THE EXPERT'S VOICE® IN CISCO

Cisco Routers for the Small Business

A Practical Guide for IT Professionals

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

Jason C. Neumann

Apress®

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

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

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

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

Lead Editor: Jonathan Gennick
Technical Reviewers: Dean Olsen, Sebastien Michelet
Editorial Board: Clay Andres, Steve Anglin, Mark Beckner, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Michelle Lowman, Matthew Moodie, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Project Manager: Sofia Marchant
Copy Editor: Octal Publishing, Inc.
Associate Production Director: Kari Brooks-Copony
Production Editor: Kari Brooks-Copony
Compositor: Pat Christenson
Proofreader: Katie Stence
Indexer: Broccoli Information Management
Artist: April Milne
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski

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

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

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

Contents

About the Author

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

About the Technical Reviewers

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

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

CHAPTER 4

Configuring a VPN Using IPSec

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

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

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

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

Preparing Your Sites

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

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

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

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

Setting Up the VPN

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

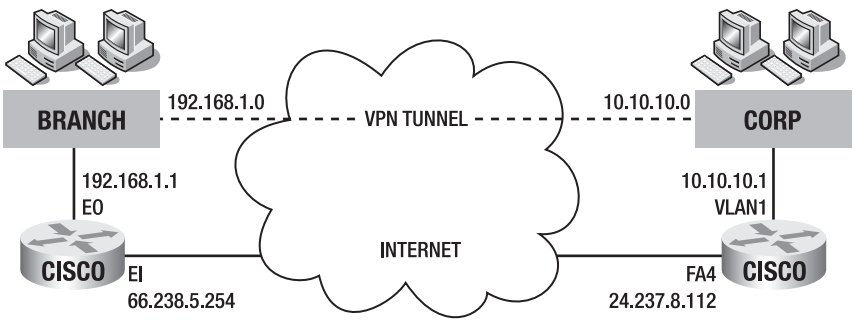

Figure 4-1. *VPN diagram*

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

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

Step 1: Create a VPN-Friendly ACL for NAT

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

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

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

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

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

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

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

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

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

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

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

Step 3: Apply Your VPN Routing Policy to NAT

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

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

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

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

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

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

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

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

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

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

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

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

Configuring IKE Phase 1

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

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

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

Step 1: Create a Key Exchange Policy

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

Step 2: Define the Encryption Type

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

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

Step 3: Define a Cryptographic Hash Function

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

Step 4: Define Your IKE Key Type

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

Step 5: Define Your IKE Key Size

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

DIFFIE-HELLMAN KEY PROTOCOL

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

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

Step 6: Create a Preshared Key

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

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

Listing 4-6. *How to Create a Preshared Key*

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

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

Next, the preshared key is associated with the static IP address of the Corporate router at address "24.237.8.112". This is the static IP address of the participating router for this VPN connection.

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

Configuring IPSec Phase 2

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

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

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

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

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

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

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

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

Step 1: Create a VPN-ACL

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

Step 2: Create a Transform Set

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

Step 3: Create a Crypto Map

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

Step 4: Set the VPN Peer

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

Step 5: Set the Transform Set

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

Step 6: Set the PFS Group

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

Step 7: Apply Your VPN ACL

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

Step 8: Apply the Crypto Map

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

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

Modifying Your IPFW-ACL

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

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

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

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

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

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

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

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

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

Verifying Your VPN Connection

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

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

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

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

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

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

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

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

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

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


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

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


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

     inbound ah sas:

     inbound pcp sas:

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

outbound ah sas:

outbound pcp sas:
```

Troubleshooting

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

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

General Network Settings

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

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

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

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

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

IKE Phase 1 Settings

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

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

IPSec Phase 2 Settings

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

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

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

When in Doubt, Print It Out

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

Summary

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

Set Up the VPN

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

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

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

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

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

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

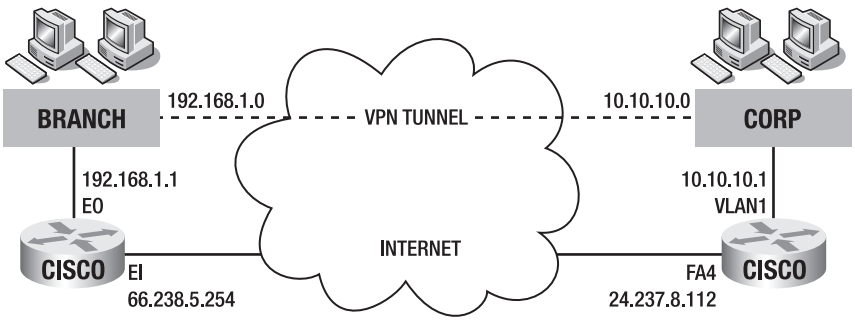

Figure 4-2. *Example VPN configuration*

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

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

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

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

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

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

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

Branch Office VPN Configuration

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

LAN Address: 192.168.1.0 on Interface E0

WAN Address: 66.238.5.254 on Interface E1

Use the following commands to create this branch office configuration:

```
site1> enable
site1# config t

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Corporate Office VPN Configuration

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

LAN Address: 10.10.10.0 on Interface VLAN1

WAN Address: 24.237.8.112 on Interface FA4

The following are the commands to carry out that configuration:

```
corp> enable
corp# config t

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

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

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

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

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

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

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

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

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

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

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


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

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

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

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

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

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

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

Troubleshoot Your VPN

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

General Network Settings

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

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

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

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

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

IKE Phase 1 Settings

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

IPSec Phase 2 Settings

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

When in Doubt, Print It Out

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