Products in the Edge

Terms you'll need to understand:

- ✓ RFC 2827 filtering
- ✓ Unicast RPF
- ✓ NAT
- ✓ PAT
- ✓ **overload**
- ✓ IKE
- ✓ ISAKMP
- ✓ DH group

Techniques you'll need to master:

- ✓ Configuring VPN termination and NAT on a router
- ✓ Configuring VPN termination and NAT on a PIX
- ✓ Configuring a VPN concentrator from the VMS
- ✓ Configuring a VPN client from the VMS

Because the edge is the part (or parts) of your network that faces the outside world, security here is both more complex and more rigorous. Although most attacks actually come from insiders (at least, according to some statistics), you have a considerable degree of control over what protocols and applications are inside the campus. In the outside world, however, with which your edge modules interconnect, there can be anything and (probably) everything. The security job in the edge is bigger and harder.

Products used in the edge include routers, switches, NIDS and HIDS, routers with a software firewall (the IOS Firewall feature set), dedicated firewall appliances (such as the PIX firewall), and VPN devices: the VPN concentrators and VPN clients (at the remote locations). We covered most of what you need to know about the first four items (routers, switches, NIDS, and HIDS) in Chapter 8, "Products in the Campus." A few additional items need to be configured on routers in the edge, so we start with that. From there, we cover the firewall capability in the IOS Firewall feature set, the PIX firewall, the VPN concentrator series, the VPN clients, and how the VPN devices are managed via the VMS.

We specifically address how to secure your communications at the edge on various devices, including a router, a PIX, a VPN concentrator, and a VPN client. You will set these up in pairs so that you can see the same VPN from two product perspectives at once. In the case of the router and PIX, it is done via the CLI. In the case of the VPN concentrator and VPN client, it is a matter of setting complementary parameters in the VMS—the CiscoWorks VPN/Security Management Solution, a GUI-based manager.

Routers Redux

The additional configuration required of routers in the edge especially applies to the perimeter routers, which directly connect to another organization (typically an ISP). They should have two types of filtering installed, to minimize the amount of illegitimate traffic that gets through them into your network. The two types are RFC 2827 filtering and Unicast RPF.

RFC 2827 Filtering

A favorite method of attackers who seek to penetrate a network is to pretend that they belong to it: They send packets with a forged source IP address from inside the network. However, if you think for even a moment about traffic flows, there is no reason for traffic with an internal source address to enter your network from an outside interface (unless you somehow looped the traffic, but with an internal source and destination address, why would

you loop it outside?). However, it could be legitimate traffic if you have multiple locations with a large address block. Therefore, be sure to implement this as filtering the address blocks that are *behind* the router and allowing corporate addresses that are not internal to this network to pass (at least, to pass on to other validation, such as your firewall).

RFC 2827 was originally written to be implemented by ISPs, which, in fact, have seldom implemented it. Written as "do not accept traffic from your customers with anything other than your network as a source address," it has been adapted informally to mean (for receivers of Internet traffic) "do not accept anything from your upstream that should not come from that direction."

Thus, in addition to filtering out any incoming traffic with a source IP address from your internal block, you should filter addresses that should never be attempting ingress from the outside world, the RFC 1918 private address space blocks (10/8, 172.16/12, and 192.168/16). Hopefully, your ISP is also performing such filtering, but you should implement this with an ACL just the same—you cannot guarantee that the ISP's filters will always be present or that they will be set for the addresses that you believe should be filtered.

In the interest of conserving bandwidth as well as security concerns and good citizenship, you should implement similar filters on egress from your network as well. There is no reason for you to send forth traffic with private IP addresses as their source: Even if you use such addresses internally, they should be NATed to a public address before they get to your perimeter router. Likewise, traffic should never pass out of your network with any source IP address except one from your address block; any traffic with a different address in the source field should be dropped, *and these drops should be logged* (so that you know you have a problem inside your network). A simple example is shown in Figure 9.1 (the figure uses a private address space, but the NAT is not shown, to minimize the clutter).

Most lines of the filters are straightforward to read; the only odd one is **deny 172.16.0.0 0.15.255.255**—the second octet of the wildcard mask might seem strange. Remember that the private address space in question is 172.16.0.0 to 172.31.255.255. This can be expressed as 172.16.0.0/12 (or 172.16/12 in shorthand). Twelve bits of network address matching leaves 20 bits of wildcard, or 16 bits for the last two octets and the rightmost 4 bits of the second octet: 00000000 00001111 11111111 11111111. Focusing on the second octet, you can see that its decimal value is 15. Using decimal subtractions properly also works (in this case, 31 – 16), but it must be done carefully; many professionals set up the problem in binary to be sure that they get it right—especially on production networks.

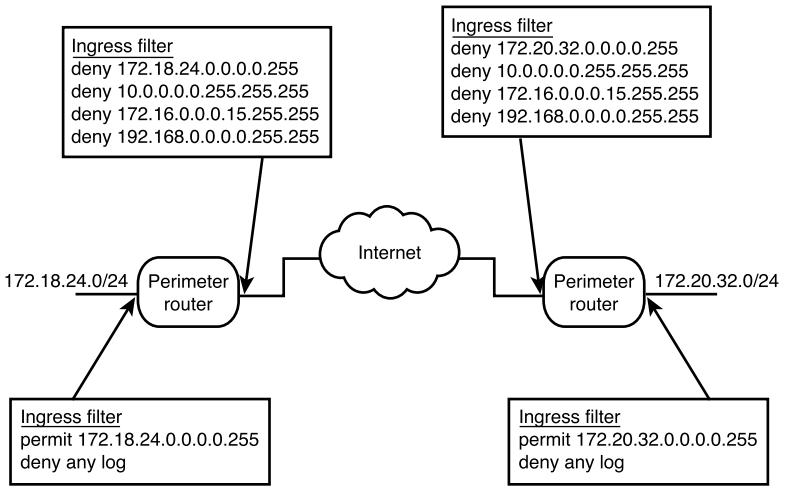

Figure 9.1 Perimeter router filtering per RFC 2827.

Unicast RPF

As with access lists, Unicast RPF consumes a few CPU cycles, but the payoff is that less bogus traffic enters your network. A hacker might be smart enough not to forge one of your internal addresses or to use a private address; instead, he might forge an address from a legitimate host somewhere else in the world. Unicast RPF checks all packets arriving on an interface first for whether such an address exists in the routing table (do I know how to reach this address?) and, second, if so, whether traffic from it should arrive on the ingress interface (does the return route exit via that interface?). If the answer to the first question is no, the packet is dropped. If the answer to the first question is yes but the answer to the second question is no, the packet is dropped. Figure 9.2 illustrates a simple example.

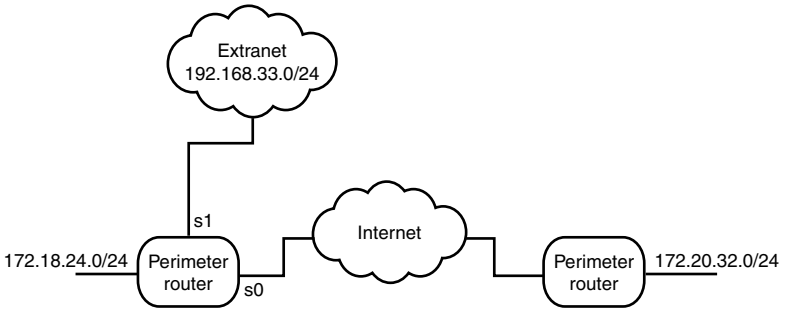

Figure 9.2 Unicast RPF example.

In Figure 9.2, suppose that a packet arrived on interface s0 with a source address of 192.168.33.125. It would pass muster on the first question but not the second: It should have arrived on port s1, with that as the source address.

In fact, this is a simplification of Unicast RPF. It relies on Cisco Express Forwarding (CEF) because it uses the Forwarding Information Base (FIB) created by CEF. It also examines the packet to see if it arrived via the best interface for that address (if there are multiple possibilities). If you want to log the drops caused by Unicast RPF, you must use an ACL in conjunction with it; using an ACL also enables you to allow such traffic to pass despite the ingress interface mismatch (a **permit** statement allows the traffic in, while a **deny** statement drops it).

Nonperimeter Routers in the Edge

You might also have nonperimeter routers in the edge, perhaps serving as multipurpose devices: routing, firewalling, and/or terminating VPNs. We need to address how to configure a router for those tasks (except routing, which is not in the purview of the CSI exam).

A router running an IOS image that contains the Firewall Feature Set (FFS) can serve as a fully adequate firewall, depending on the volume of traffic to be controlled, of course. After all, a firewall inspects traffic, compares it to a set of rules, and either allows the traffic to pass or discards it: It operates like an access list. One difference, however, is that although a firewall typically allows free passage to outgoing traffic—and responses to known outgoing traffic—its fundamental stance on incoming traffic is to deny, while a router's fundamental stance is to permit; this is modified only by the ACL. Thus, you could call the firewall a strict access list, although it has the capability to use a few other things, such as NAT and secure tunnels (IPSec). The FFS provides a set of features to do all of what firewalls do:

- Standard ACLs and static extended ACLs
- Lock-and-key (dynamic ACLs)
- Reflexive ACLs
- TCP intercept
- Context-based access control (CBAC)
- Cisco IOS Firewall IDS
- Authentication proxy
- Port-to-application mapping (PAM)

- Security server support
- NAT
- IPSec
- Neighbor router authentication
- Event logging
- User authentication and authorization

With these capabilities, why would anyone need a PIX or any other dedicated appliance? Because the router does all this in software, while the dedicated appliance offloads the heavy processing to specialized ASICs, which can do the same job much faster in hardware. The dedicated appliance, such as a PIX firewall, can handle many more simultaneous processes and connections. As long as the volume of traffic to be firewalled is not too great, a router with the FFS can do the job adequately.

Many of these FFS features were covered in the SECUR exam; for this exam, you must focus on the capability to perform NAT and terminate IPSec tunnels. Remember the placement of the router with firewall in the SMR Blueprint: It is often the choice for terminating tunnels and might easily hold the only public address for a small network. That's why NAT and IPSec tunnel termination are both important. Let's look at an example in which a router needs to perform NAT and terminate an IPSec tunnel; later, we'll use the other end of this example for the same tasks on a PIX firewall. This situation is shown in Figure 9.3.

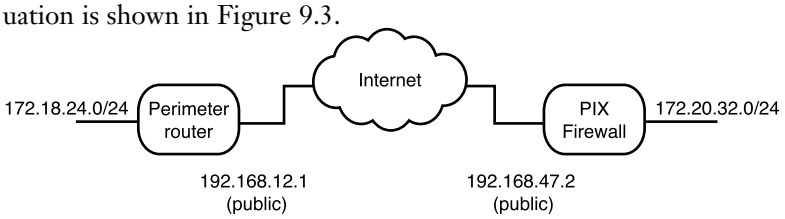

Figure 9.3 Router-to-PIX tunnel with NAT on each end.

We will use the 192. 168.*x.x* addresses as public addresses, to which all others inside the LAN segments behind the router or PIX must be translated. As a worst case, we will also assume that only one public address can be used for each LAN segment.

NAT on the Router

In this example, multiple addresses in the 172.18.24.0/24 network must all be NATed to one public address, 192.168.12.1. When NATing on a Cisco

device, you have local addresses (not routable outside the local network) and global addresses (routable outside the local network). From another point of view, you have inside addresses (addresses as they appear from inside the network) and outside addresses (addresses as they appear from outside the local network). Thus, an inside local address is the address that this host is known by *only* inside the network. The outside global address is a publicly known, publicly reachable address.

A device can be known by one address from inside the network but another from the outside (its global address). This offers a bit of protection from an attack because the attacker sends traffic to, for instance, a Web server's outside global address, and the traffic must pass through a NAT device that translates the destination address to an inside local address. The NAT device—which is firewalling—can inspect the traffic at this time. You are simply making direct attacks more difficult.

In this example, a host can have the inside local address of 172.18.24.42, but its inside global address is 192.168.12.*x*, where *x* comes from a pool assigned to the perimeter router. The translation between the two addresses is NAT. It can be a static NAT (a fixed mapping) or dynamic NAT (in which the inside global address comes from a pool and is recycled). Because one of the reasons for using NAT and private IP addresses is address conservation, assume that this perimeter router has available only one inside global address (192.168.12.3) for use by NAT. The 192.168.12.1 address is for the router interface, not other traffic; other traffic uses the 192.168.12.3 address, which is assigned only to the NAT functionality.

To configure a router to provide NAT of multiple internal addresses to fewer external addresses, you must use port address translation (PAT), which the **overload** command invokes. You can use PAT with as few as one external address. Four steps are involved in creating a PAT configuration; although they need not be done in a given order, the following is a logical order:

1. Create a standard access list to designate hosts whose inside local addresses will be translated to inside global addresses.
2. Create a NAT pool and assign the access list as the source list for the pool (if there is only one address, you list it twice, as the start point and the endpoint of the pool).
3. Assign the IP address and mask to an interface, and then designate it as the **nat inside** or **nat outside** interface (as appropriate).
4. Assign the IP address and mask to the other interface, and designate it as the other NAT interface.

In the previous example, interface e0 on the perimeter router was 172.18.24.1, and it should be the internal (LAN) interface (so that is its inside local address). Interface s0 (WAN) is 192.168.12.1, which is an outside global address, and the pool of addresses (inside global addresses) is 192.168.12.3, a "pool" of one address. Here are the commands:

1. **access-list 13 permit 172.18.24.0 0.0.0.255**
2. **ip nat pool ch9_example 192.168.12.3 192.168.12.3 netmask 255.255.255.0**

   ip nat inside source list 13 pool ch9_example overload
3. **interface e0**

   ip address 172.18.24.1 255.255.255.0

   ip nat inside
4. **interface s0**

   ip address 192.168.12.1 255.255.255.0

   ip nat outside

It really is that simple to set up NAT on the router: Create a source list, and then create a NAT pool, a NAT inside interface, and a NAT outside interface. You might have already assigned IP addresses to the interfaces, so that little bit as shown might be redundant. The key is linking the pieces with the **ip nat** command—there is only one NAT process on the router, so all entries of that command link to the same thing.

IPSec on the Router

In a large organization, it might prove worthwhile to set up a Certificate Authority and generate and distribute certificates. However, for small and midsize businesses, it is not practical. That means that IPSec will be set up with preshared keys. Obviously, the key to be used on any given IPSec tunnel must be the same at both ends, and it must not be compromised (protect it at least as well as you want the information it encrypts to be protected). Another point to be aware of is that when you set up your access lists to establish the traffic of interest for the IPSec tunnel (the crypto access lists), those must be the same on each end of the tunnel (they must refer to the

same address sets, protocols, and so on). If they do not match, the tunnel will sometimes work and sometimes fail, which is difficult to troubleshoot.

Establishing an IPSec tunnel requires two basic processes, establishing peering (using the Internet Key Exchange, IKE) and then establishing a Security Association (SA). Although the terms *IKE* and *ISAKMP* (the Internet Security Association and Key Management Protocol) are often used interchangeably, they are not the same. The short version of their relationship is that the IKE process uses the ISAKMP-mediated information exchange. Because you will use ISAKMP instead of a manually specified SA, you must either use certificates or preshared keys at the beginning, during the IKE phase (peer establishment) of creating a tunnel. Going through both the IKE and SA creation processes, six steps are required to set up the IPSec tunnel termination on a router; again, although there is no required order, this is a logical one:

1. Create a crypto isakmp policy.
2. Create a key and assign the key to a peer address.
3. Create an access list to designate the traffic to be encrypted.
4. Create a crypto ipsec transform set.
5. Create a crypto map.
6. Apply the crypto map to the outgoing interface.

Items 1 and 2 in this list establish your IKE policy; the rest are all IPSec configuration.

Everyone who takes an exam feels stress. Under stress, we do things that we would not ordinarily do and make mistakes that we would not ordinarily make. Perhaps it's stress, or perhaps it's the dreadfully slow response of the systems at the testing center most convenient to me, but during simulation exercises in an exam, I always forget **config t**—until my first two or three (yes, the systems are that slow) configuration commands simply fail.

You can expect at least one simulation in any professional-level Cisco exam. If you have done this already, arrange your mental checklist to put the items that you are most likely to forget first, if possible. In the case of these two examples, creating the access list is the unusual command (not very much like the others). I suggest putting it first in the IPSec configuration, lest you forget to include it under stress.

In the case of our example, suppose that you wanted to specify the hash and a smaller DH initialization. You would input something like this:

1. **crypto isakmp policy 13**

   hash md5

   group 1

   authentication pre-share
2. **crypto isakmp key nitT4agM#0C2%5 address 192.168.47.2**
3. **access list 113 permit 172.18.24.0 0.0.0.255 172.20.32.0 0.0.0.255**
4. **crypto ipsec transform-set example esp-des esp-md5-hmac**
5. **crypto map ch9_example 10 ipsec-isakmp**

   set peer 192.168.47.2

   set transform-set example

   match address 113
6. **interface s0**

   crypto map ch9_example

If you had to do NAT and create an IPSec tunnel for the same traffic, you could easily combine the interface-assignment commands, and you might want to use only one access list for both purposes. In that case, you would have a total of eight steps.

Warning: If you have a tunnel exchange with a PIX firewall (as shown in Figure 9.3), the PIX and the router do not share the same default SA lifetime. You can change the router's default lifetime with **crypto ipsec security-association lifetime {***number_of_seconds***}** in global configuration mode. You can also change the default on the PIX.

If you have a router simulation on the exam, don't forget to save your config. At the time this is written, the classic **copy running-configuration startup-configuration** (and its abbreviated forms) is still accepted, although the new format of **copy system:running-configuration nvram:startup-configuration** should work as well.

The PIX Firewall

When you think about the role of the PIX firewall in an SMR example, it might be segregating traffic inside the edge—creating a DMZ, for

instance—and it might be terminating IPSec tunnels (quite possibly with NAT), which dovetails with the previous example of the perimeter router. We cover the basic traffic segregation first and then discuss configuring the PIX for NAT and tunnel termination.

Traffic Segregation

The PIX (originally Private Internet Exchange) firewall uses the Adaptive Security Algorithm (ASA) to manage traffic passing through it. Outbound connections (from a higher to a lower security level) are ordinarily allowed unless they are specifically denied. The stateful traffic monitoring allows response traffic to enter. However, unsolicited traffic from a lower to a higher security level is denied unless specifically permitted. The security level is set on a per-interface basis, with a range of 0–100 (typical settings are 0 for the outside, 50 for the DMZ, and 100 for the inside interfaces).

Specifically permitting incoming traffic can be done in two ways: with conduits and static mappings (old), or with access lists (new, beginning with PIX release 5.0.(1)). ACLs are now preferred over conduits (static address mappings—usually just called statics—are required with conduits because the outside host must have a reliable address to seek out). Conduits grant access based on the same basic kind of criteria as ACLs (source and destination address, destination port, and protocol). The latter, of course, offers many refinements (such as time-of-day access) that conduits do not. Although both types of expression are valid in configuring a PIX, expect to see ACLs favored in Cisco documentation as part of the more extensive use of IOS syntax. *However*, PIX ACLs do not use wildcard masks (as IOS ACLs do); they use network masks.

NAT on a PIX

To configure NAT on the PIX, you must designate the inside addresses to be NATed and the global address pool (or single address for PAT) to be used. To use the situation in Figure 9.3 again, assuming that you have only one outbound IP address (global address) of 192.168.47.3 (192.168.47.2 is the interface address only) and you want to allow all inside users outbound access, your NAT commands would look like this:

1. **nat (inside) 1 172.20.32.0 255.255.255.0**
2. **global (outside) 1 192.168.47.3**

This is very simple and easy; the NAT ID (1, in this case) ties together the two commands. You could have been lazy, and used **0 0** in the first command

to specify all hosts, but suppose that someone compromised a host on this network, and it was sending traffic with spoofed source IP addresses. It is better to be safe than sorry: Specify the inside hosts that are to be NATed.

It is quite possible that the PIX end of the connection might be the larger network and would have multiple global addresses available. In that case, the second command would change to **global (outside) 1 192.168.47.3-192.168.47.30 netmask 255.255.255.0**. That covers the basics of NAT on a PIX firewall; it is discussed in more detail, of course, on the Advanced PIX Firewall Configuration Exam.

IPSec on a PIX

Configuring IPSec on a PIX is very similar to configuring it on a router, with just enough differences to keep you alert. There are two basic processes to configure, ISAKMP and IPSec. To take them in the same order used on the router, three steps are involved for the ISAKMP process:

1. Enable ISAKMP.
2. Create the key to be used with a given peer address.
3. Create the ISAKMP policy.

For IPSec configuration, there are six steps (two of which are optional but recommended), again not in a required order; they are listed here in a useful order:

1. Create the crypto access list (define the traffic to be encrypted).
2. Allow incoming IPSec traffic to bypass the ACL or conduit checks (if desired—you are deciding to implicitly trust all traffic arriving over the IPSec tunnel).
3. Create the transform set.
4. Set the SA lifetime in seconds (if you do not want to use the default of 28,800 seconds—recall that the router's default IPSec SA lifetime is 3600 seconds, so 1 hour might be a better choice than 8 hours if security is a strong concern).
5. Create the crypto map.
6. Apply the crypto map to the appropriate interface.

For this example, to match the IKE configuration created on the perimeter router, you would use this:

1. **isakmp enable outside**
2. **isakmp key nitT4agM#0C2%5 address 192.168.12.1 netmask 255.255.255.255**
3. **isakmp policy 13 authentication pre-share**

   isakmp policy 13 hash md5

   isakmp policy 13 group 1

Now, to match the IPSec configuration:

1. **access-list ch9 permit ip 172.20.32.0 255.255.255.0 172.18.24.0 255.255.255.0**
2. **sysopt connection permit-ipsec**
3. **crypto ipsec transform-set example esp-des esp-md5-hmac**
4. **crypto ipsec security-association lifetime 3600**
5. **crypto map ch9_example 13 ipsec-isakmp**

   crypto map ch9_example 13 match address ch9

   crypto map ch9_example 13 set peer 192.168.12.1

   crypto map ch9_example 13 set transform-set example
6. **crypto map ch9_example interface outside**

Note that the last command does not include the crypto-map ID; that's an easy mistake to make. Also, saving your configuration on the PIX is not the same as in the IOS: Use **write memory** instead (or **configure memory** if you want to merge the new configuration with the existing one).

If you will terminate a number of tunnels on a PIX, you might want to consider using the VPN Accelerator Card, which offloads the VPN functions from the main system. This card has been supported since PIX release 5.3(1) (with a DES or 3DES license). If your software is older than that, consider upgrading it to use the new hardware.

Configuring the PIX to be the other end of the example is a little more tedious than it was on the router. Setting up VPNs using the VPN concentrator and VPN client is rather different.

The VPN Concentrator

Cisco has a wide range of capabilities in the VPN Concentrator 3000 series of devices. They can support from 100 simultaneous users (on the 3005) to 10,000 (on the 3080), with a commensurate range of hardware on board. The concentrator is primarily managed through the VPN Concentrator Manager (a GUI-based manager running in a browser—you need only be on the same private network as the concentrator). You can also manage it through a CLI. Browser management can be via HTTP or HTTPS (after installing the certificate—remember that HTTPS uses SSL for authentication and encryption).

A major advantage of the VPN concentrator compared to a PIX to terminate many tunnels is that the former handles connections between tunnel clients (such as branch offices connecting to each other through the corporate system rather than directly). The concentrator also began offering AES before the PIX (with the new VAC+ and PIX OS 6.3(1), the PIX now offers 128-, 192-, and 256-bit AES). Without the recent software, however, the PIX is limited to DES, 3DES, and ESP-NULL (no payload encryption). The concentrator offers the following:

- DES
- 3DES
- ESP-NULL
- AES-128, -192, and -256

As you learned when configuring tunnels with a router and a PIX, IKE must be completed before the SA is established. Thus, it makes sense that, on a VPN concentrator, you should configure the IKE proposal parameters before configuring the rest of your IPSec parameters. All parameters are configurable via the GUI.

The VPN concentrator can act as the initiator of a connection or as a responder to another device's request in a site-to-site connection, which is also known as a LAN-to-LAN connection. However, for a connection to an IPSec client, the VPN concentrator can act as only a responder; it cannot initiate the connection.

Whether acting as an initiator or a responder, the concentrator offers its configured IKE proposals, the other party does the same, and they agree on a common configuration. The VPN concentrator offers more possibilities than the IOS or PIX: In addition to adding AES to the DES and 3DES

encryptions, the concentrator supports DH Group 7 (for use with the movianVPN client and others capable of handling Elliptic Curve Cryptography, ECC).

Between preconfigured proposals and those that you have created (custom proposals), the concentrator can handle up to 150 IKE proposals; any number of them can be active (usable) at a given time. When IKE is complete, the SA is established according to the parameters that you entered in that portion of the GUI. If your IPSec connection must pass through intervening devices that might filter it out, you should add the configuration for NAT transparency. The concentrator supports this, with a customizable higher-numbered TCP port (so that you can configure any intervening firewalls or ACLs to pass traffic on your specified port).

To create your active proposal list, navigate through the GUI to Configuration | System | Tunneling Protocols | IPSec | IKE Proposals. There you can move proposals between the active and inactive lists. You can also use the Add, Modify, and Copy buttons to create custom proposals (for instance, copy an existing proposal and then modify it). You can also delete proposals from either list, but delete with caution—if no SA is using this proposal, it is *gone*, without confirmation or undo capability.

If you want to use NAT transparency, you can invoke that by selecting Configuration | System | Tunneling Protocols | IPSec | NAT Transparency. Here you turn it on and specify the TCP port to use.

Because you likely will be using the VPN concentrator when you have many VPNs, it becomes important to be able to manage updates to the many clients. You manage this via Configuration | System | Client Update, and click on Enable to access the check box to turn it on. The concentrator can push the updates to clients, although there are differences in how this works between the software and hardware clients. With the software client, the concentrator sends an IKE packet upon connection specifying the acceptable versions of the client software. The message also contains the location from which the client can obtain an update, which the client's administrator can retrieve and install. With the hardware client, an IKE packet is sent containing a list of acceptable software and firmware versions. If the hardware client is not running an acceptable version, it is automatically updated via TFTP and reboots when the update is complete.

There is much more than this to the VPN concentrator, of course, but that is covered in the CSVPN exam.

The VPN Client

Cisco offers two types of VPN client: hardware and software. The hardware client is quite well suited to supporting a small branch office as well as individual users. The software client is especially useful for hosts (such as laptops) that might not always be connecting from a protected environment. We talk about each in turn.

The 3002 VPN Hardware Client

As with the concentrator, the 3002 VPN Hardware Client is primarily administered via a Web browser using HTTP or HTTPS. It not only offers the branch secured communications upstream, but it also can act as the local DHCP server (although a rather simple one, with only one scope and no exceptions).

The hardware client is specifically designed to interface with the Cisco 3000 series VPN concentrator, but it also works well with the PIX, IOS, and third-party IPSec devices. The 3002 acts as the initiator in all tunnels with the concentrator; its encryption, authentication, extended authentication, and mode-configuration capabilities line up with those of the concentrator. Therefore, when the hardware client initiates a tunnel, it offers its IKE proposals, and it is quite likely that the concentrator will either accept or offer counterproposals that the 3002 can accept.

In the VMS GUI, navigate through to Configuration | System | Tunneling Protocols | IPSec. Here you can designate the following:

- The remote peers (a primary Remote Easy VPN Server and Backup Remote Easy VPN Servers)
- IPSec over TCP, and the port (if desired)
- A certificate (if desired)
- The group name and password, and the username and password

The IKE proposals themselves are preconfigured on the 3002 Hardware Client.

The address of the public interface to which the VPN client connects (at the far end) is referred to in the 3002 documentation as the Easy VPN Server. This is actually a software package available for the IOS (1700, 7100, 7200), PIX, and 3000 series VPN concentrators. The Easy VPN Client is available for low-end routers, the PIX 501, and VPN clients. The Easy VPN Solution is not limited to the VPN-specific devices.

After an IPSec connection is created, users can access resources at the far end as though they were located in the LAN—that is, according to whatever permissions and authorizations have been set.

The VPN Software Client

The Cisco VPN Software Client is available for Windows (both 9*x* and NT/2K/XP OS), Mac OS X, and Linux and Solaris. Details of the GUI differ from one host OS to another, of course, and the configuration steps are user-oriented rather than administrator-oriented. The software client acts like the hardware client in terms of connecting to the headend of the tunnel. It initiates the connection, offering its preset IKE proposals. An acceptable parameter set is agreed upon (or the connection fails, of course), and the client communicates with the headend as though locally present (always subject to bandwidth limitations or connectivity problems between the two physical locations).

The software client is capable of handling tunnels over any of the following connections:

- POTS (dialup service)
- ISDN
- Cable modem service
- DSL
- LAN connection

The software client does support split tunneling. Related to the prospect of connecting to the Internet over a nontunnel connection, the client also supports a number of personal firewalls from Cisco, ZoneAlarm, ZoneLabs, BlackIce, and Sygate. It also includes an integrated firewall called the Stateful Firewall (Always On).

Summary

You have now covered all the pieces that go into the SAFE exam: the background that Cisco assumes you already have (Chapter 2, "Information Assets"; Chapter 3, "Threats"; and Chapter 4, "The Security Policy"), the management protocols and their functions (Chapter 5, "Management Protocols and Functions"), the SAFE Blueprints (Chapter 6, "The SAFE Security Blueprint"; and Chapter 7, "The Extended SAFE Blueprints"), and

the products as they might be configured in the campus (Chapter 8, "Products in the Campus") and the edge (this chapter).

You can finally put all these pieces together to see how they make a network that is designed for security as well as information exchange. In Chapter 10, "The Small Network Implementation," you do this for the small network; in Chapter 11, "The Medium Network Implementation," you do this for the midsize network; and in Chapter 12, "The Remote-User Design," you do this for the Remote User Design. After that, it's time to take some practice exams and see if you are ready for the real thing.