connection via PPP, a wireless connection, or even a LAN-based connection, such as Ethernet. Because the software implements client features, the VPN Client can have only one session active at a time.

There are many, many features that the VPN Client provides; some are based on open standards and some are proprietary to Cisco. For IPsec, it supports both main and aggressive modes for ISAKMP/IKE Phase 1; MD5 and SHA-1 HMAC functions; pre-shared keys, mutual group authentication, digital certificates, and XAUTH user authentication; DH group 1, 2, and 5 keys; and DES, 3DES, AES-128, and AES-256 (AES is new in version 3.6) encryption. To list all of the features would take about a dozen or so pages; therefore, I'll briefly cover some of the VPN Client's more important features in Table 12-1. More information on the Cisco VPN Client can be found at http://www.cisco.com/en/US/products/sw/secursw/ps2308/tsd_products_support_series_home.html.

Table 12-1. Cisco VPN Client features

Feature	Version	Description
Application Launcher	3.0	Launches an application when establishing an IPsec session to an Easy VPN Server
Auto-Initiation	3.6	Automatically initiates an IPsec session during bootup of the PC or when an IPsec session is dropped
Automatic Dialup Connection	3.0	Automatically dials an ISP or access server using Microsoft's or a third party's dialup software to establish an IPsec session
Automatic Start Before Login and Automatic Disconnect	3.0	Allows the VPN Client to bring up an IPsec session first before the user logs in to the Windows domain; likewise, allows the VPN session to terminate if the user logs out of the domain
Automatic Updates	4.6	Allows Windows 2000 and XP clients to download and install a software update automatically; versions earlier than this only receive a notification, and then the user must manually download

Cisco VPN Client Interface

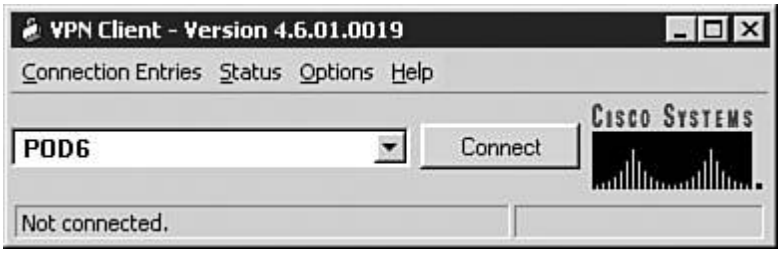
The Cisco VPN Client for Windows supports two interfaces: CLI and GUI. This book will focus on the GUI interface of the 4.6 client. Once you have installed the client, go to **Start > Programs > Cisco VPN Software Client > VPN Client** to access the GUI. Sometimes the application is referred to as the "VPN Dialer," after the older 3.x application name.

Operating Modes

The VPN Client has two operating modes:

- Simple Mode, shown in Figure 12-1

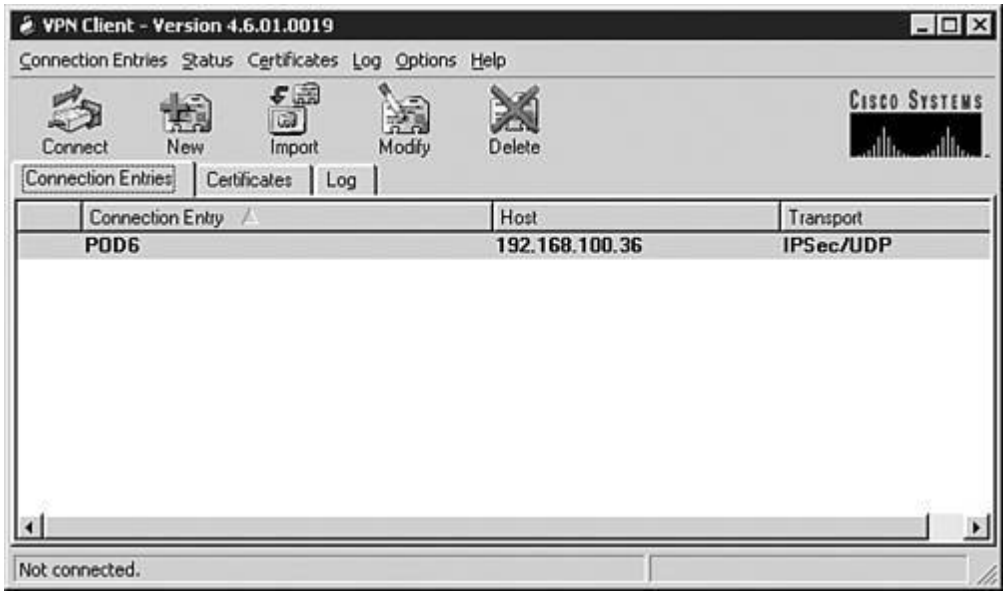
Figure 12-1. VPN Client GUI: Simple Mode



- Advanced Mode, shown in Figure 12-2 (this is the default mode)

Figure 12-2. VPN Client GUI: Advanced Mode

[View full size image]



With either mode, at the top of the window in the window bar is the name of the application and version number: "VPN ClientVersion 4.6.01.0019."

Note

You also can view the client type and version by right-clicking the VPN Client IPsec session icon (padlock) and choosing **About VPN Client**.

Below the window bar are the menu options. You'll notice that there is a difference between what the Simple and Advanced Modes display. Table 12-2 explains the menu options available for each mode. To toggle between the two modes, go to Options and choose **Advanced Mode** if you're currently in Simple Mode and **Simple Mode** if you're in Advanced Mode. For the most part, the remainder of this chapter will focus on the use of the Advanced Mode display.

Table 12-2. VPN Client Mode Menu Options

IPsec Connections

This section will discuss how to connect to an Easy VPN Server using pre-shared keys and certificates for device authentication via Advanced Mode. As you go through this part of the chapter, note that you can connect your Cisco client to a VPN 3000 concentrator running at least Version 3.0, a PIX running 6.2.2(122) or 6.3(1), an ASA running 7.0, or an IOS router running 12.2(8)T with IPsec. For a 4.6 client, the corresponding concentrator code is 4.1.6, but 3.0 is the minimal version required on a concentratorjust realize that you'll be missing out on a lot of features using a concentrator with an older version of code than a client and vice versathe same is true of a PIX, ASA, or router as an Easy VPN Server.

Creating Connections using Pre-Shared Keys

To create a new connection on your VPN Client, you can either:

- Click the **Connection Entries** tab and then click the **New** button in the toolbar.
- From the main menu, choose **Connection Entries > New**.

When you perform one of the two above processes, you are shown the screen in Figure 12-3. At the top of the screen you need to name the connection profile in the *Connection Entry* text box; this name must be compatible with the file-naming conventions used on your PC. In Figure 12-3, I called the profile "Corporate Office Connection." In the *Description* text box, you can enter an optional description. Below this is the *Host* text box, where you must enter the IP address of the Easy VPN Server the client will be connecting to. The next sections will discuss the four configuration tabs when adding a connection: **Authentication**, **Transport**, **Backup Servers**, and **Dial-Up**.

Figure 12-3. Adding a Connection: Pre-shared Keys in the Authentication Tab

Authentication Tab

If you'll be using pre-shared keys to perform device authentication during ISAKMP/IKE Phase 1, then in the Authentication tab, you'll need to click the *Group Authentication* radio button and then enter the name of the group the user belongs to, and the pre-shared key (password) twice for verification. This information needs to match what is configured on the Easy VPN Server. In Figure 12-3, the group name I configured was "Executives." If you don't

VPN Client GUI Options

I've already discussed two of the options available from the VPN Client's Options in the main menu: Preferences and the Mode selection (Simple and Advanced). This part of the chapter will discuss the other three options: Application Launcher, Windows Login Properties, and Stateful Firewall (Always On).

Application Launcher

The Application Launcher option from the Options selection in the main menu of the VPN Client allows you to run an application before establishing a session to an Easy VPN Server. You might want to do this if you've configured Start Before Logon in the Windows Login Properties of the VPN Client and you need to authenticate to some other application before bringing the tunnel up, or you need to start another application for monitoring functions before each session is established.

By default, this option is disabled. When you go to **Options > Application Launcher** from the main menu, click the *Enable* check box to enable it and then click the **Browse** button to find the application to launch. Click the **OK** button to save your changes.

Note

To use the Application Launcher feature, you also must select the *Allow launching of third party applications before logon* option in the Windows Login Properties, discussed in the next section.

Windows Login Properties

The Windows Login Properties, which you can access by going to **Options > Windows Login Properties** from the main menu, allow you to specify special login functions for Windows NT 4.0, 2000, and XP platforms (for other platforms, this will not appear as a selection in the menu tree). There are three check boxes in this window that allow you to:

- **Enable start before logon** Allows you to connect to the network via IPsec before you log on to your Windows domain. This might be necessary if the domain controller is located behind the Easy VPN Server. When this option is used, you log in to your domain after the IPsec tunnel is established.
- **Allow launching of third party applications before logon (Application Launcher, Third party dial-up application)** Allows you to launch an application before you log in to your Windows domain.
- **Disconnect VPN connection when logging off** Has the VPN Client automatically disconnect itself whenever you log out of the Windows domain. Of the three parameters, only this one is enabled by default. About the only time you would want to disable this feature is if your network is using Windows roaming profiles.

Automatic Initiation

Automatic initiation, commonly called auto-initiation (AI), allows computers to establish IPsec sessions automatically using the Cisco VPN Client. AI, new in Version 3.6, was developed primarily for wireless environments, but can be used in other situations. AI is disabled by default. When enabled, the VPN Client becomes active immediately whenever the following are true:

- The computer boots up or the computer comes out of a standby or hibernating state.
- The computer has an IP address assigned to a NIC that is associated with AI.

When both of these items are true, AI will use the appropriate connection profile to automatically set up a secure session to the Easy VPN Server in the associated connection profile. This is very useful in wireless LANs (WLANs) where you want to make the VPN process as transparent as possible to the user. In most WLANs conscious of security, a VPN is used to protect the wireless traffic when it is sent to the wired LAN. For example, depending on which

VPN Client Software Updates

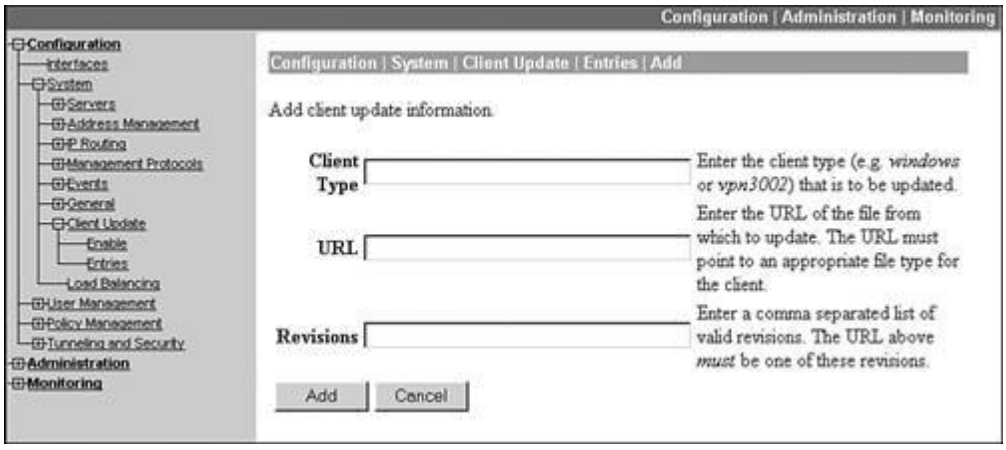
In this section I'll discuss how to upgrade the VPN Client software. Cisco VPN 3000 concentrators support two types of automatic client upgrades: the 3002 hardware client and the Windows VPN Client. I'll cover what you have to do on both the concentrator and VPN Client side of the connection.

Concentrator: Client Updates

There are two types of client updates or upgrades the concentrator supports: the Cisco VPN client software and the 3002 hardware client. Upgrades of both can be controlled on the concentrator. If you go to the **Configuration > System > Client Update > Entries** screen on the concentrator and click the **Add** button, you can add an update entry. This screen is shown in Figure 12-19.

Figure 12-19. Adding Client Updates on a Concentrator

[View full size image]



There are three parameters on the screen in Figure 12-19. The *Client Type* parameter specifies the type of client you want to upgrade. All versions of the concentrator allow you to upgrade Windows and 3002 clients. If you enter "Windows" as the parameter, this update entry applies to all Windows clients: "Win9x" applies to only Windows 95, 98, and ME; and "WinNT" applies to only Windows NT 4.0, 2000, and Windows XP. For Windows clients, your update entries should not overlap. In other words, don't specify both "Windows" and "Win9x" because this would cause the concentrator to send two update notices to the client. To specify an update for a VPN 3002 hardware client, use "vpn3002." Newer concentrator images support the upgrade of MacOS, Solaris, and Linux clients using these client types, respectively: "Mac OS X," "Solaris," and "Linux." Please note that upgrading of the non-Windows clients is a fairly new feature with the concentrators and clients.

Note

You must enter the *Client Type* in the exact case and spacing, or the client will ignore the update message.

The *URL* parameter specifies the location of the client update file (typically on a web server behind the VPN 3000 concentrator). For a 3002 client, the update must be a TFTP server, so you would enter something like this: tftp://192.168.1.1/vpn3002-4.1.7.Rel-k9.bin. Make sure you use the same case that Cisco specified on their web site. For all other clients (software-based), the update location must be a web server URL ("http" or "https"), like this: http://IP_address/client_image. With both types of URLs, you can include directories; and for HTTP URLs, you can include port numbers if the web server is running on a different port. You can also use a name for the server, but you must configure a DNS server on the concentrator to resolve the name to an address.

The *Revisions* parameter specifies what version or versions of software the clients should be running; if they are not running these versions, they should install the software in the *URL* parameter. You can specify multiple revision numbers; just separate them by a command and a space.

VPN Client Troubleshooting

The remainder of this chapter will focus on troubleshooting some common problems when using the Cisco VPN Client software. The first topic I'll discuss is the logging abilities of the software client: the Log Viewer component of the VPN Client. Then I'll discuss some common problems you'll typically come across and what to look for on your computer (or concentrator) to troubleshoot these problems:

- Authentication Problems
- ISAKMP/IKE Policy Mismatch Issues
- Address Assignment Troubleshooting
- Split Tunneling Problems
- Address Translation Problems
- Fragmentation Issues
- Microsoft Network Neighborhood Issues

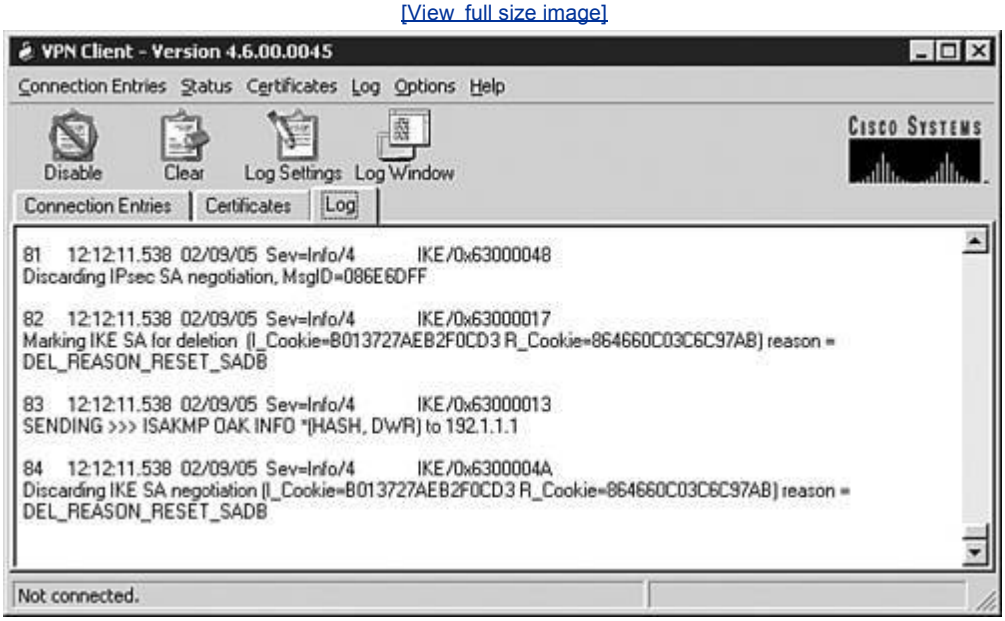
Log Viewer

In the 3.x version of the VPN Client software, the Log viewer was a separate application; in 4.x, the Log application was integrated into the VPN Client software. To view the events in the log file, perform one of the following in the 4.x GUI:

- Click the **Log** tab in the middle of the windowthis can be seen in Figure 12-22.

Figure 12-22. Log Event Viewer

[View full size image]



- From the main menu you can choose **Log > Log Window** to pull up this window.

The information in the Log event viewer window can be used to help in troubleshooting client problems. You can control the amount of logging information, search the logging information, and view the logging information from this window.

Formatting of Logging Information

Events in the Log window have the following format in two or more lines:

```
event_# time date Sev=type/level event_class/message_ID
message_description
```

Each event is assigned a unique number in sequential order. Following this is the time and date of the event, and then the type and severity level of the event. After this is the event class and message ID.

Summary

This chapter introduced you to the Cisco VPN Client for Windows. Installing, configuring, and managing it is fairly easy. And with 4.6 and the auto-update feature, managing it becomes even easier. In the last part of the chapter I devoted quite a few pages to common problems you'll see when setting up remote access sessions. The most common of these problems are: address translation, fragmentation, and Microsoft Network access. Using the client Log event viewer and Windows command tools makes most troubleshooting tasks a simple process.

Next up is Chapter 13, "Windows Software Client," where I show you how to configure Microsoft's Windows Software Client for remote access connectivity to the VPN 3000 concentrators.

Chapter 13. Windows Software Client

The last chapter discussed the use of the Cisco VPN Client to establish remote access IPsec sessions to a Cisco VPN gateway (Easy VPN Server) product, such as a concentrator, IOS router, or PIX firewall. Because many companies have policies that center on Microsoft products, you might have to use a Microsoft client to establish connections to a Microsoft VPN gateway. Likewise, your users might have to establish VPN connections to a Cisco gateway and a Microsoft gateway, requiring you to use both client products. This chapter will focus on using Microsoft's VPN client to establish PPTP or L2TP/IPsec connections to a VPN gateway.

Note

Even though Microsoft's Windows VPN client is obviously not a Cisco product, and this book is about Cisco VPNs, as a consultant I'm constantly asked questions about how to set up PPTP or L2TP over IPsec (L2TP/IPsec) sessions from a Microsoft client to a Cisco VPN gateway. Therefore I've decided to include this chapter in the book to help those people who must deal with this type of connectivity issue. This chapter, however, will focus only on the use of the client with the Windows 2000, XP, and 2003 operating systems, even though Microsoft has a different client that will run on older operating systems.

To help you with using Microsoft's VPN client, the following sections cover these topics:

- Windows Client
- Configuring the Windows VPN Client
- Configuring the VPN 3000 Concentrator
- Microsoft Client Connections
- Troubleshooting VPN Connections

Windows Client

Originally, the Microsoft Windows client software was developed for Remote Access Server (RAS) environments. Normally, you think of a remote access VPN as a solution that protects traffic from the user desktop to the VPN gateway at the corporate site, but Microsoft added flexibility into the design with PPTP and L2TP/IPsec to allow an intermediate device, typically an RAS, to perform this process on behalf of the client. In this situation, the client would dial into the RAS with a clear-text PPP connection, authenticate via PPP, and then request the RAS to set up a PPTP or L2TP/IPsec encrypted connection to the corporate RAS. Using this process offloads the protection process to the RAS instead of to an underpowered user PC.

Today, of course, most PCs and laptops should have no problem handling the processing required to protect traffic for a VPN. Therefore, in today's networks, most PPTP or L2TP/IPsec sessions start with the remote access user and terminate at the remote corporate office VPN gateway (see Chapter 4, "PPTP and L2TP," for more information on the connection and operation process of these protocols).

Understanding Features of the Windows Client

The current Windows client supports L2TP over IPsec (L2TP/IPsec) for VPN sessions, but also supports PPTP. With the L2TP/IPsec client, you can use either pre-shared keys or digital certificates for authentication. If you recall from Chapter 4, both protocols rely on PPP to perform authentication, provide protection services, and transport data.

Because of the encryption strength of 3DES, it is recommended to use L2TP/IPsec rather than either L2TP or PPTP with MPPE's encryption. Whereas 3DES supports 168-bit encryption, the highest that MPPE supports is RC-4's 128-bit encryption; and where MPPE provides only data confidentiality (encryption), IPsec provides data confidentiality, data origin authentication (using a hashing function), data integrity (using a hashing function), and anti-replay protection. Another concern with PPTP is that of security issues surrounding the use of MSCHAPv1 and v2 for authentication. Therefore, this chapter will focus on the use of L2TP over IPsec.

Cisco VPN Client versus Microsoft's L2TP/IPsec Client

Obviously both clients support IPsec; however, even though they support IPsec, this basically means that they follow the first three steps in ISAKMP/IKE Phase 1 (negotiate the IKE parameters, use DH, and perform device authentication with either pre-shared keys or certificates) and ISAKMP/IKE Phase 2. As I pointed out in Chapter 7, "Concentrator Remote Access Connections with IPsec," Cisco has added some additional functionality to ISAKMP/IKE Phase 1 which is not part of the IPsec standards. For example, Cisco supports the following for their software and hardware clients: split-tunneling (supported only in XP by the L2TP/IPsec client) and split DNS, client type and version limiting, backup server lists, IPsec over TCP, client and network extension modes, reverse route injection, load balancing, and firewall policies, to name a few. When using a Microsoft L2TP/IPsec client, you don't have access to these features when terminating the client on a Cisco VPN gateway. In other words, using Cisco VPN products, for both the gateway *and* client devices, gives you much more centralized control over policies on the gateway device. Another disadvantage of Microsoft's client is that two levels of encryption will be used for dataIPsec and MPPEadding additional overhead.

However, this is not to say that you should never use Microsoft's client. It does have some advantages related to the use of L2TP over IPsec. For example, because PPP is used, you can run multiple protocols across the connection, such as IP and IPX (Cisco VPN 3000 concentrators, though, only support IP). Plus, the client comes pre-installed on Windows 2000 and higher computers, so no extra software needs to be installed. You can even create a special installation package that installs all IPsec policies and VPN connection profiles. And before the release of Cisco VPN Client 4.6, upgrading the Cisco software client was not very easy.

Verifying that the Windows Client is Operational

Configuring the Windows VPN Client

Once the Windows L2TP/IPsec client is running, you are ready to create a connection profile to protect traffic from the client to the corporate office. Creating a connection profile is done in four steps:

1. Obtaining certificates (optionalobtaining certificates on Microsoft Windows desktops is beyond the scope of this book; please consult your local Microsoft administrator for more details): for Windows 2000 and earlier, you will need certificates to use L2TP/IPsecpre-shared keys are not supported

2. Creating a security policy

3. Requiring the use of IPsec for L2TP

4. Creating a VPN connection

The following sections will discuss the last three steps in more depth.

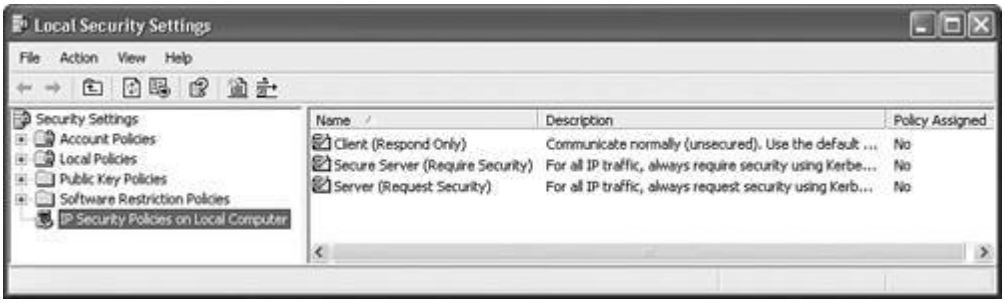
Creating a Security Policy

Before you can create a VPN connection, you'll want to create a security policy. The security policy is used to define the method of authentication, such as pre-shared keys or certificates, and how the connection is protected. Here are the steps to create a security policy that your VPN connection will use:

1. Go to **Start > (Programs) > Administrative Tools > Local Security Policy** and click **IP Security Policies on Local Machine** under the **Security Settings** heading, shown in Figure 13-2.

Figure 13-2. Local Security Settings Window

[View full size image]



2. Right-click **IP Security Policies on Local Machine** and choose **Create IP Security Policy**.

3. The IP Security Policy Wizard will begin. Click the **Next** button to go to the next screen.

4. You are taken to the IP Security Policy Name window. Enter the name of the security policy; for example, "Corporate Network Policy." Give it something descriptive that defines what the policy will be used for. Optionally, you can enter a multi-lined detailed description of the policy. Click the **Next** button when done.

5. You are taken to the Requests for Secure Communication window. This window allows you to activate the default response rule to use when the remote VPN gateway requests a security policy that doesn't match any that have been defined. There is only a check box on this screen labeled *Activate the default response rule*it is recommended to keep this checked. Click the **Next** button when done.

6. You are taken to the Default Response Rule Authentication Method window, shown in Figure 13-3. Here you define the type of authentication to use with the VPN gateway: Windows 2000 with Kerberos v5, certificates (if you've installed them), or pre-shared keys. In this example, I entered a pre-shared key of "cisco." This also will need to be

Configuring the VPN 3000 Concentrator

Now that I've explained how to configure the Microsoft client, I'll focus on things you need to configure on Cisco VPN 3000 concentrators to support Windows client access. I covered many of these screens in Chapter 8, "Concentrator Remote Access Connections with PPTP, L2TP, and WebVPN," so I'll briefly explain the items that you must create or change to support Microsoft connections in the following sections. Here are the basic items you need to configure or add to your concentrator's configuration:

- IKE Proposals
- IPsec SAs
- Group Configuration: Specific and/or Base Group
- Address Management
- User Configuration

IKE Proposals

First, on your concentrator you need to ensure that you either have created or activated an IKE proposal that will work with the Microsoft L2TP/IPsec client (assuming that you're not using plain L2TP or PPTP). On the concentrator, go to **Configuration > Tunneling and Security > IPsec > IKE Proposals.**

Proposal names beginning with "CiscoVPNClient" will *not* work with Microsoft's L2TP/ IPsec clientthe Microsoft client doesn't understand the concept of groups and XAUTH. Therefore, you need to choose a proposal that doesn't do XAUTH authentication, like those beginning with "IKE."

IKE proposals that will work include these properties:

- **Authentication mode** Pre-shared keys and RSA and DSA certificates
- **Authentication algorithm** MD5 or SHA1
- **Encryption algorithm** DES or 3DES
- **Diffie-Hellman Group** Group 1 or 2

If you'll be using pre-shared keys, two of the included concentrator proposals that will work are IKE-DES-MD5 and IKE-3DES-MD5just make sure you haven't deactivated these security policies for your L2TP/IPsec client (see the "Creating a Security Policy" section earlier in the chapter). Refer to Figures 7-18 and 7-19 in Chapter 7, "Concentrator Remote Access Connections with IPsec" for examples of these screens.

Note

I've experienced a lot of headaches with customizing Microsoft's L2TP/IPsec client. In many instances, I've had to use an IKE Policy that had only DES for encryption, DH group 1 keys, and either MD5 or SHA for an HMAC function because of the quirks in Microsoft's software.

IPsec SAs

Next, on the VPN 3000 concentrator, create an IPsec SA transform that will be used to protect the ISAKMP/IKE Phase 2 data connections for the Windows client. To do this, go to **Configuration > Policy Management > Traffic Management > SAs.** Click the **Add** button to add a new SA, and then enter the following for the SA parameters:

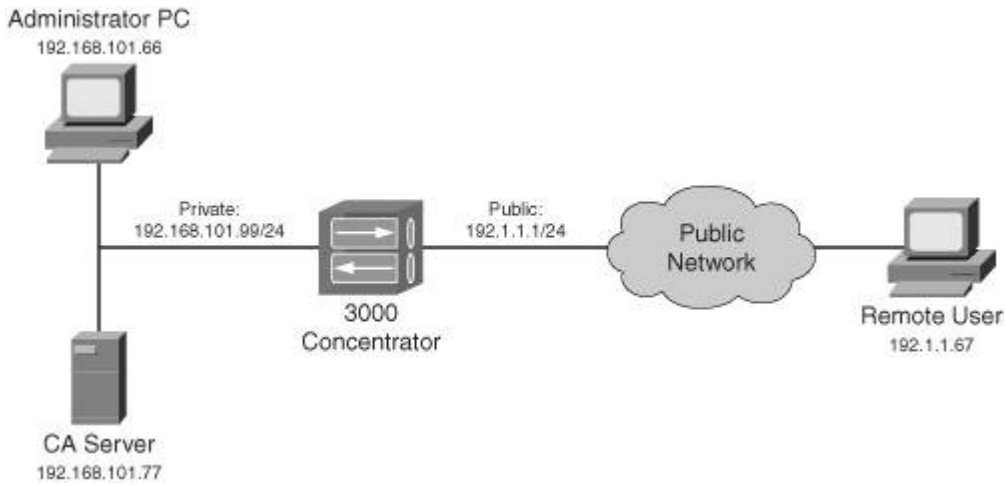
- **SA Name** "ESP-L2TP-Transport," or something similar
- **Authentication Algorithm** ESP with MD5 or SHA
- **Encryption Algorithm** DES or 3DES

Microsoft Client Connections

Now that you have configured the Microsoft client(s) and VPN 3000 concentrator, the client can now establish a connection to the concentrator. The following sections will discuss how to establish a connection from the client to the concentrator. The network shown in Figure 13-21 illustrates the process.

Figure 13-21. L2TP/IPsec Client and VPN 3000 Concentrator Example

[View full size image]



Connecting to a VPN Gateway

On the Microsoft computer, open the Network Connections window in one of the following ways:

- On Windows 2000, go to **Start > Settings > Network and Dialup Connections,** right-click and choose **Explore**.
- On Windows XP, go to **Start > My Network Places > View network connections**.

In this window there should be a section entitled *Virtual Private Network,* listing the VPN connections you have set up from the "Creating a Microsoft VPN Connection" section earlier in the chapter. Their statuses should say "Disconnected." Either double-click the name of the VPN connection profile or right-click the name and choose **Connect**. You should see the Connection window, shown in Figure 13-22.

Figure 13-22. Microsoft VPN Client Connection Window

Troubleshooting VPN Connections

Of course, once you have created the configuration on the client and concentrator, there's no guarantee it will work the first time you try to bring up a session. I've already talked about a few troubleshooting processes related to routing previously in the chapter. Now I'll focus on tools you can use on the concentrator and the Microsoft client.

Concentrator Troubleshooting Tools

I already have discussed most of the tools you can use on the concentrator to troubleshoot connectivity problems (see Chapter 11, "Verifying and Troubleshooting Concentrator Connections"). First, I would use the Filterable Event Log, since it is less CPU-intensive for the concentrator, and it supports filtering actions. Because the events displayed in the log file typically are captured at levels 35, you might need to add an event class (**Configuration > System > Events > Classes)**. Here are the classes that would apply to Microsoft clients:

- **AUTH** Authentication functions (such as PAP or MS-CHAP) with PPP
- **CERT** Device authentication with certificates when using L2TP/IPsec
- **IKE** ISAKMP/IKE Phase 1 problems
- **IPSEC** ISAKMP/IKE Phase 2 problems
- **L2TP** L2TP problems
- **PPP** L2TP's and PPTP's use of PPP (like MPPE)
- **PPTP** PPTP problems

Use the Filterable Event Log to get an idea as to what the problem might be and then add the required event class(es). Then go back to the Filterable Event Log to pinpoint the problem.

Tip

When done troubleshooting, be sure to delete the event classes you added; otherwise you'll be capturing more logging information than necessary.

Microsoft Client Troubleshooting Tools

In most cases, you'll be able to troubleshoot the problem from the concentrator's end; however, in those instances when you can't, or if you're not connecting to a VPN 3000 concentrator, there are quite a few things you can do from the Microsoft client end of the connection to pinpoint the problem. The following sections will discuss some of the things you can do to help troubleshoot VPN connection problems on a Microsoft platform. The tools I focus on are used on a Windows XP system; however, many of them will work on Windows 2000 and earlier systems.

IP Security Monitor Snap-In

The IP Security Monitor snap-in application for MMC allows you to view your security policy settings, and the policies being used for any existing L2TP/IPsec connections. To add the IP Security Monitor Snap-in to MMC, follow these instructions:

1. Go to **Start > Run** and enter **mmc**; then click the **OK** button.
2. Go to **File > Add/Remove Snap-in**.
3. In the **Standalone** tab, click the **Add** button.
4. Find **IP Security Monitor**, click it, and then click the Add button.
5. Click the **Close** button in the Add Standalone Snap-in window and the **OK** button in the Add/Remove Snap-in window.

Summary

This chapter introduced you to the setup and configuration of a Windows-based PC to use the PPTP, L2TP, and L2TP/IPsec protocols to secure connections to a VPN gateway. Because of the limited use of centralized policy control, I prefer to use Cisco VPN clients, like the Cisco VPN software client, the 3002 hardware client, the PIX 501 and 506E security appliances, and the 800, UBR900, and 1700 series routers as remote access clients. However, if you need to support both types of clients, the Cisco VPN gateway products can easily accommodate this need.

Next up is Chapter 14, "3002 Hardware Client," where I show you how to use the Cisco 3002 hardware client to set up secure connections with SOHO sites to a VPN 3000 concentrator at the corporate site.

PREV NEXT

Chapter 14. 3002 Hardware Client

The last two chapters focused on the Cisco VPN Client and Microsoft's L2TP/IPsec client. Both of these clients are software clients, where VPN software is installed on a PC. Software clients allow a single user to secure a session to a remote VPN gateway. But software clients don't scale well, especially in environments where more than one person at a location needs to establish a secure session to a remote destination. In this case, it would be better to use a centralized device, like a remote access hardware appliance or a site-to-site gateway device.

This chapter will focus on using one of the Cisco hardware clients: the VPN 3002 hardware client. I'll discuss the 3002's features and deployment options, your initial access to the 3002, connection and authentication options, and administrative tasks. Chapters 18, "Router Remote Access Connections," and 22, "PIX and ASA Remote Access Connections," will discuss two other Cisco hardware clients: low-end routers and PIXs.

Overview of the 3002 Hardware Client

The 3002 is a hardware version of the Cisco VPN software client. Like the software client, it is easy to use, but provides much more scalability where a large number of SOHO sites need to connect to a central site. It is easy to deploy and support; most IPsec functionality is hidden from the user who has to deploy and manage it, just as is the case with the Cisco VPN Client software. The 3002 fully supports Cisco Easy VPN Remote technologies and is used as a remote access device, protecting traffic for users behind it.

Because the 3002 is an Easy VPN Remote, you can centralize policies on an Easy VPN Server and push them down to the 3002. The 3002 supports two modes for protecting users' traffic to a central site: client and network extension modes (these will be discussed later in the "Connection Modes" section). The following sections will discuss the features, models, and deployment options for the 3002.

3002 Features

The 3002 is easy to deploy and support. It already is a DHCP client on its public interface, acquiring addressing information dynamically from the ISP (the 3002 also supports PPPoE on its public interface in addition to static IP addressing). It has a static IP address on its private interface and is a DHCP server on the private interface, giving out addressing information to internal devices. It supports a Quick Configuration process similar to the VPN 3000 concentrators, making it easy to set it up to connect to an Easy VPN Server. Here are some of the 3002's many features:

- It is fully IPsec-compliant and supports the following IPsec functions: Automatic configuration of ISAKMP/IKE Phase 1 Policies and Phase 2 transformsyou don't need to configure these on the client; authentication with pre-shared keys and digital certificates; DH groups 1, 2, and 5; ESP in tunnel mode, NAT-T, IPsec over UDP, and IPsec over TCP; DES, 3DES, AES-128, and AES-256 encryption algorithms; and SHA-1 and MD5 HMAC functions.

  ### Note

  Due to limited room for storing transforms on the 3002, it does not support AES-192 for encryption.

- It supports Path MTU discovery and manual MTU adjustment to reduce the likelihood of IPsec fragmentation.
- It can be upgraded from policies defined by an Easy VPN Server and pushed down to the 3002 during IKE Phase 1 in the IKE Mode Config step, allowing the 3002 to automatically download a new software image from a TFTP server and then reboot itself.
- It supports client and network extension modes. In client mode, the 3002 is assigned a single internal address and PATs all user's addresses to the internally assigned address from the concentrator; this makes all devices at the SOHO appear as one logical device from the corporate office's perspective. In network extension mode, the 3002 simulates an L2L tunnel, where no address translation is performed on the SOHO devices' addresses, making them appear as an extension of the corporate office network.
- It provides authentication credentials to bring up the IPsec tunnel (default or unit authentication), or one (interactive unit authentication) or more users (user authentication) can provide this information.
- It is fully compliant with the load balancing features of the VPN 3000 concentrators and ASA security appliances via VCA, and supports a backup server list for redundancy.
- It supports up to 253 devices behind its private interface.

3002 Models

Initial Access to the 3002

The 3002 hardware client supports both a command-line interface (CLI), which is menu-driven, and a graphical user interface (GUI). Both are discussed in this part of the chapter. The 3002 has the following default configuration:

- DHCP client on its public interface
- Static IP address of 192.168.10.1 on its private interface
- DHCP server on its private interface

So, unlike with the VPN 3000 concentrators, you don't need to use the CLI to initially configure the 3002. Because the 3002 has an IP address on its private interface, you just need to set up your PC to acquire its addressing information via DHCP and then use a web browser and point it to 192.168.10.1.

Command-Line Interface

The menu-drive CLI is used primarily for troubleshooting of the 3002 when you cannot access the 3002 via a web browser; otherwise, the web-based GUI is used to manage and troubleshoot the 3002. Accessing the CLI of the 3002 typically is done via the console port with a Cisco rollover cable. When you access the CLI, you'll be prompted to log in to the 3002. The default username and password are just like those for the 3000 concentrators: "admin" for both.

Example 14-1 shows a simple login process. As you can see from the CLI output, the interface looks the same as the CLI on the 3000 concentrators; how you move around the CLI is the same as on the concentrators (See Chapter 6, "Concentrator Product Information" for more information about using the concentrator's CLI). However, because the 3002 is a client, you will not see all the options you see on the concentrator's CLIand vice versa.

Example 14-1. 3002 CLI Login and Main Menu

```
Login: admin
Password: admin


               Welcome to
              Cisco Systems
         VPN 3002 Hardware Client
          Command Line Interface
Copyright (C) 19982004 Cisco Systems, Inc.


1) Configuration
2) Administration
3) Monitoring
4) Save changes to Config file
5) Help Information
6) Exit


Main ->
```

Graphical User Interface

With few exceptions, you'll be using a web browser to manage the 3002. Because the 3002 already is set up with an IP address on its private interface (192.168.10.1) and is acting as a DHCP server on this interface, you only need to connect a PC to this interface and set it to acquire its addressing via DHCP. Then point your web browser to the 3002 like this: **http://192.168.10.1.** You'll be presented with the login screen shown in Figure 14-2.

Figure 14-2. 3002 Web Access

Authentication and Connection Options

Once you have used Quick Configuration to set up the 3002 or have modified its configuration, you'll want to verify that an IPsec tunnel can be brought up to the configured VPN gateway. The following sections will discuss three possible options you can use to perform authentication: Unit Authentication, Interactive Unit Authentication, and Individual User Authentication. If the 3002 will be connecting to a VPN 3000 concentrator, you also must enable the correct policy on the concentrator. Under the "Building the IPsec Tunnel" subheading, I also will discuss three possible options of performing authentication for use of the IPsec tunnel when Interactive Unit or Individual User Authentication is enabled.

Unit Authentication

The default authentication method on the 3002 is called Unit Authentication, sometimes referred to as the default method. With the default method, the XAUTH remote access username and password are configured on the 3002 itself; therefore, no users behind the 3002 have to perform any type of authentication to either bring up the IPsec tunnel or to use an existing tunnel.

With the default method, there are three ways of bringing up the IPsec tunnel to the VPN gateway: one brings up the tunnel automatically and two require user intervention to bring up the tunnel. Here are the three connection options (these only apply to client mode connections, because network extension mode automatically brings up the tunnel):

- **Automatically** A user behind the 3002 sends traffic destined beyond the 3002. Even if the traffic is not meant for the corporate site, the 3002 still will bring up the tunnel to determine what the split tunneling policy is.
- **Manually** With this approach there are two options:

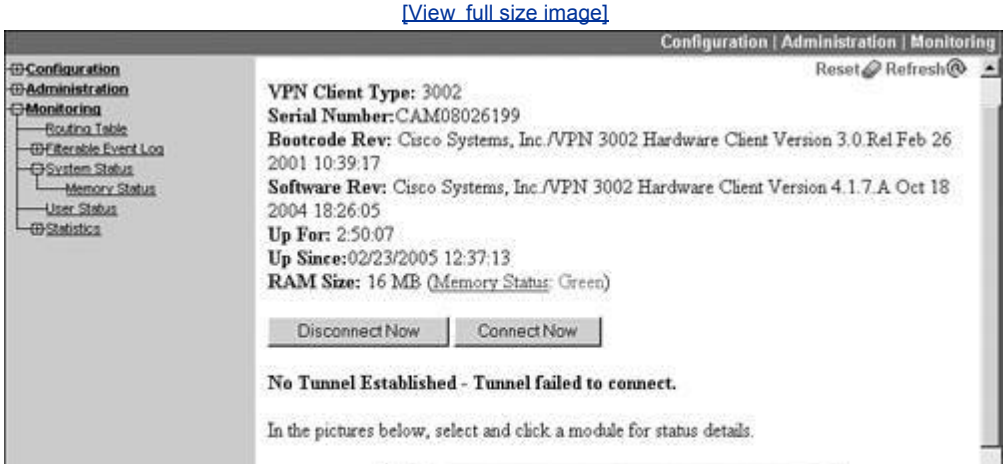
  - A user behind the 3002 opens up a web browser connection to the private interface of the 3002, and in the web browser window clicks the **Connection/Login Status** hyperlink and then clicks the **Connect Now** button.

  - A user behind the 3002 opens up a web browser connection to the private interface of the 3002 and logs in to one of the three accounts: admin, config, or monitor. Only the first one is enabled, by default. To enable the other two, go to **Administration > Access Rights > Administrators** after logging in to the admin account. For user access, I would enable only the monitor account, because it has read-only privileges on the 3002; mostly, it's restricted to the Monitoring menu structure. Once logged in, the user goes to the **Monitoring > System Status** screen and clicks the **Connect Now** button to bring up the tunnel. Of the two manual options, this is the least preferred because you are letting SOHO users log in to the 3002, which might present a security risk.

For testing purposes, as an administrator, you'll probably go to the **Monitoring > System Status** screen to bring up the tunnel. When no tunnel is up, you'll see the screen shown in Figure 14-15.

Figure 14-15. System Status Screen: No Tunnel

[View full size image]



Connection Modes

The Cisco hardware clients, including the 3002, support two connection modes to an Easy VPN Server: client mode and network extension mode. Both of these modes were discussed in Chapter 3, "IPsec." The following sections will discuss the two modes and how to enable them.

Client Mode

The top part of Figure 3-3 from Chapter 3 shows an example of client mode. As a quick review, in client mode, the Easy VPN Server assigns the remote access hardware client a single, internal IP address. For devices behind the hardware client to access the corporate site across the tunnel, the hardware client performs PAT on the inside packets to the internal IP address assigned to it by the Server. From the corporate office's perspective, it looks as if a single device is connected to the network, when in reality, it could be a few hundred (with the 3002, the limit is 253 devices off of the private interface).

Note

The default behavior of the 3002 is to use client mode; in other words, it's already set up to PAT addresses going across the tunnel.

Network Extension Mode

The bottom part of Figure 3-3 shows an example of network extension mode. Network extension mode simulates an L2L tunnel. Please note that it is *not*, however, an L2L session. The hardware client, like the 3002, is still a client and connects to the Easy VPN Server, where in ISAKMP/IKE Phase 1, XAUTH and IKE Mode Config occur (which is not the case with a true L2L tunnel session). The main difference is that the remote access client will share the network number and subnet mask of its private interface, during ISAKMP/IKE Phase 1, with the Server via reverse route injection (RRI).

The main limitation of client mode is that the central office cannot initiate a connection to a device behind the hardware client because PAT is being used. With network extension mode, no address translation occurs; typically, the network number off of the private interface of the hardware client is a unique subnet of a network used from the corporate office. Using this mode, a device at the central office can access a device at the SOHO easily. This might be necessary for management, file-sharing, or VoIP purposes.

Note

In network extension mode, only the directly connected network of the private interface is advertised to the Easy VPN Server. If you had more than one network behind the hardware client, these additional networks would be unknown to the VPN gateway. Therefore, if you had multiple subnets behind your hardware client, you would have to use client mode or use a different VPN device that supported L2L sessions.

The configuration of network extension mode involves three configuration steps: two on the 3002 and one on the Easy VPN Server. The following sections will discuss the setup of these items on both the 3002 and VPN 3000 concentrators.

Network extension mode version L2L sessions

You might ask, Why not use an L2L session instead of using network extension mode? If you recall from Chapter 3, "IPsec," an L2L session goes through three basic steps in ISAKMP/IKE Phase 1: Negotiate the Phase 1 policies, use DH to share keying information, and perform device authentication. Network extension mode requires a hardware client device, which means additional steps will occur: XAUTH, IKE Mode Config, and RRI, the latter being discussed later in the "Routing and Reverse Route Injection" section.

Using a hardware client, you have more control over policy configuration and enforcement than you would over an L2L session. Plus, if the Easy VPN Server is a

Administrative Tasks

The Administrative section available on the 3002 is similar to that of the 3000 concentrators. As a quick overview, you can perform the following administrative tasks on the 3002 from the Administration screens:

- Update the 3002's software image.
- Reboot or shutdown the 3002.
- Test connectivity with ping and traceroute.
- Manage the 3002's files in Flash memory.
- Administer the three accounts on the 3002: admin, config, and monitor (only the first is enabled, by default).
- Change the idle timeout for management sessions and encrypt the 3002's configuration file.
- Import, enroll, and manage certificates.

As you can see from this list, these are similar to the 3000 concentrator's administrative tasks; plus, how you perform them is essentially the same. Therefore, instead of repeating the information covered in Chapter 10, I'll discuss only two topics in this section that are important for management functions of the 3002: how to access the 3002 from its public interface and how to upgrade it.

Accessing the 3002 from its Public Interface

Because the 3002 is a SOHO device, you'll need to be able to access it periodically from the central site for management purposes. This can be accomplished in one of two ways: across an IPsec tunnel to the private interface of the 3002 or to the 3002's public interface. The problem with the former method is that the IPsec tunnel might not be coming up, which means you'll need to perform the latter; and by default, management access to the public interface of the 3002 is disabled.

Only encrypted management access is allowed on the public interface, which makes sense because you don't want to manage the 3002 across a public network in clear text. Both HTTPS (SSL) and SSH are supported and are enabled as followed:

- To enable HTTPS access on the public interface, go to the **Configuration > System > Management Protocols > HTTP/HTTPS** screen on the 3002 and make sure that both the *Enable HTTPS* and the *Enable HTTPS on Public* check boxes are checked: the former should be, by default, but the latter isn't. Click the **Apply** button to accept the changewhen you do this, your current web browser management session is terminated and you'll need to log back in to the 3002.
- To enable SSH access (for CLI access) on the public interface, go to the **Configuration > System > Management Protocols > SSH** screen on the 3002 and make sure that both the *Enable SSH* and the *Enable SSH on Public* check boxes are checked. The former should be, by default, but the latter isn't. Encryption algorithms supported are 3DES, RC-4, and DES; you can choose which one or ones you want to use. Click the **Apply** button to accept the change(s). Unlike HTTPS access, your web browser management session is not terminated upon accepting your changes.

Note

As of 4.1.7 software on the 3002, only SSHv1.5 is supported.

Upgrading the 3002

You can upgrade the 3002 manually on the 3002 or automatically during ISAKMP/IKE Phase 1 based on update parameters you have configured on an Easy VPN Server, such as a VPN 3000

Summary

This chapter showed you the basics of using the Cisco 3002 hardware client to establish IPsec connectivity to a central site Easy VPN Server. Because of its similarity in management to the VPN 3000 concentrators, many administrators prefer these devices over small-end routers or firewalls for SOHO VPN hardware appliances. I prefer to use the 3002 when split tunneling is not being used and I have more than one person at the SOHO site. I've deployed quite a few of these and because of their simplicity and easy maintenance, they are one of my favorite VPN SOHO appliances.

Next up is Part IV, "IOS Routers," where I show you how to use routers for L2L and remote access sessions (gateways and remote access clients).

PREV NEXT

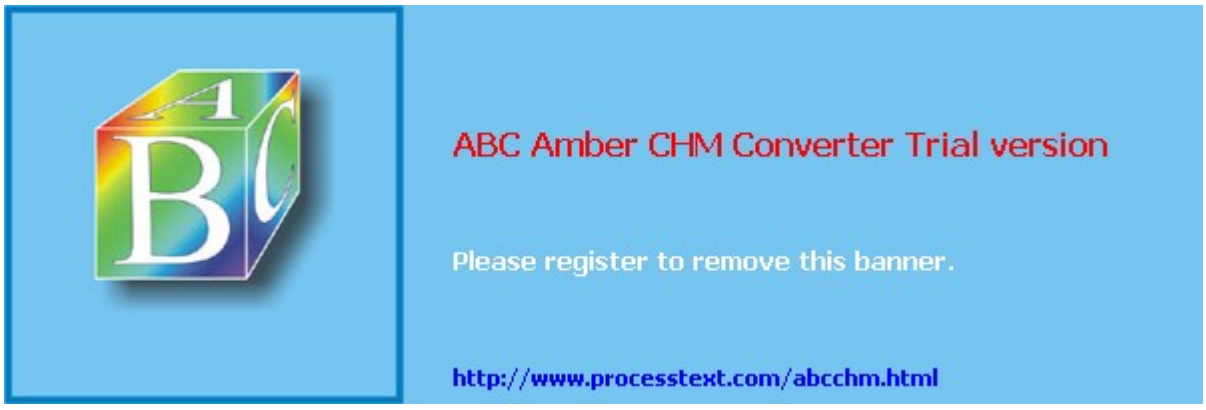
Part IV: IOS Routers

PREV NEXT

Chapter 15. Router Product Information

Part IV will discuss the use of Cisco IOS-based routers to initiate and terminate VPN sessions. Cisco routers are very flexible and can be used for site-to-site (LAN-to-LAN or L2L) sessions, VPN gateways, and remote access clients. I'll focus primarily on how to configure Cisco routers using the IOS CLI interface; however, other GUI-based products, such as Security Device Manager (SDM) and CiscoWorks VMS Router Management Center (MC), can be used to configure VPNs on routers.

In this chapter, however, I'll introduce you to the Cisco router product line that supports VPN capabilities. I'll discuss some typical VPN deployment scenarios used with Cisco routers, and some of the advantages they have over other Cisco VPN products.



Router Deployment Scenarios

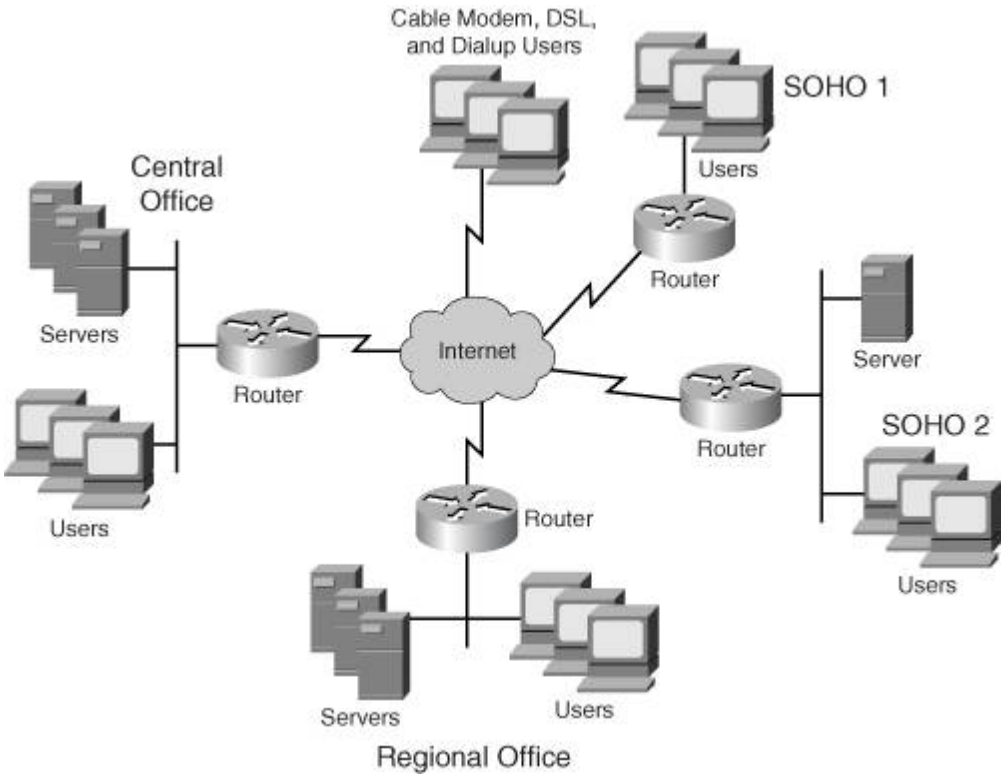
Cisco VPN capabilities have their roots in Cisco IOS-based routers, where VPNs were first introduced in the Cisco router line. Cisco routers support many VPN solutions including IPsec, PPTP, L2TP, and WebVPN (in the newest IOS versions). Because of their flexibility, routers can be used in many different situations. This section will focus on how routers can be used with L2L and remote access VPNs in your network, including the router's special advantages over other Cisco VPN products.

L2L and Remote Access Connections

Cisco routers support L2L and remote access connections. As mentioned in Chapter 8, "Concentrator Remote Access Connections with PPTP, L2TP, and WebVPN," I prefer to use Cisco routers for L2L sessions, but concentrators for remote access sessions. Routers support enhanced routing, quality of service (QoS), and L2L scalability capabilities over Cisco PIX and ASA security appliances and VPN 3000 concentrators. However, concentrators scale better for remote access sessions and are easy to set up.

Figure 15-1 shows an example in which routers are being used to terminate VPN sessions. At the Central Office, a VPN-enabled router is used to terminate an L2L session from the Regional Office, and remote access sessions from the Cable Modem, DSL, and dial-up users, and the SOHO 1 and SOHO 2 sites. Because the number of remote access devices is small, it is more cost-effective to let the Central Office router terminate these versus buying an extra product such as a VPN 3000 series concentrator. Plus, because SOHO 2 has a server that the Central Office needs to access, the session can be set up using either remote access with network extension mode or as an L2L session.

Figure 15-1. Using Routers for VPN Solutions



Special Capabilities of Routers

As I mentioned in Chapter 8, I prefer (and Cisco recommends) to use routers for L2L sessions. I'm constantly asked why routers are better for this role than the VPN 3000 concentrators and the PIX and ASA security appliances. There are four main features the Cisco IOS-based routing products have over Cisco 3000 concentrators and PIX/ASA security appliances when it comes to VPN implementations: data transport, routing scalability, media translation, and quality of service (QoS). The next four sections will discuss these features in more depth.

Router Product Overview

Of the Cisco VPN solutions, their IOS-based routing platforms provide the most options to choose from. Because the Cisco product line for routers changes quite often, I've decided that in this chapter, I'll focus not on any particular router model, but will provide a brief comparison between the various models.

Table 15-1 shows a comparison of the various router models. Please note that some routers support hardware encryption via an add-on encryption card or module, and that there might be a few modules or cards to choose from for a particular router platform (the table lists the fastest encrypting module for each platform).

Table 15-1. Router Product Comparison

Router Model	Location	VPN Sessions	VPN Throughput (in Mbps)
SOHO 90 series[1]	SOHO	8	3DES: 1
830 series	SOHO	10	3DES: 7; AES: 2
850 series	SOHO or small branch office	5	3DES and AES: 8
870 series	SOHO or small branch office	10	3DES and AES: 30
1700 w/VPN module	Small branch office	100	3DES: 15
1841 w/AIM-VPN BP II Plus	Small to medium branch office	800	3DES and AES: 95
2600XM w/AIM-VPN/ BP II module	Medium branch office	800	3DES: 22; AES: 22
2691 w/AIM-VPN/EP II module	Medium branch office	800	3DES: 150; AES: 150
2621, 3640, or 3660 with hardware card	Medium branch office		3DES: 32 Mbps
2800s w/AIM-VPN/EP II-Plus	Enterprise branch office	1,500	3DES and AES: 145
3725 w/AIM-VPN/EP II module	Enterprise branch office	800	3DES: 186; AES: 186
3745 w/AIM-VPN/HP II module	Enterprise branch office	2,000	3DES: 190; AES: 190
3825 w/AIM-VPN/ EP II-Plus	Enterprise branch or regional office	2,000	3DES and AES: 175
3845 w/AIM-VPN/ HP II-Plus	Enterprise branch or regional office	2,500	3DES and AES: 185
7100 w/SM-VAM	Enterprise branch	3,000	3DES: 145; AES:

Summary

This chapter introduced you to Cisco router products that support VPN capabilities. I wanted to give you an overall idea as to where Cisco routers play a role in implementing VPNs, and what kind of router you should use when creating your VPN design. Routers are best used when you need a one-box solution, combining features such as security, QoS, routing, WAN interfaces, and voice support, or if you have many L2L sessions you need to support. Therefore, you might already have a router-based network and want to add VPN functionality to it. You might easily be able to do this with the existing equipment you have, or at a minimal outlay of cost for equipment and software. Plus, out of all of the Cisco VPN offerings, the VPNSM can provide 19 Gbps VPN throughput from a single chassis, which is quite a lot of sessions!

Next up is Chapter 16, "Router ISAKMP/IKE Phase 1 Connectivity," where I discuss how to configure a router to establish a Phase 1 management connection as well as many of its advanced Phase 1 capabilities, like being able to perform the functions of a certificate authority (CA).

Chapter 16. Router ISAKMP/IKE Phase 1 Connectivity

This chapter is the first chapter on configuring routers for VPN implementations, where I'll focus on setting up ISAKMP/IKE Phase 1 to establish a management connection to a remote IPsec peer (site-to-site or remote access). I assume that you have a basic understanding of the Cisco IOS.

In this chapter, I'll discuss the commands that are used to create an ISAKMP/IKE Phase 1 policy for your management connection and the three types of supported device authentication: pre-shared keys, RSA encrypted nonces, and RSA signatures. I'll also briefly discuss how to manage and monitor your management connections. I'll wrap up the chapter discussing a new feature of the IOS, where you can have a Cisco IOS router perform the functions of a certificate authority (CA) or registration authority (RA).

Note

Even though I'll briefly discuss some **show** and **debug** commands in this chapter, I'll reserve most of this discussion for Chapter 19, "Troubleshooting Router Connections."

IPsec Preparation

As with any type of VPN implementation, one of the first steps you'll go through is preparation. This is important for IOS-based routers because the configuration is more complex than setting up a GUI product like the Cisco VPN 3000 concentrators. The next few sections will briefly cover some basic information you'll need to gather, basic configuration tasks you'll perform on your routers, and additional ACL statements you'll need to add to perimeter routers to build IPsec sessions.

Gathering Information

The first step you'll take for your preparation is to gather information that will help you implement your IPsec sessions. If you recall from Chapter 3, "IPsec," two sets of connections are built: a management connection in ISAKMP/IKE Phase 1 and two unidirectional data connections in Phase 2. Each set of connections has protection properties you need to define. For example, with the Phase 1 connection, you'll need to minimally define the following:

- Device authentication type: pre-shared keys, RSA encrypted nonces, or RSA signatures (digital certificates)
- HMAC function: MD5 or SHA-1
- Encryption algorithm: DES, 3DES, or AES
- Diffie-Hellman (DH) group: 1, 2, or 5
- Lifetime of the connection: defaults to 86,400 seconds

For the Phase 2 connection, you'll need to define the following:

- What traffic is to be protected: crypto ACL
- To whom the protected traffic should be forwarded: IPsec peer
- How the traffic is to be protected: transform set
- What traffic should be protected to which peer: crypto map entry
- If Perfect Forward Secrecy (PFS) should be used to share new keying information for the data connections: DH groups 1, 2, or 5
- How long the data connection is valid: defaults to 3,600 seconds

Of course, the above items are the minimum items you'll need to identify and configure. In this chapter, though, I'll focus only on the first set of bullet points.

Allowing IPsec Traffic

One of the tasks you'll need to perform is to allow IPsec traffic into your network on your perimeter routers and firewalls. For a perimeter router, at a minimum you'll need to add the following entries into your ACL (assuming it's a numbered ACL and your perimeter router is using ACLs for filtering):

```
Router(config)# access-list ACL_# permit udp host remote_peer_IP
                  host local_router's_IP eq 500
Router(config)# access-list ACL_# permit ahp host remote_peer_IP
                  host local_router's_IP
Router(config)# access-list ACL_# permit esp host remote_peer_IP
                  host local_router's_IP
Router(config)# access-list ACL_# permit udp host remote_peer_IP
                  host local_router's_IP eq 4500
```

The first ACL statement allows the management connection to the router. With your ACL statements for L2L sessions, you want to be as specific as possible with the remote peer's IP address and the local router's IP address; with remote access sessions you might not know

ISAKMP/IKE Phase 1 Policies

One of the first steps you'll take in setting up IPsecL2L or remote accessis to define your ISAKMP policies for your ISAKMP/IKE Phase 1 management connection. The following subsections will discuss how to create your policies and the following section will define how to configure the device authentication information you've chosen for your Phase 1 policies.

Enabling ISAKMP

If you have a router with the IPsec feature, ISAKMP/IKE is enabled by default. To enable or disable it, use the following command:

```
Router(config)# [no] crypto isakmp enable
```

You need to disable ISAKMP/IKE only if the remote peers do not support it, in which case you'll have to configure *all* parameters and keys for the data connection manually instead of having ISAKMP/IKE negotiate the parameters and create keying material dynamically; however, this is rarely done.

Creating Policies

An ISAKMP/IKE policy defines how the management connection is to be created, authenticated, and protected. You can have more than one policy on your router. You might need to do this if your router has multiple peers and each peer has different abilities or policy configurations. If you own the entire network and all the routers are Cisco routers, typically you would have a single policy that would encompass any management connection to any of your peering Cisco routers.

A single ISAKMP/IKE policy contains the following parameters: prioritization or sequence number, encryption algorithm, hashing function, authentication method, DH key group, and connection lifetime. Here are the commands to create a policy for the management connection:

```
Router(config)# crypto isakmp policy priority
Router(config-isakmp)# encryption {des | 3des | aes}
Router(config-isakmp)# hash {sha | md5}
Router(config-isakmp)# authentication {rsa-sig | rsa-encr | pre-share}
Router(config-isakmp)# group {1 | 2 | 5}
Router(config-isakmp)# lifetime seconds
Router(config-isakmp)# exit
```

The **crypto isakmp policy** command creates a unique ISAKMP/IKE management connection policy on the router, where each policy requires a separate number. Numbers can range between 110,000. Executing this command takes you to a subcommand mode where you enter the configuration for the policy. The **encryption** command specifies which encryption algorithm to use; the **hash** command specifies the HMAC function to use; the **authentication** command specifies the method to use for device authentication (you'll also need to configure the actual authentication information that you have decided to use, discussed in the "ISAKMP/IKE Phase 1 Device Authentication" section later); the **group** command specifies the DH key group to use; and the **lifetime** command specifies the lifetime of the management connection. If you don't specify a particular parameter in a policy, it has a default value, as follows:

- Encryption algorithm: DES
- HMAC function: SHA-1
- Authentication method: RSA signatures (certificates)
- DH group: 1
- Lifetime: 86,400 seconds

Likewise, a default, unnumbered ISAKMP/IKE policy exists on the router with the above configured values; so if these are sufficient, you don't need to configure an ISAKMP/IKE policy

ISAKMP/IKE Phase 1 Device Authentication

Because the configuration of device authentication can be complex, at least when it comes to RSA encrypted nonces and especially digital certificates, I've separated the configuration process for authentication from the ISAKMP/IKE Phase 1 policy configuration and will cover it in its own section.

Note that Cisco routers support three methods of authenticating IPsec devices (peers): Pre-shared keys, RSA encrypted nonces, and RSA signatures (digital certificates). The following sections will discuss the configuration of these authentication methods.

ISAKMP/IKE Identity Type

Before I discuss the three ways of configuring device authentication, I first need to discuss the use of a router's identity type. The ISAKMP/IKE identity type specifies how each peer sends its identity to the remote peer; it will send either its IP address or its host name. This is used only when pre-shared (symmetric) keys or RSA encrypted nonces (asymmetric pre-shared keys) are used. This information is used by the remote peer to determine what pre-shared key information should be used to perform the device authentication.

The default is to have the router send its IP address. This works well if the local device has only one IP address that it will always use to initiate or terminate IPsec sessions. However, in certain cases this will cause problems if the local device can use more than one IP address (thus, more than one interface) to communicate to the remote device. In this case, I recommend that you use the router's host name for the identity type.

To specify the identity type, use the following command:

```
Router(config)# crypto isakmp identity {address | hostname}
```

address is the default type; so if your router will always use the same IP address to reach the remote peer, you don't need to configure this command. I recommend that if you use the **hostname** parameter, you should configure a static resolution table on your router for the name to the multiple IP addresses associated with the peer; if you don't do this, and DNS resolution fails, authentication also will fail and no IPsec tunnel will be built. To build a static DNS table, use the following command:

```
Router(config)# ip host hostname address1 [address2...address8]
```

You can list up to eight IP addresses per host name.

Note

Please note that the **crypto isakmp identity** command is a global command; you can't change the identity type on a peer-by-peer basis. Therefore, if you set it to **hostname** on one device, all of the other IPsec peer devices must be configured the same.

Pre-Shared Keys

Of the three methods of device authentication, configuring what Cisco calls "pre-shared keys" is the easiest. With pre-shared keys, the same key (symmetric) is used on both peers to perform the device authentication. Pre-shared keys, symmetric or asymmetric, commonly are used when you have a small number of devices with which you need to establish IPsec tunnels. When you add more and more IPsec devices to the network, however, pre-shared keys scale poorly and thus digital certificates are used to solve the device authentication scalability issues. The one advantage that pre-shared keys have over certificates, especially symmetric pre-shared keys, is that they are simple to set up on the two IPsec peers.

Configuring Pre-shared Keys

To configure a symmetric pre-shared key for device authentication on a router, use one of the following two commands, based on the identity type used on the router:

```
Router(config)# crypto isakmp {0 | 6} key keystring
```

Monitoring and Managing Management Connections

In the next two sections I'll discuss some **show**, **clear**, and **debug** commands you can use to view and manage your ISAKMP/IKE Phase 1 management connections. This chapter introduces these commands and Chapter 19, "Troubleshooting Router Connections," will provide an in-depth coverage of these commands as they relate to troubleshooting IPsec sessions.

Viewing ISAKMP/IKE Phase 1 Connections

When a management connection is being built, it will go through various states. The current state of this connection can be seen with this command:

```
Router# show crypto isakmp sa [detail]
```

Example 16-26 illustrates the use of this command. In this example, only one management connection exists. The *state* column indicates what state the connection is in. Table 16-1 explains the various states a connection can be in.

Example 16-26. Viewing Management Connections

```
r3640# show crypto isakmp sa
dst             src             state          conn-id slot status
192.1.1.40      192.1.1.20      QM_IDLE              1    0 ACTIVE
```

Table 16-1. Management Connection States

State	Explanation
MM_NO_STATE	When using main mode, the ISAKMP SA is in an infancy state and has not completed; you'll typically see this appear when a management connection fails to establish.
MM_SA_SETUP	When using main mode, the policy parameters have been negotiated between the peers successfully.
MM_KEY_EXCH	When using main mode, the peers have performed DH and created a shared secret key, but device authentication hasn't occurred yet.
MM_KEY_AUTH	When using main mode, the peers have passed authentication and will transition to a QM_IDLE state.
AG_NO_STATE	When using aggressive mode, the ISAKMP SA is in an infancy state and has not completed; you'll typically see this appear when a management connection fails to establish.
AG_INIT_EXCH	The first exchange in aggressive mode has completed, but device authentication hasn't been performed yet.
AG_AUTH	When using aggressive mode, the peers have passed authentication and will transition to a QM_IDLE state.
QM_IDLE	The management connection has been built and can be used during ISAKMP/IKE Phase 2 to build data connections. This is commonly referred to as *quiescent* mode.

You can view more details about the management connections by adding the **detail** parameter to the **show crypto isakmp sa** command, as illustrated in Example 16-27. Here you can see information like the type of encryption algorithm used ("aes"), the HMAC function

Routers as Certificate Authorities

As of IOS 12.3(4)T, Cisco routers can perform the function of a CA; RA functionality was added in a later IOS release. As a CA, routers can accept certificate requests using SCEP (which means that they have to run an HTTP server) and manual enrollment with cut-and-paste of the PKCS #10 information.

The CA server feature was added mostly for small shops that wanted to use an existing router for certificate services instead of purchasing a stand-alone product. However, the Cisco CA server feature does have limitations; it isn't a full-blown CA product. Here are some of its restrictions:

- When acting as an RA, the CA must be an IOS router.
- Only a central design with one CA is supported.
- As a CA, time services (NTP) must be running on the router or the router must have a hardware clock; otherwise, the CA process will not start on the router.

Like getting a certificate on a router, there are many steps involved in setting up and managing a router acting as a CA:

1. Generating and exporting an RSA key information (optional)
2. Enabling the CA (required)
3. Defining additional CA parameters (optional)
4. Handling enrollment requests (required)
5. Revoking identity certificates (optional)
6. Configuring a server to run in RA mode (optional)
7. Backing up a CA (optional, but highly recommended)
8. Restoring a CA (optional)
9. Removing CA services (optional)

Of the nine steps, note that only Steps 2 and 4 are required. The following sections will discuss these steps, and will show a simple example of a router configured as a CA.

Battle of the CAs

One of the questions I'm constantly asked as a consultant is which CA product a company should use. For an in-house CA, I always sided with the CA product that comes bundled with Microsoft 2000 and 2003 Server/Advanced Server products. SCEP is an add-on from the Windows resource kit. I always liked using this product because it was simple to set up and manage and it came with the operating system. However, Microsoft's application has a downside. If you experience problems obtaining certificates from it, troubleshooting these problems from within Windows is not an easy proposition. When it works, it works great, and when it doesn't work, you want to pull your hair out in frustration in finding the problem.

When Cisco introduced the CA functionality on a Cisco router, I immediately downloaded it on one of my 3640s and played around with the code. I really liked the flexibility in its configuration, and the ability to see detailed debugging information with interactions between other certificate-requesting devices. Even so, the Cisco CA solution has a problem, too. If you want to handle lots of certificates, the certificates will have to be stored on an external box; this slows down the validation part of certificates when a peer requests your certificate from

Summary

This chapter showed you the basics of setting up ISAKMP/IKE Phase 1. The defining of the policy statements on a router is straightforward. With three types of device authentication to choose from, pre-shared keys is the simplest to configure but scales the least; certificates are the hardest to implement, but scale the best. And with the ability of a Cisco router to function as a CA, you can deploy certificate services easily using existing equipment.

Next up is Chapter 17, "Router Site-to-Site Connections," where I show you how to configure your router to establish various types of site-to-site sessions with remote peers, covering topics such as static and dynamic crypto maps, the Tunnel Endpoint Discovery (TED) protocol, dynamic multipoint VPNs (DMVPNs), and many others.

PREV NEXT

Chapter 17. Router Site-to-Site Connections

Where the last chapter focused on ISAKMP/IKE Phase 1, this chapter will focus on ISAKMP/IKE Phase 2 and what you have to do on a router to set up IPsec site-to-site or LAN-to-LAN (L2L) sessions. As mentioned in Chapter 15, "Router Product Information," routers typically are the best solution for L2L sessions because they support advanced QoS, routing, and L2L scalability features. Some of the topics I discuss here are applicable to remote access connections for an Easy VPN Server, such as static and dynamic crypto maps, address translation, and many others, which you'll see in the next chapter as well. Throughout the chapter I'll go through many examples that illustrate the different types of configurations I discuss. However, I'll reserve an in-depth discussion of troubleshooting commands, such as **show** and **debug**, for Chapter 19, "Troubleshooting Router Connections."

ISAKMP/IKE Phase 2 Configuration

Once ISAKMP/IKE Phase 1 completes (negotiates the Phase 1 policies, performs DH, and authenticates the peer), the management connection is established. The management connection is then used to build the two unidirectional data connections during ISAKMP/ IKE Phase 2. In its simplest form, there are three components that need to be configured for L2L sessions for ISAKMP/IKE Phase 2:

- Define the traffic that needs to be protected (crypto ACL).
- Define how that traffic is to be protected (transform set).
- Define to whom the traffic should be forwarded (crypto map).

If the remote L2L peer acquires its address dynamically, and you want to allow the remote peer to build an L2L session to your router, you'll have to build a dynamic crypto map with an entry for this peer. If you're using certificates for device authentication, you can use a new feature in the IOS called distinguished named-based crypto maps. This feature allows you to have more control over what certificate you'll accept from a particular peer. The following section will discuss the three bullets above and these two additional, yet optional, configuration tasks.

Defining Protected Traffic: Crypto ACLs

One method of defining what traffic needs to be protected is to create a crypto ACL. A crypto ACL is basically an ACL with **permit** or **deny** statements in it, where a **permit** statement specifies the traffic is to be protected and a **deny** statement specifies the traffic doesn't need to be protected. Named or numbered ACLs can be used. The crypto ACL created on one peer should be mirrored on the other peer; this ensures that traffic that is to be protected locally by your router is processed correctly by the remote peer.

Tip

In most instances, if you don't mirror crypto ACLs on the two peers, the ISAKMP/IKE Phase 2 connection will fail.

For example, if I want to protect traffic between two networks192.168.1.0/24 connected to RouterA and 192.168.2.0/24 connected to RouterBI would use the configuration shown in Example 17-1 (I assume you are familiar with ACLs on IOS routers). The names or numbers of the ACLs on the two peers don't need to match; however, I've found that when a router is connected to many different peers, possibly in a partially or fully meshed environment, using the same ACL name for each peer-to-peer session makes it easier to manage. Notice that in the two ACL **permit** statements, the addressing information is mirrored (reversed).

Example 17-1. Mirroring Crypto ACLs

```
! RouterA's configuration
RouterA(config)# ip access-list extended mirrored
RouterA(config-ext-nacl)# permit 192.168.1.0 0.0.0.255
                          192.168.2.0 0.0.0.255
! RouterB's configuration
RouterB(config)# ip access-list extended mirrored
RouterB(config-ext-nacl)# permit 192.168.2.0 0.0.0.255
                          192.168.1.0 0.0.0.255
```

Cisco highly discourages the use of the keyword **any** in a crypto ACL, because this can create connectivity problems. For example, if you want multicast traffic to enter the router's external interface where IPsec is applied, this will cause the router to drop the traffic unless it is protected by IPsec. This is also true of other kinds of traffic, like ICMP messages. If you do use the keyword **any** in a **permit** statement, be sure to precede the statement with specific **deny** statements for traffic that shouldn't be protected.

Tip

Viewing and Managing Connections

Now that I've discussed how to build some basic types of L2L connections, I'll focus on how to view and manage these connections using some basic **show, clear**, and **debug** commands in the following two sections.

Viewing IPsec Data SAs

To view your ISAKMP/IKE Phase 2 data SAs, use the following **show** command:

```
Router# show crypto ipsec sa [map crypto_map_name | address |
                         identity | interface interface_type_and_#]
                         [detail]
```

If you don't enter any optional parameters, all data SAs are displayed. The **address** parameter sorts the SAs based on the peers' IP addresses. The **identity** parameter displays a summarized view. You can qualify what data SAs are displayed with additional parameters: the **map** parameter allows you to restrict the SAs displayed to the crypto map specified, whereas the **interface** parameter restricts the displayed SAs to those terminated on the specified interface. The **detail** parameter also will display send and receive error counter statistics.

Example 17-11 illustrates the use of this command. In this example, the *local ident* and *remote ident* specifies the traffic to be protected based on the crypto ACL. The *current peer* specifies the remote peer's address. The first two *#pkts* lines specify the number of IPsec packets encapsulated and deencapsulated, encrypted and decrypted, hashed and verified. Below this are the inbound and outbound SAs. Because only ESP is used for the data connections, only two SAs are seen (*inbound esp sas* and *outbound esp sas*). In both cases, the SAs are protected by AES-128 and MD5. AH and PCP (compression) are not used and thus no SAs for these exist.

Example 17-11. Using the *show crypto ipsec sa* Command

```
r3640# show crypto ipsec sa
interface: Ethernet0/0
    Crypto map tag: mymap, local addr 192.1.1.40
   protected vrf: (none)
   local  ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (192.168.2.0/255.255.255.0/0/0)
   current_peer 192.1.1.20 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 5, #pkts encrypt: 5, #pkts digest: 5
    #pkts decaps: 5, #pkts decrypt: 5, #pkts verify: 5
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 192.1.1.40, remote crypto endpt.: 192.1.1.20
     path mtu 1500, ip mtu 1500
     current outbound spi: 0xED39B285(3979981445)

     inbound esp sas:
      spi: 0x5B5A20FC(1532633340)
        transform: esp-aes esp-md5-hmac ,
        in use settings ={Tunnel, }
        conn id: 3001, flow_id: SW:1, crypto map: mymap
        sa timing: remaining key lifetime (k/sec): (4458063/3572)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE
     inbound ah sas:
     inbound pcp sas:
     outbound esp sas:
      spi: 0xED39B285(3979981445)
```

Issues with Site-to-Site Connections

As you can see from the previous section, setting up L2L sessions on a router is a little more difficult than setting them up on a concentrator; but the configuration process is still fairly straightforward. However, there can be a handful of issues that you'll need to deal with when setting up and using L2L sessions, including:

- Migration to an IPsec-based design
- Filtering of IPsec traffic
- Address translation usage
- Non-unicast traffic
- Configuration simplification
- IPsec Redundancy
- Scalability

The following sections will cover each of these issues and explain solutions you can use to help you with your IPsec implementation.

Migration to an IPsec-Based Design

One issue you might have when implementing a large number of IPsec L2L sessions is that data won't transfer if only one peer is configured with IPsec, or if IPsec is misconfigured. To alleviate these problems, Cisco introduced the IPsec Passive Mode feature in IOS 12.2(13)T. IPsec Passive Mode allows a router to accept both encrypted and unencrypted traffic that matches a crypto ACL. Routers will attempt to negotiate a protected session, if specified. If they are successful, they'll protect the traffic before sending it; otherwise, they'll forward the traffic unencrypted.

IPsec Passive Mode Process

When a router configured for IPsec Passive Mode receives a packet that needs to be sent to a peer using IPsec to protect it, the router will try to establish an SA to the peer. The router waits 10 seconds for a tunnel to be established. Within the 10 seconds, the router will drop any packets that need to be forwarded to the remote peer if it cannot buffer them up while waiting for an SA to be established; after 10 seconds, if no SA can be established, packets will be forwarded to the remote peer in clear text. When the latter happens, the router will generate a warning message indicating that IPsec Passive Mode is being used:

```
Unencrypted traffic is sent to X.X.X.X because crypto optional
    is configured
```

X.X.X.X is the IP address of the remote peer. If an SA can be established, the SA is used to protect the packet before forwarding it to the remote peer. If the router receives unprotected traffic from the remote peer, the following message is displayed:

```
Unencrypted traffic is received from X.X.X.X because crypto optional
    is configured
```

Both messages are rate-limited to ensure that they don't waste resources on the router; by default, they are generated once a minute no matter how many instances of the two processes have occurred.

Tip

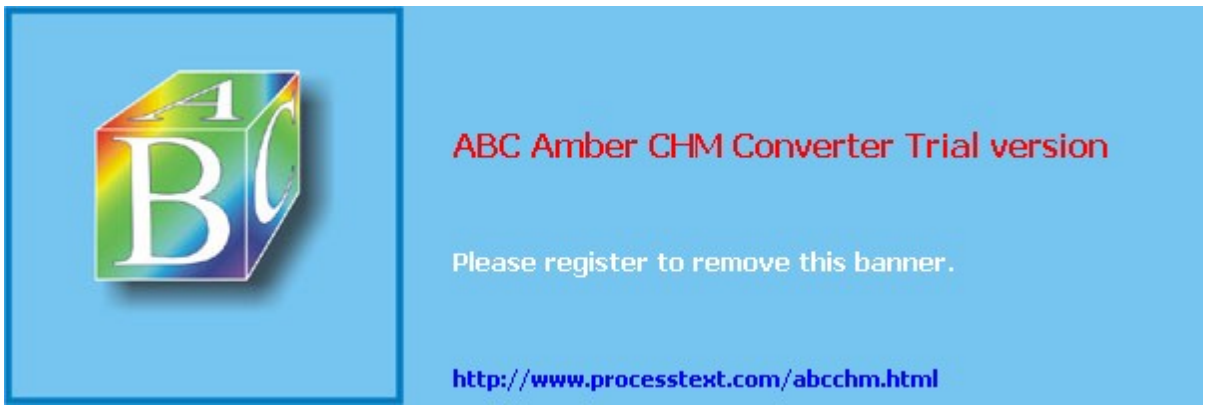
Look for the above messages in your router's log output. These messages tell you of peers that either haven't been configured or have been configured incorrectly.

IPsec Passive Mode Configuration

Summary

This chapter showed you the basics of setting up L2L sessions on routers. I've focused primarily on the ISAKMP/IKE Phase 2 components for building these connections. The order of the chapter has led you through the features as they became available in the Cisco IOS: static crypto maps, dynamic crypto maps, TED, GRE across IPsec, and DMVPN.

Next up is Chapter 18, "Router Remote Access Connections," where I show you how to configure your router to be an Easy VPN Server or Remote (Client).



Chapter 18. Router Remote Access Connections

In this chapter I'll focus on using a Cisco router for remote access sessions. I'll discuss how you can use a router as a VPN gateway (Easy VPN Server), terminating remote access sessions from client devices, like the Cisco VPN Client software and VPN 3002 hardware clients. Because routers commonly are used for LAN-to-LAN (L2L) sessions, they're typically not used as Easy VPN Servers; however, for a small number of remote access clients, it is common to use an existing router for this function instead of purchasing a stand-alone remote access device like a VPN 3000 concentrator. I'll discuss how to terminate both L2L and remote access sessions on the same router.

I'll also discuss how you can use a router as an Easy VPN Server and how to set up a small-end router as a remote access client, called an Easy VPN Remote. Routers commonly are used as Remotes for small office, home office (SOHO) networks when you need complex QoS policies or have Internet connections that are non-Ethernet-based, such as ISDN, xDSL, or serial. At the end of the chapter I'll discuss a new remote access feature supported by Cisco Routers: WebVPN. Starting in late releases of 12.3T, a Cisco router can be used to terminate WebVPN sessions.

Note

Because of page constraints, I'll focus only on IPsec and WebVPN remote access VPNs in this chapter; PPTP and L2TP/IPsec are not covered.

Easy VPN Server

Easy VPN is a Cisco feature that allows you to deploy IPsec remote access devices easily using Cisco VPN Client software; their 800, 900, and 1700 series routers; the VPN 3002 hardware client; and the 501 and 506E PIX security appliances. Policies are defined on the Easy VPN Server and pushed down to the Easy VPN Remotes during IKE Mode Config.

However, as you'll see in this section, the configuration of the Easy VPN Server function from the router's CLI is not a simple endeavor, which is why I prefer using the VPN 3000 concentrators over Cisco routers or PIXs for Easy VPN Servers. Given that the configuration of an Easy VPN Server is more difficult than a concentrator, you might wonder why you should even bother with a router. Well, if you already have a perimeter router with extra capacity and need to support only a small number of remote access users, it doesn't make any sense to spend money to buy a VPN 3000 concentrator: instead, just use the existing router to perform this function.

Note

Please note that the Cisco Security Device Manager (SDM) software makes it easy to set up a router as an Easy VPN Server or Remote, hiding the commands and their syntax behind a user-friendly GUI interface. The discussion of SDM is beyond the scope of this book.

The Easy VPN Server function was introduced in IOS 12.2(8)T, and Cisco has added features to it constantly to match support with other Cisco Easy VPN Server products. Routers, when functioning as Easy VPN Servers, do *not* support the following options:

- RSA encrypted nonces or DSS certificates for device authentication
- DH group 1 keys
- AH for data encapsulation
- Transport mode for data connections
- Manual keying for data connections (ISAKMP/IKE is required)
- PFS for data connections: as of IOS 12.3(4)T, PFS support has been added
- IPsec over TCP and IPsec over UDP: standards-based NAT-T, however, is supported

To help you set up a router as an Easy VPN Server, the following sections cover these topics:

- Easy VPN Server Configuration
- VPN Group Monitoring
- Easy VPN Server Configuration Example

Easy VPN Server Configuration

Many of the tasks that are configured for L2L sessions, as discussed in Chapter 17, "Router Site-to-Site Connections," also are configured for IPsec remote access sessions, including ISAKMP/IKE Phase 1 policies, transform sets, dynamic crypto maps, and static crypto maps. Because I've already discussed these commands in Chapters 16 and 17, I'll only focus on the Easy VPN Server commands in the following sections. After this, I'll put all of these components together (from this and the previous two chapters) in an Easy VPN Server example.

Defining AAA

AAA (authentication, authorization, and accounting) is used to restrict traffic to or through a router. Easy VPN uses AAA to implement group policies and authorization and also for XAUTH (user) authentication. This section will focus on only the basics of AAA in regard to its use with an Easy VPN Server: for more detailed information on using AAA on a Cisco router, read my book entitled *Cisco Router Firewall Security* published by Cisco Press (ISBN 1587051753).

Easy VPN Remote

Besides supporting the Easy VPN Server function, certain routers also can be Easy VPN Remotes. These routers include the 800, ubr900, and 1700 series routers. This was introduced in IOS 12.2(4)YA and 12.2(13)T. As you saw in the last section, setting up an Easy VPN Server on a router from the CLI is not the simplest process in the world. Because of this complication and because administrators at remote offices might not be very Cisco-savvy at configuring IPsec sessions, Cisco has simplified the configuration for Easy VPN Remote devices. In other words, there are very few commands you need to configure to set up a router as an Easy VPN Remote.

Cisco accomplishes this by using the same process used with the Cisco VPN Client software: hiding all of the IPsec details from the user. Policies are defined on an Easy VPN Server and pushed down to the Remote during IKE Mode Config. These advantages allow you to deploy a large number of Remotes quickly and easily.

Note

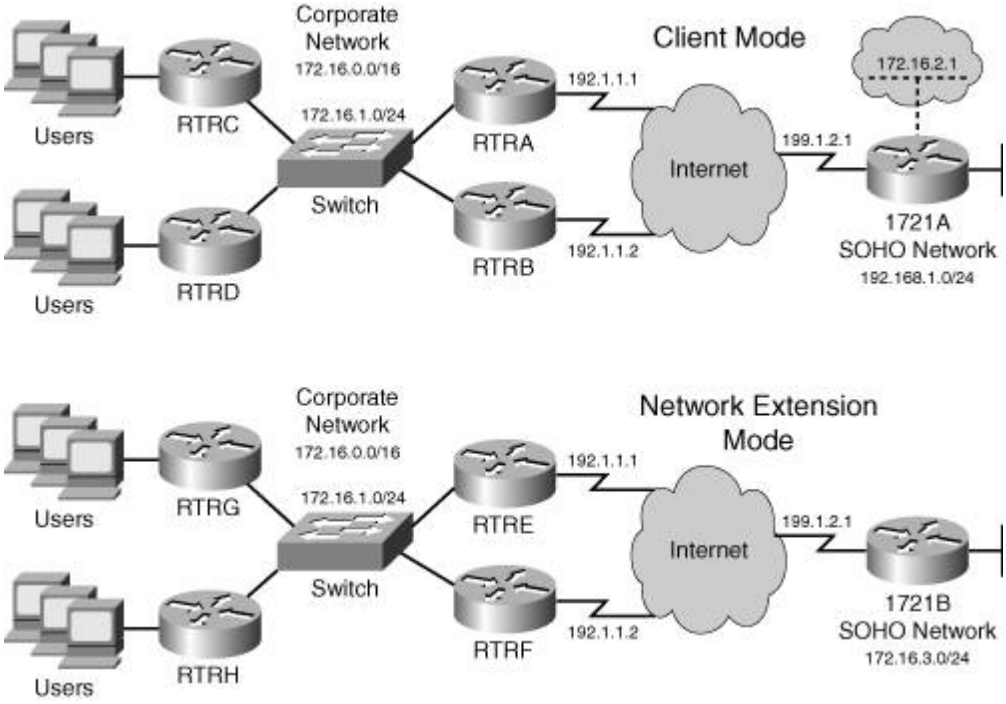
Even though Cisco officially supports Remote functionality on the 800, ubr900, and 1700 series routers, the Remote commands work on other routers. I've successfully set up 3620 and 3640 routers as Remotes. However, don't expect any help from the Cisco TAC if you have a problem with an unsupported Remote router client.

Easy VPN Remote Connection Modes

Cisco Easy VPN Remote routers are more similar to Cisco 3002 hardware clients. Both support client and network extension modes, as shown in Figure 18-2. If you recall from Chapter 3, in client mode, the Easy VPN Remote is assigned a single internal IP address; all devices behind the Remote have PAT performed on them by the Remote to send their traffic across the IPsec tunnel.

Figure 18-2. Easy VPN Remote with Client and Network Extension Modes

[View full size image]



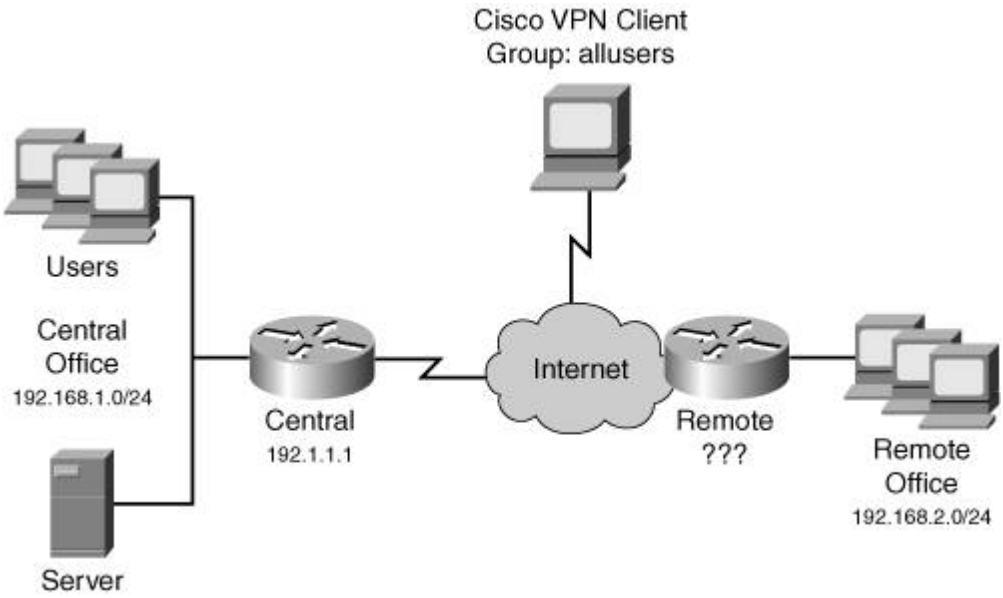
The main limitation of client mode is that devices behind the Easy VPN Server can't initiate connections to devices behind the client-mode Remote; in this case, you would use network extension mode. Because of the word "easy" in Easy VPN, you don't have to set up NAT or PAT on the Remote device. Cisco Easy VPN software will do this automatically. The only requirement is that the Remote act as a DHCP server for its internal devices. Cisco does make

IPsec Remote Access and L2L Sessions on the Same Router

Figure 18-3 shows an example network that has both remote access and L2L sessions. In this example, the remote office network is acquiring its address dynamically, via DHCP, from its connected ISP, which also is true of the remote access clients. Because the central office router doesn't know the remote office router's IP address, you would have to configure the pre-shared key for the router as 0.0.0.0 0.0.0.0 with no XAUTH. However, doing this would cause XAUTH to not work for the remote access clients. One solution would be to use certificates instead of pre-shared keys; for small networks, though, this might not be cost-effective or practical.

Figure 18-3. Remote Access and L2L Connections

[View full size image]



With IOS 12.2(15)T, however, you now can use ISAKMP/IKE profiles to match on other properties to determine how to do authentication during ISAKMP/IKE Phase 1, such as a client's group name, a peer's IP address, a fully qualified domain name, and other information than just the IP address of the peer. You also can define other properties to use for the ISAKMP/IKE Phase 1 connection.

Caution

Because of a bug in 12.2(15)T, the wildcarding of pre-shared keys doesn't work. This has been fixed in 12.3(3) and 12.3(2)T. Therefore, to use this feature, be sure you are running one of these two IOS versions or a later version.

The following sections will cover these topics:

- Central Office Router Configuration
- Remote Access and L2L Example Configuration

Central Office Router Configuration

When a central office router needs to terminate remote access and L2L sessions with pre-shared keys, and one or more of the L2L peers acquires its address dynamically, you can create an ISAKMP/IKE profile and use it with peers that match the components of the profile.

There are a few components you need to configure on your router to allow pre-shared keys for this particular situation above and beyond your normal configuration for L2L and remote access clients:

WebVPN

In 12.3(14)T Cisco introduced SSL VPN functionality, called WebVPN, on Cisco IOS-based routers. This feature allows you to set up a router to terminate user-based SSL VPNs. Cisco took much of the WebVPN technology that exists in the VPN 3000 concentrators and ported it to the IOS routers. In other words, an IOS-based router, with WebVPN implemented, is by and large a secure web proxy.

The user desktop requirements are essentially the same as those for WebVPN sessions terminating on concentrators:

- Supported SSL-enabled browser: Internet Explorer, Netscape, Mozilla, or FireFox
- Sun Microsystems Java Runtime (for the Port Forwarding, or thin client, feature only)
- Optionally a supported e-mail client: Microsoft Outlook, Netscape Mail, or Eudora

However, Cisco VPN 3000 concentrators support more SSL VPN features than Cisco routers; for example, WebVPN-enabled routers have the following limitations, and others, when compared to the concentrators:

- Only SSLv3 is supported: TLS is not supported.
- The Cisco Secure Desktop and Cisco SSL VPN Client are not yet supported; therefore, a WebVPN router can only terminate clientless and thin client connections.
- URLs referenced by Macromedia Flash are not supported for secure retrieval.

What the user sees when accessing the WebVPN deviceconcentrator, ASA, or router is the same, requiring no additional user training. Therefore, you might start deploying the Cisco WebVPN solution using routers, and then migrate to the use of VPN 3000 concentrators, which are more scalable when it comes to WebVPN services, and also support more SSL VPN features. The transition between using a router and a concentrator to terminate WebVPN services is transparent to the user because the user still sees the same information when accessing the WebVPN device, authenticating via a user account, and then using services via the WebVPN web home page and toolbar (this is true with the clientless and thin client implementation).

I discussed how to use WebVPN from the user-side in Chapter 8, "Concentrator Remote Access Connections with PPTP, L2TP, and WebVPN." Therefore, this chapter focuses only on how to set up a router as a WebVPN server, not how to use WebVPN services as a user. See Chapter 8 for using WebVPN services as a user.

WebVPN Setup

The setup and maintenance/monitoring of WebVPN connections on IOS routers is not difficult. The following steps summarize this process:

Step 1. Configuring Prerequisites: AAA, DNS, and Certificates (required)

Step 2. Configuring WebVPN (required)

Step 3. Creating URL and Port Forwarding Entries for the Home Page (optional, but recommended)

Step 4. Maintaining, monitoring, and troubleshooting WebVPN connections (optional)

In the following sections, I'll discuss the commands necessary to perform the above steps.

Step 1: Configuring Prerequisites

Before you begin configuring WebVPN on your router, you'll need to perform some prerequisite configuration tasks that are necessary for WebVPN. These include:

- Setting up AAA for authenticating WebVPN users

Summary

This chapter showed you the basics of setting up a router for remote access services, including Easy VPN Server, Easy VPN Remote, and WebVPN. Normally, routers are not used as Easy VPN Servers, because the configuration is complex and the VPN 3000 concentrators and PIX and ASA security appliances support more enhanced remote access capabilities. However, there are many cases where routers are used as Easy VPN Remote devices, especially in cases where non-Ethernet Internet connections or enhanced QoS are required. The last part of the chapter dealt with a new feature on routers: WebVPN. With this feature, you can set up very basic SSL VPNs to Cisco routers.

Next up is Chapter 19, "Troubleshooting Router Connections," where I show you how to use **show** and **debug** commands to troubleshoot router VPN sessions.

Chapter 19. Troubleshooting Router Connections

This chapter will focus on how to troubleshoot IPsec sessions on Cisco routers. I've broken the chapter into two areas on troubleshooting: ISAKMP/IKE Phase 1 and Phase 2 issues. I'll show you how ISAKMP/IKE Phase 1 and 2 connections are built using **debug** commands and what to look for when there is a problem with either of these phases. I'll also discuss a new feature in the IOS called VPN Monitoring, which allows you to determine problems with IPsec sessions more easily. The last part of the chapter will deal with one main issue with any type of VPN implementation: fragmentation.

Note

This chapter by no means covers all possible problems you'll experience with IPsec sessions on Cisco routers. However, I hope to provide you with the necessary background so that troubleshooting IPsec sessions is a simpler process. I could easily talk about troubleshooting IPsec sessions on routers for over 200 pages, but because of all of the other topics in this book, I'll keep my discussion to a reasonable number of pages. Plus, the solutions I discuss here, such as how to troubleshoot fragmentations problems, can be applied easily to other Cisco VPN products.

ISAKMP/IKE Phase 1 Connections

In the first part of this chapter I'll focus on troubleshooting ISAKMP/IKE Phase 1 connections. If you recall from Chapter 3, "IPsec," the management connections built during Phase 1 are used to pass IPsec management traffic; no data traverses these connections. These connections are important, however, because they are used to build the data connections for Phase 2.

I've broken this part of the chapter into three areas. In the first part, I discuss the commands used to troubleshoot ISAKMP/IKE Phase 1 connections; the two sections following this are specific to L2L and remote access implementations, respectively.

Overview of the Phase 1 Commands

You can use the following router commands to troubleshoot ISAKMP/IKE Phase 1 connections:

- **show crypto isakmp sa** Displays the status of all management connections.
- **debug crypto isakmp** Displays the steps taken to build a management connection and data connections via the management connection.
- **debug crypto pki** {**messages**| **transactions**}: Displays the interaction between the router and CA for certificate enrollment and authentication functions.
- **debug crypto engine** Displays events related to encrypting and decrypting packets, and applies to both Phase 1 and Phase 2 connections.
- **clear crypto isakmp** [*SA_ID_#*] Deletes all of the specified management SAs.

The show crypto isakmp sa Command

Example 19-1 illustrates the use of the **show crypto isakmp sa** command. I discussed this command in Chapter 16, "Router ISAKMP/IKE Phase 1 Connectivity." Table 16-1 in that chapter explained the states. If you recall, QM_IDLE indicates the successful setup of the connection to the associated peer. If you're seeing MM_NO_STATE or AG_NO_STATE, this indicates that there is a problem with the initial setup of the connection.

Example 19-1. The *show crypto isakmp sa* Command

```
spoke1# show crypto isakmp sa
dst             src             state          conn-id slot
192.1.1.40      192.1.1.42      QM_IDLE              2    0
```

The two most common problems that might cause a management connection from being set up are:

- You forgot to activate the crypto map or profile on the remote peer router's interface.
- There is no matching ISAKMP/IKE Phase 1 policy on the remote peer.

If you see a state of MM_KEY_EXCH or AG_INIT_EXCH, probably device authentication failed. For pre-shared keys or RSA encrypted nonces, make sure you've configured the pre-shared keys correctly. For certificates, make sure:

- The certificates haven't expired.
- The date and time are correct on the two peers.
- The certificates haven't been revoked.

Tip

You can use the **debug crypto isakmp** command for more detailed troubleshooting of the building of the management connection based on the output of the **show crypto isakmp sa** command.

ISAKMP/IKE Phase 2 Connections

In this section I'll discuss some router commands you can use to troubleshoot ISAKMP/ IKE Phase 2 connections. I'll begin by describing briefly the commands you can use and then, in later sections, discuss some of these commands in more depth.

Overview of the Phase 2 Commands

If you're experiencing problems with establishing IPsec data connections with an IPsec peer, there are several commands you can use to help pinpoint the problem. Here's a brief summary of these commands:

- **show crypto engine connections active** Displays each data SA that was built and the amount of traffic traversing each.
- **show crypto ipsec sa** Displays the data SAs established between two IPsec peers, and the components used to protect the connections and statistical information.
- **debug crypto isakmp** Displays the steps taken to build a management connection and data connections via the management connection (see "The **debug crypto isakmp** Command" section previously in the chapter).
- **debug crypto engine** Displays events related to encrypting and decrypting packets and applies to both Phase 1 and Phase 2 (see "The **debug crypto engine** Command" section previously in the chapter).
- **debug crypto ipsec** Displays the actual creation of the two unidirectional data SAs between two peers.
- **clear crypto sa** [**counters** | **map** *map_name* | **peer** *IP_address*| **spi** *IP_address* {**ah** | **esp**} *SPI_#*] Clears the statistics (**counters**), all data SAs associated with a crypto map (**map**), all data SAs associated with a peer (**peer**), or a particular data SA to a particular peer.

The following sections will discuss some of these commands in more depth.

The show crypto engine connection active Command

The **show crypto engine connection active** command displays the active SAs (management and data connections) terminated on the router, and the number of data packets encrypted and decrypted for each SA. Example 19-8 illustrates the use of this command.

Example 19-8. Using the *show crypto engine connection active* Command

```
r3640a# show crypto engine connection active
  ID Interface     IP-Address  State Algorithm        Encrypt  Decrypt
   1 Ethernet0/0   192.1.1.40  set   HMAC_SHA+AES_CBC       0        0
2001 Ethernet0/0   192.1.1.40  set   AES+SHA                0        5
2002 Ethernet0/0   192.1.1.40  set   AES+SHA                5        0
```

The first entry (ID #1) is the management connection and the following two entries (ID #2001 and #2002) are the two data connections. A state of "set" indicates that the connections have been fully established.

The show crypto ipsec sa Command

The **show crypto ipsec sa** command displays the crypto map entry information used to build data connections and any existing data connections to remote peers. Example 19-9 illustrates the use of this command. At the top of the display, you can see that the crypto map called "mymap" has been activated on ethernet0/0. The *local ident* and *remote ident* entries display the traffic that is to be protected (traffic between 192.168.2.0/24 and 192.168.3.0/24). The *#pkts encaps* and *#pkts decaps* displays the number of packets encapsulated or de-encapsulated using IPsec (AH or ESP); likewise, you can see the number of packets encrypted and decrypted, and the number of packets where a hash function was created or

New IPsec Troubleshooting Features

There are two new IPsec troubleshooting features you can use in the IOS:

- IPsec VPN Monitoring: IOS 12.3(4)T
- Invalid Security Parameter Index Recovery: IOS 12.3(2)T

The following three sections will discuss both of these features.

IPsec VPN Monitoring Feature

IPsec VPN monitoring is a feature new in IOS 12.3(4)T. This feature allows you to monitor VPN sessions to provide for enhanced troubleshooting. These enhancements include:

- Adding a description to IKE peers so that it becomes easier to identify the peer other than using their IP address or FQDN.
- Clearing a crypto session: before IOS 12.3(4)T, you had to clear both the Phase 1 and 2 connections to a peer individually to tear down the crypto session; in IOS 12.3(4)T, you can tear down both sets of connections with a single command.

The following two sections will discuss these enhancements.

Configuring IKE Peer Descriptions

To configure IKE peer descriptions, use the following configuration:

```
Router(config)# crypto isakmp peer {address peer_IP_address |
                    hostname peer_hostname}
Router(config-isakmp-peer)# description description
```

You first must specify the identity of the peer (based on the configuration of the **crypto isakmp identity** command), which takes you into a subcommand mode. The **description** command allows you to assign a 80-character description, including spaces, for the remote peer. This description will then appear in the output of various **show** commands.

Note

If multiple remote peers sit behind the same PAT device, you cannot use address as an identity type for a description, since they'll all have the same IP address.

Seeing Peer Descriptions in show Commands

There are two **show** commands that take advantage of the use of descriptions:

- **show crypto isakmp peer** [*IP_address_of_peer*] Briefly displays the IPsec peer connections and descriptions. Example 19-17 illustrates the use of this command, where the local peer (192.1.1.40) is connected to the remote peer 192.1.1.42. A description of "Connection to SiteA" was assigned to this peer.

  Example 19-17. Using the *show crypto isakmp peer* Command

  ```
  RTRA# show crypto isakmp peer
  Peer: 192.1.1.42 Port: 500 Local: 192.1.1.40
   Description: Connection to SiteA
   Phase1 id: 192.1.1.42
  ```

- **show crypto session**[**local** *local_IP_address*] [**remote** *remote_IP_address*] [**detail**] Displays status information for active crypto map sessions. Example 19-18 illustrates the use of this command without the **detail** parameter and 19-19 with it. In Example 19-18, you can see the peer the router is connected to (192.1.1.42) and that the management and two data connections were built. Table 19-1 explains the statuses that can appear in the output of this command. If no flow exists, this could be

Fragmentation Problems

IP supports a maximum length of 65,536 bytes for an IP packet; however, most data-link layer protocols support a much smaller length, called a maximum transmission unit (MTU). Based on the supported MTU, it might be necessary to break up (fragment) an IP packet to transmit it across a particular data-link layer media type. The destination then would have to reassemble the fragments back into the original complete IP packet.

The problem with fragmentation is that it requires additional CPU and memory resources on devices performing fragmentation. This is not to say that the actual source and destination will perform the fragmentation. For example, assume that the source and destination are on a Token Ring network with an MTU size of 4K. However, between these two networks is an Ethernet network. To transmit the 4KB data payloads across a 1500-byte Ethernet MTU, an intermediate device connected to the Token Ring and Ethernet networks would have to fragment the original packet. If the intermediate device supports fragmentation, additional resources (CPU and memory) are required to perform fragmentation on the original payload, and on the destination device, to reassemble the fragments back into a complete payload. In some instances, the intermediate device won't perform fragmentation and might drop the packet; in other cases, the intermediate device might notify the source of the correct MTU size to use.

When using a VPN to protect data between two VPN peers, additional overhead is added to the original data, which might require that fragmentation occur. Table 19-2 lists fields that might have to be added to the protected data to support a VPN connection. Please note that multiple protocols might be necessary, increasing the size of the original packet. For example, if you're using an L2L DVMPN IPsec connection between two Cisco routers where you've implemented a GRE tunnel, you'll have this additional overhead: ESP, GRE, and the outer IP header. If you have an IPsec software client connecting to a VPN gateway where the traffic is going through an address translation device, you'll have this additional overhead: ESP, UDP header for NAT-T, and the outer IP header for the tunnel mode connection.

Table 19-2. VPN Overhead

Protocol	Additional Bytes
ESP (encryption and hash)	56
AH	24+
GRE	24
NAT-T/IPsec over UDP (UDP part)	8
IPsec over TCP (TCP part)	20
L2TP	12
PPTP	48
Outer IP header in IPsec tunnel mode or PPTP/L2TP	20
PPPoE	8

Given that this book is about VPNs, my concern is that I need to get protected traffic between two devices and I definitely don't want fragmentation occurring, because this places extra overhead on the source (or intermediate) and destination devices and runs the risk of breaking the VPN session. The remainder of this chapter will focus on problems that fragmentation can create and how you can deal with them on a Cisco IOS router. Please note that much of what I talk about here applies also to the VPN concentrator, PIX and ASA

Summary

This chapter showed you the basics of troubleshooting IPsec VPNs on Cisco routers. It is by no means an all-encompassing coverage of troubleshooting VPN problems; however, I've tried to discuss some of the most common problems you'll experience and how to pinpoint those problems using **show** and **debug** commands on Cisco routers.

Next up is Part IV, "PIX Firewalls," where I show you how to set up VPN features on Cisco PIX and ASA appliances.

PREV NEXT

Part V: PIX Firewalls

PREV NEXT

Chapter 20. PIX and ASA Product Information

In Part V, I'll discuss the use of Cisco PIX and ASA security appliances to initiate and terminate VPN sessions. PIXs and ASAs are very flexible and can be used for site-to-site (LAN-to-LAN or L2L) sessions, VPN gateways, and remote access clients. I'll cover topics on how to configure the PIX security appliances using the new 7.0 Finesse Operating System (FOS) and its predecessor, 6.x (the ASA only supports version 7.0).

In this chapter, however, I'll introduce you to the Cisco PIX and ASA security appliance solutions, focusing on VPN deployment scenarios and their VPN capabilities. I'll also discuss some of the advantages that Cisco PIXs and ASAs have over other Cisco products when being used for VPN solutions.

PIX Deployment Scenarios

The Cisco PIX and ASA VPN capabilities have their roots in Cisco IOS VPN technologies. VPNs were first introduced in the Cisco IOS router product line and then added to the PIXs in an early 5.x release. Like the routers and the concentrators, Cisco PIXs support many VPN solutions including IPsec, PPTP, and L2TP. Because of their flexibility, they can be used in many different situations. The ASA was introduced in the spring of 2005. The ASA is a unique hybrid security appliance, having abilities from the PIX, VPN 3000, and IDS 4200 sensors. This section will focus on how PIX and ASA security appliances can be used to enhance a VPN solution in your network.

Specifically, the section will cover the following:

- L2L and Remote Access Connections
- The Special Capabilities of PIXs and ASAs

L2L and Remote Access Connections

PIXs and ASAs support L2L and remote access connections. For remote access solutions, the PIXs and ASAs can be Easy VPN Servers and the PIX 501 and 506E can be Easy VPN Remotes (clients). As I mentioned in Chapter 9, "Concentrator Site-to-Site Connections," I prefer to use Cisco routers for L2L sessions and concentrators for remote access connections. With the introduction of the ASA security appliances, they also can terminate SSL VPNs, with similar SSL capabilities compared to the VPN 3000 concentrators.

Routers support enhanced routing and QoS capabilities over Cisco PIX and ASA security appliances and VPN 3000 concentrators. Plus, VPN 3000 concentrators scale better for remote access connections and are easy to set up. However, the Cisco PIX and ASA security appliances, first and foremost, provide better-integrated and more comprehensive security services than routers and concentrators. Therefore, if you need to enhance your VPN solution with security and firewall functions and place it in one box, or if you need enhanced address translation services for VPNs that terminate on a VPN device, the PIX or ASA is a much better choice than a router or a concentrator.

Special Capabilities of PIXs and ASAs

As I mentioned in Chapter 6, I prefer to use PIXs or ASAs in a VPN solution when I need advanced address translation capabilities in addition to advanced firewall and security services. There are three main features the PIX and ASA security appliances have over Cisco VPN 3000 concentrators and IOS-based routers when it comes to VPN implementations: address translation, stateful firewall services, and redundancy.

Address Translation

The PIX was originally developed by Network Translation as an address translation device back in 1994. From the beginning, the PIX has had its roots in address translation. The concentrator's address translation capabilities are very minimal and Cisco routers' capabilities are based primarily on address translation involving two logical locations: inside and outside. However, the PIX's address translation capabilities can handle multiple interfaces easily, with different translation policies for different interfaces. Policy address translation is one of its main strengths. Many times I've attempted to configure complex address translation policies, such as bidirection NAT on a multi-interfaced router, and then shortly gave up and easily configured the same policies on a PIX.

Stateful Firewall Services

With the introduction of FOS 6.x and 7.0, the PIX and ASA security appliances provide one of the best, if not the best, integrated stateful firewall services in the market, including support for both IPv4 and IPv6. Besides performing stateful firewall functions, they support superb application layer inspection and filtering capabilities, including detailed inspection of application layer information such as HTTP, FTP, SMTP, ESMTP, multimedia applications, voice, and many others. They support advanced guard and detection features to protect against TCP flood attacks, DNS spoofing, fragmentation attacks, web server attacks, and e-mail attacks. The PIX and ASA also can be used to detect and block instant messaging applications,

PIX and ASA Feature and Product Overview

In the next two sections, I'll discuss briefly some of the VPN features and VPN capabilities of the PIX and ASA security appliances. This information can help you in determining if the PIX or ASA is the right product platform for you, and which PIX or ASA appliance you should choose for a VPN implementation.

PIX and ASA VPN Features

Cisco PIX and ASA security appliances support many VPN features. They are fully IPsec-compliant and support both L2L and remote access services, where they can perform both Easy VPN Server and Remote functions. The features listed below include features that have existed in older FOS versions, and those that are new in FOS 7.0.

- All PIXs and ASAs support AES, DES, and 3DES encryption algorithms (with AES added in FOS 6.3) and MD5 and SHA hashing functions; DH groups 1, 2, and 5 are supported, with FOS 7.0 adding support for DH group 7.
- As an Easy VPN Remote, the PIX can use client and network extension modes with RRI, and can perform device, interactive user, and individual user authentication (like the 3002 hardware client); ASAs cannot be Easy VPN Remotes, but they can be Easy VPN Servers.
- VPN client security posture enforcement allows the PIX and ASA to perform NAC for VPN clients, restricting clients' access based on their operating system and type, and installed security product software (like antivirus and personal firewall software)new in FOS 7.0.
- Like the VPN 3000 concentrators, the PIXs and ASAs can perform automatic software updates for the 3002 hardware clients and Cisco VPN software clientsnew in FOS 7.0.
- The PIX and ASA can act as a hub in a hub-and-spoke topologynew in FOS 7.0, where traffic can be inspected and policies enforced before allowing traffic to flow between the spokes via the hub PIX or ASAnew in FOS 7.0.
- The PIX has supported NAT-T since FOS 6.2; however, FOS 7.0 has added support for IPsec over TCP and Cisco IPsec over UDP.
- The PIX and ASA can route OSPF traffic across VPN tunnels and can inject RRI routes into an OSPF routing processnew in FOS 7.0.
- In FOS 7.0, support has been added to enroll for certificates manually. Other features are the export of private keys, support for hierarchical CA implementations, the increase of RSA key sizes to 4,096 bits, support for DSA-based X.509 certificates, and use of an IOS router acting as a CA.
- The ASA supports the termination of SSL VPNs from SSL clients. Currently, the PIX does not support this feature; the ASA's SSL VPN capabilities are more comparable to the VPN 3000 concentrators than to Cisco routers.

As you can see from the preceding list, many VPN improvements have been added in FOS 7.0.

Note

Up until 6.3, all of the PIXs supported the same FOS, and with a few exceptions of features, all had the same capabilities. When FOS 7.0 was introduced, it was initially available only on the 515/515E and higher PIXs, with possible support for the 501 and 506E to be added later.

Pre- and Post-FOS 7.0

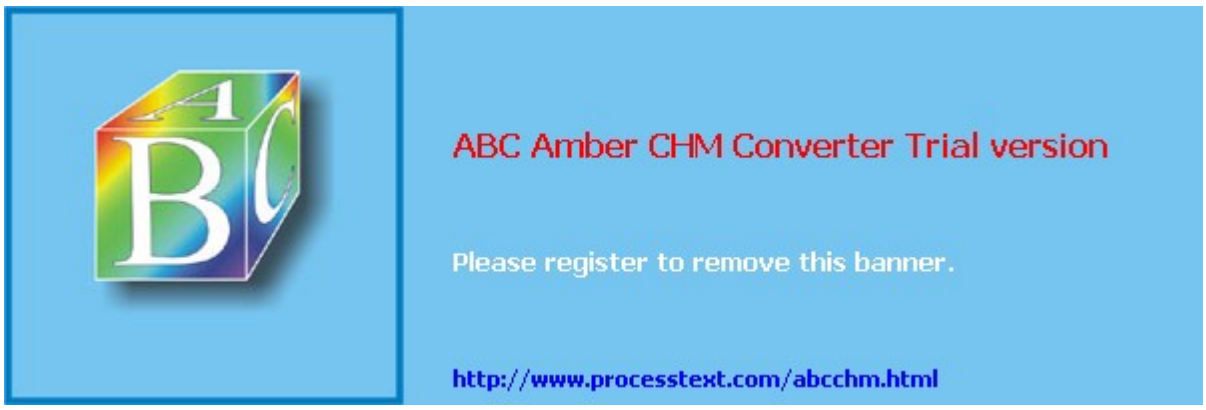
In FOS 6.3 and earlier, I typically shied away from using the PIX as a VPN solution because of the many VPN features that it lacked (the ASA only supports FOS 7.0 and later). Probably my biggest issue with the pre-7.0 FOS was that traffic that entered a PIX's interface could not leave the same interface. This meant that the

Summary

This chapter introduced you to the PIX and ASA security appliance family. I've attempted to give you a brief introduction to the PIXs and ASAs so that you have an understanding of their capabilitiesstrengths and weaknessesand therefore can properly choose the correct solution, and the correct PIX or ASA, when designing a VPN implementation.

Next up is Chapter 21, "PIX and ASA Site-to-Site Connections," where I show you how to create L2L sessions on these devices.

PREV NEXT



Chapter 21. PIX and ASA Site-to-Site Connections

In this chapter I'll discuss how to configure IPsec LAN-to-LAN (L2L) sessions on the PIX and ASA security appliances. The first part of the chapter focuses on the components you'll need to configure the management connection, much of which applies to remote access sessions, and the second part will focus on configuring the components of the data connections. At the end of the chapter I'll illustrate an example of an L2L session between PIXs/ASAs.

In April 2005, Cisco introduced a new version of the Finesse Operating System (FOS) for the PIX security appliances, called 7.0. Currently this is supported only on the 515/515E PIXs and higher. Likewise, in May 2005, Cisco introduced the new Adapter Security Appliance (ASA) devices, which support PIX, VPN concentrator, router, and IDS features all in one box. Fortunately, much of the code and commands found in the 7.0 PIX security appliances are the same as those found in the ASA devices. However, the 501 and 506/506E PIXs only support the FOS 6.3 software. Because the 7.0 software is new, and the 6.*x* software is still in wide use, I'll point out differences in the configurations of both operating systems throughout the chapter where appropriate.

Note

In version 7.0, the PIX/ASA supports VPN only in single mode, commonly called routed mode. VPNs are not supported when your PIX/ASA is configured for multiple security contexts (multi-mode) or in an Active/Active stateful failover configuration. In FOS 6.3 and earlier, the stateful failover feature of the PIXs did not provide stateful failover for VPN sessions; in FOS 7.0, this enhancement has been added. The configuration of failover and stateful failover on the PIX/ASA, however, is beyond the scope of this book. Topics such as tunnel groups, which were added in FOS 7.0, I'll address in Chapter 22, "PIX and ASA Remote Access Connections," where it is more appropriate.

ISAKMP/IKE Phase 1 Management Connection

In this first part of the chapter, I'll focus on the components necessary to allow IPsec traffic into the PIX/ASA and to build a management connection to a remote peer. Much of what I discuss here is applicable to both L2L and remote access sessions.

Allowing IPsec Traffic

Your first task is to allow IPsec session traffic into your PIX/ASA. Unlike Cisco routers, PIX/ASA devices behave differently when traffic is flowing through them. With these security appliances, interfaces are assigned security levels, and based on security level configurations, traffic is not allowed to flow from a lower to a higher level, by default. In most cases, your IPsec session traffic will be terminated on the device's outside interface, which is, from the FOS's perspective, the least secure.

You can use two methods to allow VPN traffic into or through the PIX from a less- to more-secure interface:

- Access control lists (ACLs)
- ACL bypassing

The following two sections will discuss both options. After this, I'll discuss one issue with moving traffic between interfaces that have the same security level.

Using ACLs to Allow IPsec Traffic

If you decide you'll use ACLs to allow IPsec traffic into your PIX/ASA, your ACL configuration will look something like the following:

```
appliance(config)# access-list ACL_name_or_# permit udp
                        remote_peer_IP_address subnet_mask
                        local_IP_address subnet_mask eq 500
appliance(config)# access-list ACL_name_or_# permit esp
                        remote_peer_IP_address subnet_mask
                        local_IP_address subnet_mask
appliance(config)# access-list ACL_name_or_# permit udp
                        remote_peer_IP_address subnet_mask
                        local_IP_address subnet_mask eq 4500
appliance(config)# access-list ACL_name_or_# permit tcp
                        remote_peer_IP_address subnet_mask
                        local_IP_address subnet_mask eq port_#
appliance(config)# access-list ACL_name_or_# permit protocol
                        remote_protected traffic subnet_mask
                        local_protected traffic subnet_mask
                        [protocol_information]
```

The first ACL statement allows the management connection to be built. The second, third, and fourth statements are for the data connections. The second statement assumes that ESP traffic can reach the PIX/ASA without having to go through a PAT device. The third statement is for ESP traffic that is being encapsulated using NAT-T. New in FOS 7.0 is the ability of encapsulating ESP traffic in TCP, which statement four would allow. You can list up to ten TCP ports for IPsec over TCP, but the default is port 10,000; you'll need to list all ports you've configured the PIX/ASA to use, if any, in separate ACL statements.

The last ACL statement in the preceding syntax allows the tunneled packets to pass the ACL check. As mentioned in Chapter 17, "Router Site-to-Site Connections," routers process IPsec packets twice on the ACL (depending on the IOS version): once when they're protected and again after the router has verified the protection and decrypted them (in their clear-text form). PIX/ASA devices do the same thing. You'll need to list, probably in multiple ACL statements, the traffic that is allowed *through* the tunnel.

One other thing to point out about the ACL configuration is that in FOS 6.3, both AH and ESP are supported; however, in the initial release of FOS 7.0, only ESP is supported. If you are

ISAKMP/IKE Phase 2 Data Connections

This part of the chapter will discuss the components you need to configure for the Phase 2 data connections in an IPsec L2L session for your PIX/ASA security appliance. This includes specifying what traffic to protect and how it should be protected. If you've configured L2L sessions on routers, the process and commands described here will be nothing new to you.

Note

Please note that even though Cisco has added many IPsec enhancements in 7.0, the PIX/ASA still lags behind routers when it comes to deploying scalable L2L IPsec sessions. For instance, 7.0 doesn't support advanced QoS, GRE tunnels, or DMVPN. Therefore, I typically use the PIX/ASA with a small number of L2L sessions where I need advanced address translation configurations, or want to enhance the security of the IPsec implementation.

Specifying Traffic to Protect

ACLs are used to specify what traffic is to be protected for an IPsec session to a particular peer, commonly called a crypto ACL:

```
appliance(config)# access-list ACL_name_or_# {permit | deny}
                     protocol_name_or_#
                     src_IP src_subnet_mask [protocol_info]
                     dst_IP dst_subnet_mask [protocol_info]
```

Any ACL command in the grouping that has a **permit** statement specifies traffic to be protected, whereas a **deny** or implicit deny statement specifies traffic that doesn't have to be protected. Unlike on an IOS router, PIX/ASA ACLs use a subnet mask to match on a range of addresses, whereas routers use a wildcard (inverted) mask. The source IP address/network specifies locations connected to the local PIX/ASA and the destination IP address/ network specifies locations connected to the remote peer.

Caution

As with IOS routers, you should be very careful about using the keyword **any** (0.0.0.0/0) for the source or destination address in a crypto ACL, because this includes broadcast and multicast addresses and can cause certain data transmissions to fail. Therefore, you should be as specific as possible concerning what traffic is to be protected.

Defining How to Protect Traffic

As with IOS routers, a transform set is used to specify how traffic is to be protected. How this is configured is similar to configuration for a Cisco IOS router. Here are the commands to create a transform set on a PIX/ASA:

```
appliance(config)# crypto ipsec transform-set transform_set_name
                     transform1 [transform2 [transform3]]
appliance(config)# [no] crypto ipsec transform-set transform_set_name
                     mode transport
```

Each transform set must have a unique name. Within each set, you define between one to three transforms: one for AH authentication, one for ESP authentication, and one for ESP encryption. Here are the valid transform names: **ah-md5-hmac**, **ah-sha-hmac**, **esp-md5-hmac**, **esp-sha-hmac**, **esp-des**, **esp-3des**, **esp-aes** (128-bit), **esp-aes-192**, **esp-aes-256**, and **esp-null** (no encryption). The default connection mode of the transform is tunnel. This can be changed with the **mode** parameter after you already have created a transform set (this is different from a router, where the mode is configured in the transform set's subcommand mode). You can use the **show crypto ipsec transform-set** command in 6.3 to view your configured transform sets. In 7.0, use the **show running-config crypto** command.

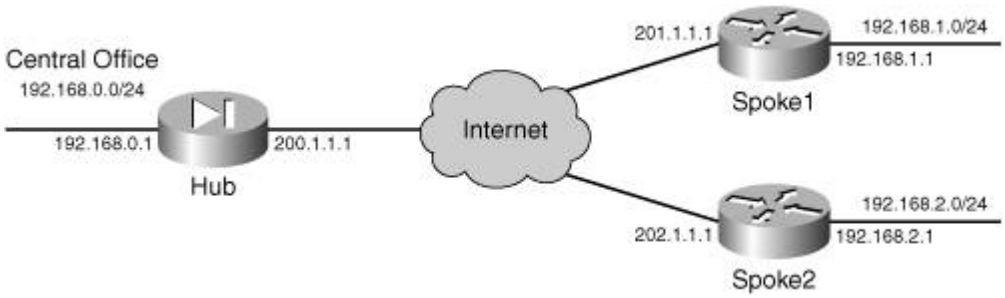
Note

L2L Connection Examples

Now that you have a basic understanding of the commands used to build L2L IPsec sessions on a PIX/ASA appliance, I'll show you a couple of examples that illustrate the configurations. The first example will be based on FOS 6.3 for the PIX appliances and the second example on 7.0. I'll use the same situation in both examples, shown in Figure 21-1. In the figure, one PIX functions as a hub and two PIXs as spokes connecting to the central site via the hub PIX.

Figure 21-1. L2L Simple Hub-and-Spoke Design

[View full size image]



FOS 6.3 L2L Example

Examples 21-4, 21-5, and 21-6 illustrate the configuration of the hub and two-spoke PIXs using FOS 6.3. Each example contains reference numbers, which are explained following the specific example.

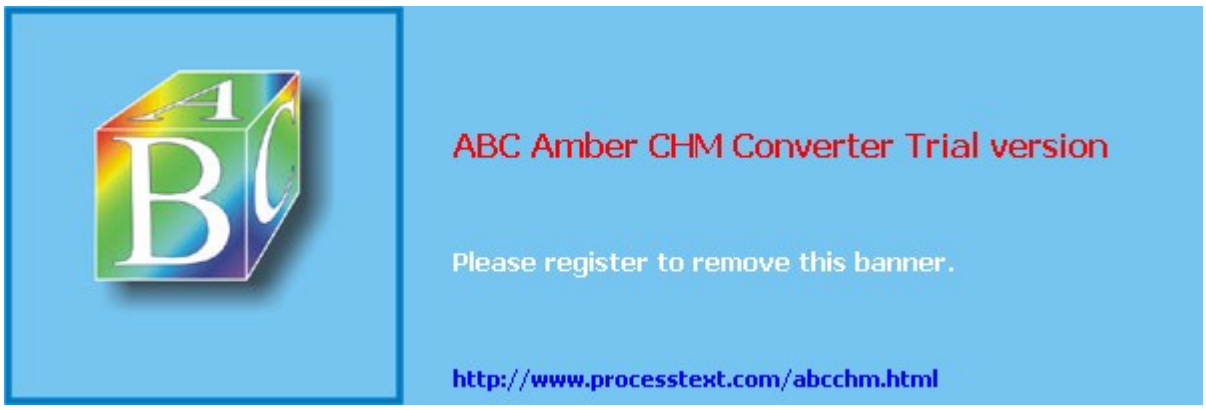
Example 21-4. Hub Configuration in FOS 6.3

```
hub(config)# ip address outside 200.1.1.1 255.255.255.0
hub(config)# ip address inside 192.168.0.1 255.255.255.0
hub(config)# route outside 0.0.0.0 0.0.0.0 200.1.1.2 1
hub(config)# access-list nonat permit ip                          (1)
                        192.168.0.0 255.255.255.0
                        192.168.1.0 255.255.255.0
hub(config)# access-list nonat permit ip
                        192.168.0.0 255.255.255.0
                        192.168.2.0 255.255.255.0
hub(config)# nat (inside) 0 access-list nonat
hub(config)# sysopt connection permit-ipsec                       (2)
hub(config)# isakmp enable outside                                (3)
hub(config)# isakmp identity address
hub(config)# isakmp policy 10 authentication pre-share
hub(config)# isakmp policy 10 encryption aes
hub(config)# isakmp policy 10 hash sha
hub(config)# isakmp policy 10 group 2
hub(config)# isakmp keepalive 20 3
hub(config)# isakmp key cisco123 address 201.1.1.1               (4)
                        netmask 255.255.255.255
                        no-xauth no-config-mode
hub(config)# isakmp key cisco123 address 202.1.1.1
                        netmask 255.255.255.255
                        no-xauth no-config-mode
hub(config)# access-list 101 permit ip                            (5)
                        192.168.0.0 255.255.255.0
                        192.168.1.0 255.255.255.0
hub(config)# access-list 102 permit ip
                        192.168.0.0 255.255.255.0
                        192.168.2.0 255.255.255.0
hub(config)# crypto ipsec transform-set mytrans                   (6)
                        esp-aes esp-sha-hmac
hub(config)# crypto map mymap 20 ipsec-isakmp                     (7)
hub(config)# crypto map mymap 20 match address 101
hub(config)# crypto map mymap 20 set peer 201.1.1.1
hub(config)# crypto map mymap 20 set transform-set mytrans
```

Summary

This chapter showed you the basics of setting up ISAKMP/IKE Phase 1 policies and configurations on Cisco PIX and ASA security appliances and on L2L IPsec sessions. As you can see from this chapter, when compared to Chapter 16 and 17 on IOS routers, Cisco security appliances have less capabilities when it comes to features with ISAKMP/IKE Phase 1 connectivity and scalable L2L IPsec sessions.

Next up is Chapter 22, "PIX and ASA Remote Access Connections," where I show you how to set up a PIX/ASA as an Easy VPN Server and a PIX as an Easy VPN Remote.



Chapter 22. PIX and ASA Remote Access Connections

Where the last chapter focused on IPsec L2L sessions with Cisco security appliances, this chapter will focus on their IPsec Easy VPN Server and Remote features. Both PIXs and ASA security appliances can perform the role of an Easy VPN Server, acting as a VPN gateway for client (Remote) devices; however, only the PIX 501 and 506/506E security appliances currently can perform the role of an Easy VPN Remote or hardware client. Cisco plans to release a low-end ASA appliance later that will also support this functionality.

PREV NEXT

Easy VPN Server Support for 6.*x*

Since the release of FOS 6.2, the PIX security appliances, from the 501 all the way up to the 535, can perform the function of an Easy VPN Server; and with the addition of the ASA appliances, they, too, can perform this function. Normally, I prefer to use a VPN 3000 concentrator to support a large number of remote access users; however, if you already have a PIX/ASA appliance in place and need to support only a small number of clients, you can use your existing PIX/ASA for this function. This is preferable to a router if your PIX supports a VAC+ encryption card to perform hardware encryption and your router lacks this, or if you need advanced address translation capabilities or security functions or features, which the router might lack.

The configuration of an Easy VPN Server is different if you're running FOS 6.3 or earlier when compared to 7.0. Because of the differences, I've split up the configuration explanation into the following two sections: one for 6.3 (this section) and one for 7.0 (the following main section).

Easy VPN Server Configuration for 6.*x*

Starting in FOS 6.2 and in later FOS releases, the PIX and ASA (7.0) appliances support the Easy VPN Server function, which allows them to terminate IPsec sessions from Easy VPN Remote devices, including the Cisco VPN Client software, the 3002 hardware client, the 800, ubr900, and 1700 routers, and the PIX 501 and 506E security appliances. As with the routers and concentrators performing the Easy VPN Server function, groups are used to apply policies to the Remotes connecting the Server.

Configuring an Easy VPN Server is broken into these components:

Step 1. Create an address pool for remote access devices' internal addresses with the **ip local pool** command (this is required only for client mode connectionsnetwork extension mode connections do not require this).

Step 2. Define group policies for remote access users with the **vpngroup** command.

Step 3. Disable address translation for the users' internal addresses with the **nat** (*interface*) **0 access-list** *ACL_name* command (discussed in Chapter 21).

Step 4. Enable XAUTH with the **crypto map** *map_name* **client authentication** command.

Step 5. Create ISAKMP policies with the **isakmp policy** command (discussed in the last chapter).

Step 6. Define a compatible tunnel-mode transform set with the **crypto ipsec transform-set** command (discussed in the last chapter).

Step 7. Create a dynamic crypto map with the **crypto dynamic-map** command (discussed in the last chapter).

Step 8. Create a static crypto map and enable it with the **crypto map** command (discussed in the last chapter).

Step 9. Enable IKE Mod Config for the static crypto map that has the dynamic crypto map reference with the **crypto map** *map_name***client configuration** command.

Step 10. Allow IPsec traffic with an ACL or the **sysopt connection permit-ipsec** command (discussed in the last chapter).

As you can see from the above steps, many of the things you have to configure I've already discussed in the last chapter, "PIX and ASA Site-to-Site Connections." Therefore, I'll focus primarily on Steps 1, 2, 4, and 9 in my discussions throughout this section. With hardware clients connecting to the PIX Server, additional items may be configured, like the type of connection (client versus network-extension mode) and the type of authentication (default,

Easy VPN Remote Support for 6.*x*

Starting in FOS 6.2 and 6.3, a PIX 501 or 506/506E can perform the function of an Easy VPN Remote, initiating client connections to an Easy VPN Server, like a VPN 3000 concentrator, a Cisco IOS router, or another PIX or ASA security appliance. If you recall from Chapter 14, "3002 Hardware Client," and Chapter 18, "Router Remote Access Connections," the 3002 and 800, ubr900, and 1700 routers are also hardware clients. One advantage of setting up a low-end PIX as a Remote is that minimal configuration is needed on the PIX to establish an IPsec session to an Easy VPN Server; policies are centralized on the Server and pushed down to the Remote during IKE Mode Config, and individual users behind the PIX do not need to load an IPsec software client on their desktops.

Tip

A low-end PIX is an ideal hardware client for SOHO environments where you have a cable or DSL modem connection and need to use split tunneling; the PIX provides many security features to deal with the traffic that is not protected. It's my preferred hardware client in this situation. Also, the low-end PIXs can perform Server functions to handle a small number of Remote connections, which is ideal for the 23 users that need to work from home.

The PIX Remote supports many of the features of the Cisco VPN Client software and the VPN 3002, including:

- Pre-shared key and certificate device authentication
- Client and network extension modes
- Split tunneling and split DNS
- Backup server lists (6.3)
- AES encryption (6.3) and DH group 5 (6.3)
- Unit authentication (6.2), individual user authentication (6.3), and secure unit authentication (6.3), which is similar to Interactive Unit Authentication on the 3002

The following two sections will discuss the commands to configure a low-end PIX as a Remote and also an example configuration.

Note

Version 7.0 is not supported on the 501 and 506/506E PIX firewalls; only on the PIX 515/515E, 525, and 535. Therefore, it is not currently possible to configure a PIX (or ASA) running 7.0 as an Easy VPN Remote device.

6.*x* Easy VPN Remote Configuration

Like a low-end IOS router Remote device, configuring a low-end PIX Remote is straightforward and requires few commands. Use the following commands to set up the PIX Remote to connect to the Easy VPN Server:

```
pix(config)# vpnclient vpngroup group_name password preshared_key
pix(config)# vpnclient username XAUTH_username
             password XAUTH_password
pix(config)# vpnclient server Server_IP_address_1
             [Server_IP_address_2 ...Server IP address_11]
pix(config)# vpnclient mode{client-mode| network-extension-mode}
pix(config)# vpnclient mac-exempt MAC_addr_1 MAC_mask_1
             [MAC_addr_2 MAC_mask_2]
pix(config)# vpnclient management tunnel IP_address_1 subnet_mask
             [IP_address_2 subnet_mask]
pix(config)# management-access interface_name
pix(config)# vpnclient nem-st-autoconnect
```

Easy VPN Server Support for 7.0

Continued support for Easy VPN Server is provided in FOS 7.0. Many, many new enhancements have been made to the Server function, putting the PIX and ASA more in line with the IPsec capabilities of the VPN 3000 concentrators. However, the configuration of an Easy VPN Server is very different than what I discussed earlier for 6.*x*.

This part of the chapter will focus on the configuration of a PIX/ASA as an Easy VPN Server. The following are tasks you need to perform on the 7.0 security appliance to set it up as a Server:

- Enabling ISAKMP, including policies (discussed in the last chapter)
- Defining IP address pools (discussed previously in the "Address Pool Configuration for 6.*x*" section)
- Creating tunnel groups (this is new in 7.0)
- Specifying where user accounts are located, locally on the PIX/ASA or on an AAA server (discussed previously in the "XAUTH User Authentication Configuration for 6.*x*" section)
- Creating user accounts, if specified locally
- Defining IPsec transform sets for data connections (discussed in the last chapter)
- Creating a dynamic crypto map for remote access users (discussed in the last chapter)
- Referencing the dynamic crypto map as an entry in a static crypto map (discussed in the last chapter)
- Activating the static crypto map on the PIX/ASA's interface

Because I described many of the above tasks previously in this and the last chapter, the next few sections will focus on configurations unique to FOS 7.0, including:

- Understanding Tunnel Groups
- Defining Group Policies
- Creating Tunnel Groups
- Creating User Accounts for XAUTH

Following these sections, I'll also discuss issues with remote access sessions and solutions that are provided in FOS 7.0, and an example that will illustrate the configuration of an Easy VPN Server running FOS 7.0.

Understanding Tunnel Groups

The main IPsec change from 6.3 to 7.0 is the introduction of the tunnel group feature. Tunnel groups allow you to define VPN session policies associated with a particular session or group of sessions, like a related group of remote access users or L2L sessions. Tunnel groups are used to simplify the configuration and management of your IPsec sessions. By default, two tunnel groups already are created on your PIX/ASA: "DefaultL2LGroup" for L2L sessions and "DefaultRAGroup" for remote access sessions. A tunnel group might include parameters such as general policy information and information to build IPsec sessions.

A group policy is used to define attributes associated with a user or group of users. It is associated with a tunnel group. A default group policy, called "DfltGrpPolicy," exists on the security appliance for users who are not associated with a specific remote access group (similar to the Base Group on a VPN 3000 concentrator).

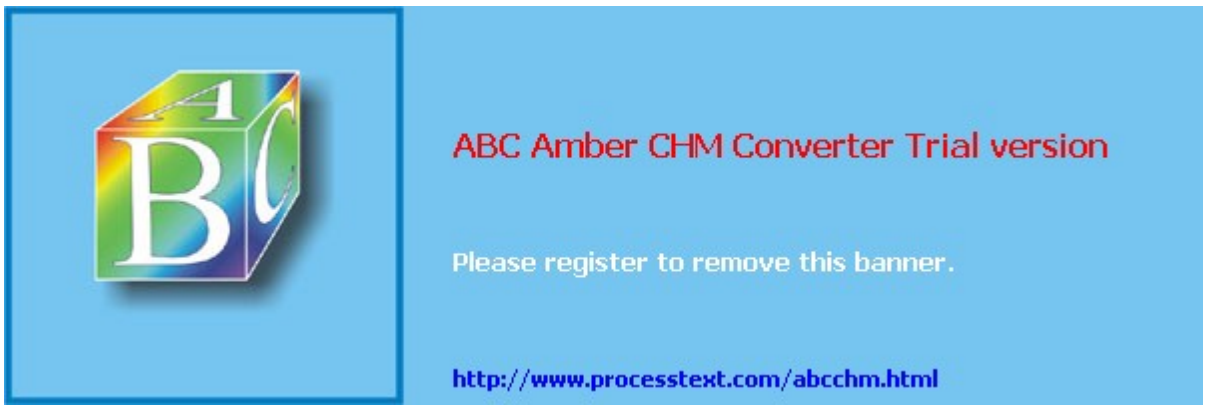
The use of tunnel groups involves three configuration steps:

Step 1. Define group policies.

Summary

This chapter showed you the basics of setting up your security appliance as an Easy VPN Server using both the older PIX FOS (6.*x*) and the newer 7.0 (PIXs and ASAs). And in 7.0, support for WebVPN was added for the ASAs. In 6.*x*, the PIX 501 and 506E can also be Remotes.

Next up is Chapter 23, "Troubleshooting PIX and ASA Connections," where I show you how to use basic security appliance commands to troubleshoot the setup of VPN sessions.



Chapter 23. Troubleshooting PIX and ASA Connections

This chapter will focus on how to troubleshoot IPsec sessions on Cisco PIX and ASA security appliances. The layout of this chapter is similar to that found in Chapter 19, "Troubleshooting Router Connections." I've broken the chapter into two areas on troubleshooting: ISAKMP/IKE Phase 1 and ISAKMP/IKE Phase 2 issues. With these two areas, I'll show you how ISAKMP/IKE Phase 1 and 2 connections are built, and what to look for when there is a problem with either of these phases.

This chapter by no means covers all possible problems you'll experience with IPsec sessions on Cisco security appliances. However, I hope to provide you with the basic background knowledge so that troubleshooting IPsec sessions on the appliances is a simpler process.

ISAKMP/IKE Phase 1 Connections

In the first part of this chapter I'll focus on troubleshooting ISAKMP/IKE Phase 1 connections. If you recall from Chapter 3, "IPsec," the management connection built during Phase 1 is used to pass IPsec management traffic; no user data traverses this connection. This connection is important, however, because it is used to build the two data connections for Phase 2. I've broken this part of the chapter into three areas:

- An overview of the ISAKMP/IKE Phase 1 troubleshooting commands
- Examining your management connections
- Examining the building of L2L and remote access management connections
- Troubleshooting Easy VPN connections on a Remote

Overview of the Phase 1 Commands

You can use several commands to troubleshoot ISAKMP/IKE Phase 1 connections on the security appliances, including the following:

- **show isakmp sa [detail]** Displays the status of any management connections.
- **show [crypto] isakmp stats** Displays the statistics of the management connections (FOS 7.0 only).
- **show [crypto] isakmp ipsec-over-tcp stats** Displays the statistics of any IPsec over TCP connections the management connection is managing (FOS 7.0 only).
- **debug crypto isakmp** Displays the steps taken to build a management connection and data connections via the management connection.
- **debug crypto vpnclient** Displays the interaction between the appliance, acting as an Easy VPN Remote, and the Easy VPN Server (FOS 6.3 only).
- **debug crypto ca [messages | transactions]** Displays the interaction between the appliance and CA for certificate enrollment and authentication functions; the optional parameters are new in FOS 7.0. The 7.0 version of this command produces similar output compared to the **debug crypto pki** command discussed in Chapter 19; therefore, I won't cover it in this chapter.
- **debug crypto engine** Displays events related to the encryption/decryption problems on the appliance.
- **clear [crypto] isakmp sa** [*SA_ID_#*] Deletes all the management SAs or a specific management connection by specifying the SA ID number.

As you can see from the above list, not all commands are supported in all FOS versions. The following sections will discuss some of the more important commands, related to troubleshooting connectivity processes, in more depth.

Note

Before FOS 7.0, I found the output of **debug** commands less administrator-friendly than the debug output from IOS routers. In FOS 6.3 and earlier, I tended to try to troubleshoot IPsec problems from the remote peer and would look at the PIX's debug output only when I was still having problems trying to pinpoint the problem. However, Cisco has rectified most of my concerns in regard to this in FOS 7.0. In FOS 7.0, the debug output is much more similar to the debug output of IOS-based routers.

The show isakmp sa Command

Example 23-1 illustrates the use of the **show isakmp sa** command with an appliance running FOS 6.3. The output of this command is very similar to the **show crypto isakmp sa** command in Chapter 16, "Router ISAKMP/IKE Phase 1 Connectivity." Table 16-1 in that chapter explains

ISAKMP/IKE Phase 2 Connections

In this section I'll discuss some security appliance commands you can use to troubleshoot ISAKMP/IKE Phase 2 connections. I'll begin by briefly describing the commands you can use and then, in later sections, I'll discuss some of these commands in more depth.

Overview of the Phase 2 Commands

If you're experiencing problems with establishing IPsec data connections with an IPsec peer, you could use several PIX/ASA commands to help pinpoint the problem. Here's a brief summary of these commands:

- **show crypto engine [verify]** Displays the usage statistics for the appliance's crypto engine (FOS 6.x only); the verify parameter runs the Known Answer Test (KAT), which checks the integrity of the cryptography engine used by the appliance.
- **show crypto interface [counters]** Displays the VAC/VAC+ card installed in the appliance and, optionally, traffic statistics for the card (FOS 6.x only).
- **show crypto accelerator statistics** Displays the VAC/VAC+ card installed in the appliance and, optionally, traffic statistics for the card (FOS 7.0 only).
- **show crypto protocol statistics** {**ikev1** | **ipsec**} Displays general traffic statistics about the management or data connections (FOS 7.0 only).
- **show [crypto] ipsec sa** Displays the data SAs established between two IPsec peers, and the components used to protect the connection and packet statistical information.
- **debug crypto isakmp** Displays the steps taken to build a management connection and data connections via the management connection (see "The **debug crypto isakmp** Command" section previously in the chapter).
- **debug crypto ipsec** Displays the actual creation of the two unidirectional data SAs between two peers.
- **clear crypto [ipsec] sa [counters | map***map_name*|**peer** *IP_address*| **entry** *IP_address*{**ah** | **esp**} *SPI_#*] Clears the statistics (**counters**), all data SAs associated with a crypto map (**map**), all data SAs associated with a peer (**peer**), or a particular data SA to a particular peer (**entry**).

The following sections will discuss some of the above troubleshooting commands in more depth; these commands discussed below are the more common ones used to troubleshoot data SA problems.

The show crypto ipsec sa Command

The **show crypto ipsec sa** command displays the crypto map entry information used to build data connections, and any existing data connections to remote peers. There are two forms of the command, depending on which FOS version your appliance is running.

In FOS 6.x and earlier, the following syntax applies:

```
pix63# show crypto [ipsec] sa [map map_name | address | identity]
```

Without any specified parameters, the crypto map information used to create the data SAs is displayed, in addition to traffic statistics, and the inbound and outbound connections and their SPI numbers. The **map** parameter allows you to display only the SAs associated with the specified crypto map name. The **address** parameter sorts the output based on the IP address of the SA. The **identity** parameter sorts the display by SA flows.

In FOS 7.0, the following syntax can be used:

```
pix70# show crypto [ipsec] sa [entry | identity| map map_name |
                        peer peer_IP_address ] [detail]
```

Summary

This chapter showed you the basics of troubleshooting IPsec sessions on Cisco security appliances. The commands and processes used are very similar to those on IOS-based routers, reducing your learning curve if you already have experience with IPsec tunnels on IOS-based routers. Remember that in FOS 7.0, the **debug** commands have a level qualifier which affects the amount of debug output the command generates.

This chapter completes the configuration and troubleshooting part of this book. Next up is Part VI, "Case Study," where I pull many of the important elements together from this book and apply them to an example company's VPN implementation.

PREV NEXT

Part VI: Case Study

PREV NEXT

Chapter 24. Case Study

This chapter is different from the rest of the chapters in this book. From the beginning of this book, each chapter has focused on VPNs and various VPN implementation types, and how these are configured on Cisco VPN 3000 concentrators, software and hardware clients, IOS-based routers, and PIX and ASA security appliances. Throughout the book I've tried to include many configuration examples, illustrating common scenarios and possible problems you might face while implementing VPNs.

This chapter doesn't introduce any new material. Instead, this chapter will focus on the use of the concepts and features discussed in this book and implement many of them in a case study environment. In other words, I'll create a fictitious company with various Cisco VPN-capable products and bring together much of the VPN knowledge I've discussed so far and apply it to my fictitious company. Using a case study, you can see more easily where certain VPN solutions make more sense than other solutions.

Throughout this case study, I'll discuss why I'm using certain VPN implementations over others and, within a certain product, why I'm using one particular feature instead of another. The configurations I'll put together focus primarily on the implementation of VPNs. I'll discuss other non-VPN items that are important to the design, but in most cases the configuration of those items is beyond the scope of this book.

Company Profile

The network shown in Figures 24-1 and 24-2 illustrates this case study. The company in this network has a corporate office, a handful of regional sites, a few dozen branch offices, and hundreds of remote access users that connect via the Internet. The following sections will discuss the necessary requirements for these components.

Figure 24-1. Company Internet Connections

[View full size image]

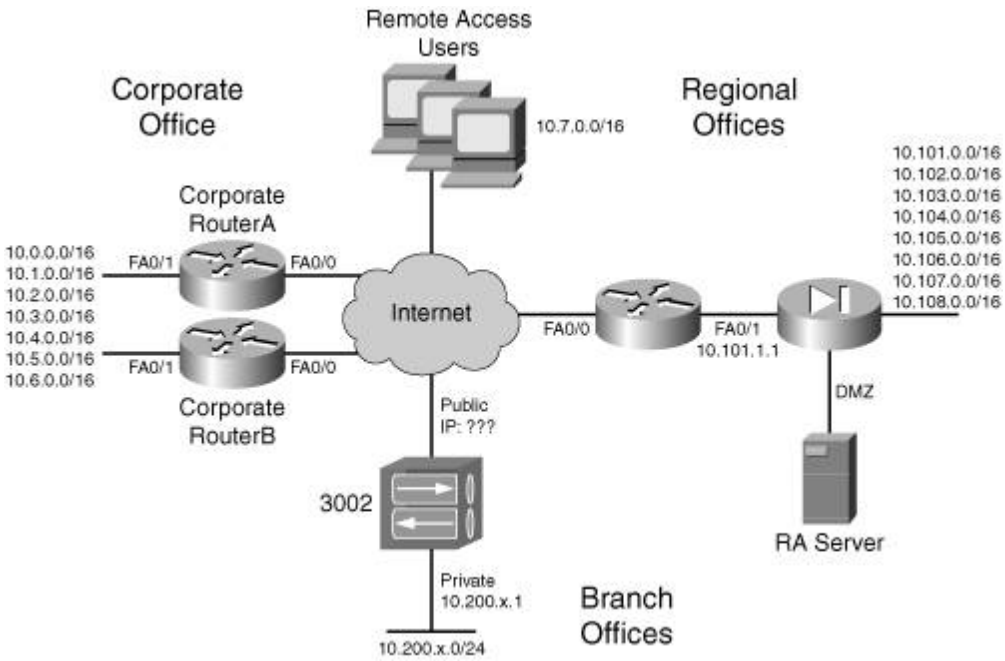
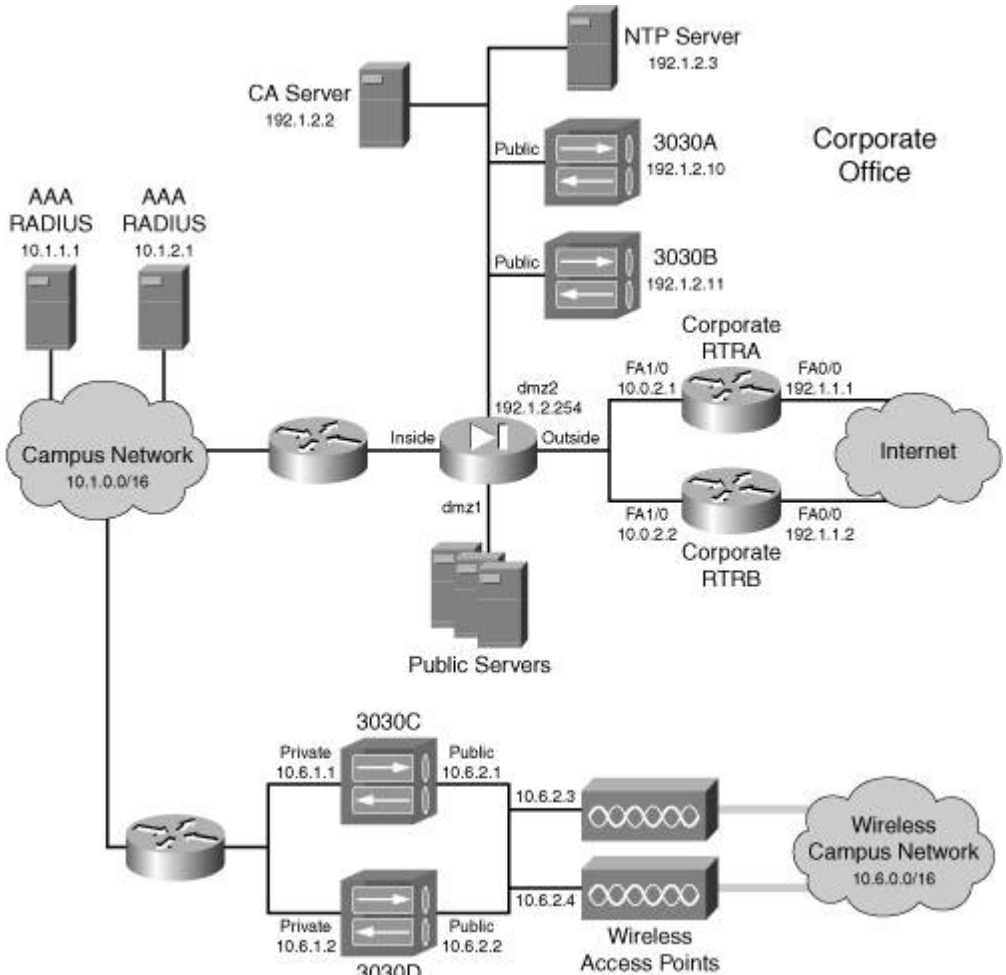


Figure 24-2. Company Campus Network

[View full size image]



Case Study Configuration

Now that I've defined the policies for this network, I'll explain the configuration for the following VPN-enabled devices:

- DMVPN routers
- VPN 3030 concentrators
- VPN 3002 hardware clients
- Cisco VPN software clients

I've broken up the configuration explanation into the following sections:

- **Perimeter router configuration** Discusses the DMVPN configuration on both the dual corporate routers and the regional routers.
- **Internet remote access configuration** Discusses the configuration of the DMZ2 concentrators, the 3002 hardware clients, and the Cisco VPN Client for Internet users.
- **Main campus wireless configuration** Discusses the configuration of the campus concentrators for wireless users and the setup of the Cisco VPN Client software for the wireless users.

Perimeter Router Configuration

A dual-DMVPN design with dual hubs is used to interconnect the corporate office and regional sites via IPsec site-to-site sessions. Certificates are used for device authentication. The following sections will discuss the configuration of each device.

Basic VPN Configurations on the Routers

All of the routers will require certain commands that will be similar in their configuration. These commands will perform the following:

- Allowing VPN traffic
- Enabling ISAKMP
- Defining ISAKMP policies
- Configuring DPD
- Obtaining and using certificates
- Defining IPsec transform sets and profiles

This section will discuss the configurations that are common to all the perimeter routers. Example 24-1 shows this base configuration:

Example 24-1. Base Perimeter Router Configuration

```
perim(config)# access-list 100 remark insert other ACLs here
perim(config)# access-list 100 permit udp any any eq 500
perim(config)# access-list 100 permit udp any any eq 4500
perim(config)# access-list 100 permit esp any any
perim(config)# access-list 100 permit tcp any host 192.1.2.2 eq 80
perim(config)# access-list 100 permit tcp any host 192.1.2.3 eq 123
perim(config)# access-list 100 permit udp any host 192.1.2.3 eq 123
perim(config)# access-list 100 remark insert other ACLs here
perim(config)# interface fastethernet0/0
perim(config-if)# ip access-group 100 in
perim(config-if)# exit
perim(config)# ntp server 192.1.2.3 key 99 source fastethernet0/1
perim(config)# ntp authenticate
perim(config)# ntp authentication-key 99 md5 55ab8971F
```

Summary

This case study showed you the use of the many VPN features discussed in this book. This is by no means the only possible solution to the given scenario; however, the goal of this chapter was to show you how to implement Cisco VPN features in an integrated fashion. As you will learn, each networking situation has its own set of unique problems and issues and its own set of possible solutions.

A Last Bit of Advice

As you can see from this case study, even though the network is medium-sized, the configuration is quite involved. Therefore, I would highly recommend that you implement one VPN component at a time, making the setup and troubleshooting process easier. For example, I would probably set up the L2L DMVPN configuration first, and then focus on the remote access. And as a final note in this book, VPNs are probably one of the most complicated technologies you'll deal with in the security field. I've learned a long time ago that I don't know everything and am always being introduced to problems I've never seen before. Therefore, having a very good foundation in the understanding of VPNs is very important (I can't stress this last point enough), which is why the first part of the book was devoted to VPN technologies and why each part of the book contained a troubleshooting chapter. With a good foundation, you should be able to design a solid, secure VPN solution and troubleshoot any problems that may crop up. And last, good luck with your VPN endeavors! Cheers!

Index

PREV NEXT

Index

PREV NEXT

Index

Index

Index

[SYMBOL] [A] [B] [C] [D] [E] [F] [G] [H] [I] [J] [K] [L] [M] [N] [O] [P] [Q] [R] [S] [T] [U] [V] [W] [X]

Index

[SYMBOL] [A] [B] [C] [D] [E] [F] [G] [H] [I] [J] [K] [L] [M] [N] [O] [P] [Q] [R] [S] [T] [U] [V] [W] [X]

Index

Index

Index

Index

Index

Index

PREV NEXT

Index

Index

Index

[SYMBOL] [A] [B] [C] [D] [E] [F] [G] [H] [I] [J] [K] [L] [M] [N] [O] [P] [Q] [R] [S] [T] [U] [V] [W] [X]

Index

Index

Index

Index

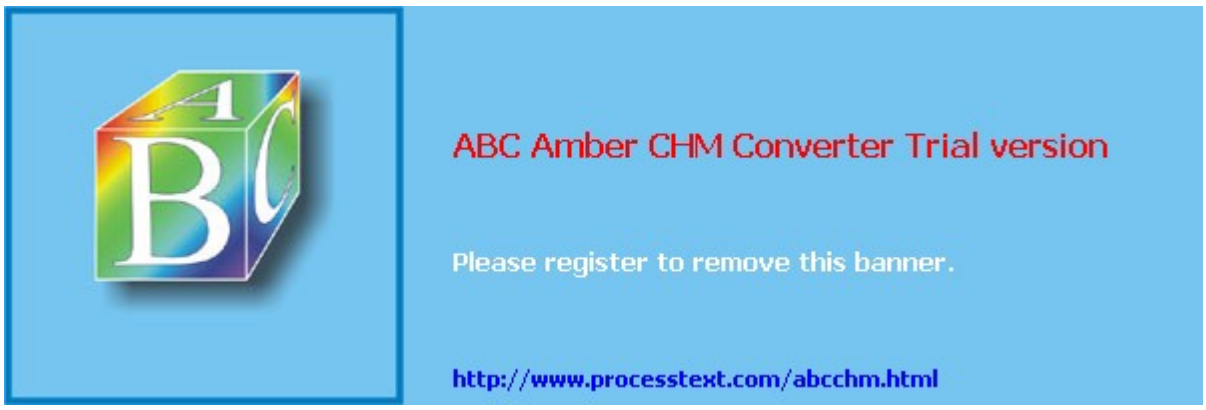
PREV NEXT

Index

Index

Index

Index



Index

[SYMBOL] [A] [B] [C] [D] [E] [F] [G] [H] [I] [J] [K] [L] [M] [N] [O] [P] [Q] [R] [S] [T] [U] [V] [W] [X]

Index

Index

PREV