Products in the Campus

Terms you'll need to understand:

- ✓ TTY line
- ✓ VTY line
- ✓ Strong passwords
- ✓ IDS
- ✓ NIDS
- ✓ HIDS
- ✓ **ip audit**
- ✓ CS ACS
- ✓ Intrusion prevention
- ✓ Zero-day attacks
- ✓ VMS
- ✓ IBNS

Techniques you'll need to master:

- ✓ Creating Access Control Lists/access lists
- ✓ Using VLAN segmentation
- ✓ Performing NIDS/HIDS tuning
- ✓ Performing network device access management

The largest part of any network, at least in the number of hosts present, is almost always the campus, that portion of the network inside the main interface to the outside world (the edge). However, although it has more hosts, it is a bit simpler in its security configuration needs because nothing should (not *will*, only *should*) get in this far from the outside. Of course, you still have to protect resources from the possible intrusion from the outside, just as you must protect them from internal threats (which you can never discount—remember, they are the larger number of attacks, according to some studies).

In addition, some of what you do for security reasons parallels things that you probably already do for network traffic management. That makes the campus a good place to start our look at what Cisco products are present and how they should be configured in a network designed along the SAFE Blueprint principles.

The SAFE Blueprints do not require Cisco products; every version of the SAFE Blueprints specifically says that. So why are we looking at Cisco products and how to configure them to work in a network designed according to a SAFE Blueprint? Look again at the exam description:

"The Cisco SAFE Implementation exam tests the knowledge and skills needed to use and implement the principles and axioms presented in the SAFE Small, Midsize, and Remote (SMR) User White Paper. Candidates are tested on knowledge of how the following devices can be used to create a complete end-to-end solution: IOS routers, PIX Firewalls, VPN Concentrators, Cisco IDS Sensors, Cisco Host IDS, and the Cisco VPN Client."

In addition, there are test questions about securing routers and switches.

The bottom line is, you are most unlikely to pass the exam if you cannot *implement* a SAFE Blueprint on Cisco equipment.

Inside the campus, you need to know how to implement security with routers and switches, basic IDS (both NIDS and HIDS) configuration, and basic AAA. So we'll discuss those subjects in that order.

Routers

Implementing security with routers actually involves two different things: securing the routers themselves and using routers to secure the network. If you don't control the router, you might not be controlling the traffic; that makes controlling (securing) the router the first step.

Securing routers is not just a matter of giving them a strong password. It means that all the paths into the router are under your control, and it means managing what someone (who is permitted access) can do after a successful login; we call that securing access. But that is not all that goes into securing

a router: There is also securing what the router is allowed to do—securing services and management protocols to be used. These are really two different things, so we cover them one at a time.

Securing Access

Before you can secure access to the router, you must know the paths into it:

- Console port
- Auxiliary (aux) port
- OOB connection
- In-band connection

All interactive connections are referred to as lines because they use the IOS TTY (originally "teletypewriter") abstraction. Local connections (those that are directly wired in, such as the console or aux ports) are called *TTY lines* (or just TTYs). Those that connect via the network (in- or out-of-band) are virtual TTY, or *VTY lines* (VTYs). Telnet, of course, is the protocol used to emulate a direct connection (like a console port) over the indirect connection, or VTY. You are "virtually" directly connected when you use Telnet over an indirect connection.

You can disable access to any line (TTY or VTY) by configuring it with the **login** and **no password** commands. This condition of "no access" is the default on VTYs but not TTYs. *However*, you should think twice before you use this: You must always have a way into the router to fix any problems (whether or not they are caused by the lack of access). You should certainly consider using this procedure on any line that you never anticipate using; for those lines that you do anticipate using (such as console or Telnet connections), use authentication and strong passwords. You can also control the protocols allowed on any given line by using the **transport input** and **transport output** commands in line configuration mode.

On a network of any size, most of the configuration updates will be done via VTYs. It is common to use Telnet connections for this (either in-band or OOB), but SAFE recommends the Secure Shell protocol (SSH) instead. You can configure this on a given line with the **transport input ssh** command. If you are connecting from a remote host, whether in-band or OOB, you might choose to require IPSec (because that is often applied on edge devices—we cover it in Chapter 9, "Products in the Edge").

In addition to limiting the protocols that can be used on a given line, you should restrict which addresses can connect via that line: Use a standard IP access list (all you need is source addresses—protocols are handled with the **transport input** command) to specify the workstations allowed to connect.

You might restrict this to a subset of your address block, or you might prefer to have the option to use any workstation inside your network. If so, however, someone inside the network (or someone able to spoof an inside address) could open VTY sessions on every line and lock you out of anything but a direct connection.

Most devices that run the IOS have a limited number of VTYs (usually five). You might want to assign a more restrictive access list to the last of these; if you restrict the connections to line VTY 4 to only those that come from a single, designated administrative workstation, then if a hacker does manage to lock up the other VTYs, you should still be able to get in through that one. Preventing such a lock-up (at least, one that lasts for any extended period of time) can be done by establishing a timeout on idle lines with the **exec-timeout** command in line configuration mode.

Other protocols that you might want to use include TFTP or FTP for backing up configuration files and for downloading software images. You should restrict those protocols to only the servers that you designate, via an extended IP access list. For that matter, access for all management protocols that you choose to use (you need not choose them all) should be restricted to your designated servers only, and you should log any denials.

Next, we need to address two interrelated topics: passwords and AAA. Passwords should always be encrypted through the **service password-encryption** command, and you should always use an enable secret rather than an enable password. The passwords used for various TTYs and VTYs should be strong:

- A minimum of eight characters
- Both uppercase and lowercase letters used
- Numbers included
- Special characters (from the uppercase set of keyboard numbers, such as **@#$**) included

As a rule, every form of access to the router should be authenticated via AAA, if possible. The SAFE validation laboratory principally used TACACS+, although RADIUS was used with the VPN concentrator. However, it is possible (though hopefully unlikely) that you might lose the AAA server or connectivity to it at the same time that you need to make a configuration change to a router. That is why the AAA implementation should contain a back door—a protected back door, of course.

To enable AAA, you must enter the command **aaa new-model** (if this is the first implementation of AAA—if it is not and you want to start from scratch, first erase the existing AAA configuration with the command **no aaa new-model**). The back door comes from an entry along the lines of **aaa authentication login no_tacacs line**, which is applied very carefully. Taking this command apart, you have the following:

- **aaa authentication** specifies that you are configuring identity validation.
- **login** specifies the action to be authenticated.
- **no_tacacs** is the name of this procedure list.
- **line** is the method to be used (use the line password).

If this command is applied to the console 0 TTY (with the command **login authentication no_tacacs** in line configuration mode), there is a means to be authenticated to the router based on the line password, and the incapability to reach the AAA server is not fatal.

As part of your AAA implementation, you make needed authorizations, such as for exec—starting up an exec shell (as in **aaa authorization exec tacacs+**)—and network—for such Layer 2 network protocols as PPP, SLIP, and ARAP (as in **aaa authorization network tacacs+**). Of course, if someone does something to the router, you want to know who has been logged in, when, and what they did; you need to have turned on accounting, which, of course, does not use quite the same commands: **aaa accounting {exec|network|...} start-stop tacacs+**. In this case, the initiation and termination of exec or network activity will be recorded by the accounting server portion of the TACACS+ server. The **wait-start** option (instead of **start-stop**) requires the process to hold up until the accounting process signals that it has begun.

Finally, you should configure a banner to be seen on logins. This should not welcome guests or anyone else. Instead, it should warn that this device is the property of the organization and is to be used only for the authorized activity of that entity.

Securing Services and Management

With access and configuration on the router taken care of, you must control what the router is allowed to do: what services it will run, with whom it will exchange information, and how it will know to trust them. Likewise, you must be sure that the management protocols it exchanges information with are secured.

Routers often have (possibly depending on the actual release of the IOS that is loaded) a number of services that are turned on by default. These services reflect the "good old days" when networks were small and everyone could trust each other. We all know that those days are long gone. Whether Cisco should change the default setting for these services is open to debate; the point for you to remember today is that these services should be turned off.

The order in which you issue the commands does not matter much, although most people put these commands at the beginning of a configuration file, to be sure they are read first. Here, in alphabetical order, is a list of service-management commands that you should include in your configuration file on every router, regardless of its location:

- **no cdp run**
- **no ip bootp server**
- **no ip domain-lookup**
- **no ip http server**
- **no ip source-route**
- **no service finger**
- **no service tcp small-servers**
- **no service udp small-servers**

What are these, and why should you care? We discussed the reconnaissance value of CDP in Chapter 5, "Management Protocols and Functions," and we stated then that Cisco recommends turning it off; if you don't turn it off everywhere, at least turn it off on all external-interface routers and on all interfaces (via **no cdp enable** in interface configuration mode) that connect to those routers.

You might or might not be familiar with BOOTP, an older protocol for assisting diskless workstations. Because the servers to support them were centralized, router software was coded to forward BOOTPREQUEST and BOOTPREPLY message traffic (in the IOS, the **ip helper-address {**IP address**}** command in interface configuration mode enables such forwarding). This traffic ran over UDP ports 67 and 68, respectively. DHCP has replaced BOOTP in almost all instances; if you have diskless workstations (they are making a comeback in certain areas), DHCP can provide everything they need. The IOS still contains a default setting enabling the router to act as a BOOTP server. However, there is no need for any of your routers to do that: Turn off the BOOTP server.

You might or might not agree with the next command, but remember, you are configuring routers, not servers or workstations. Using **no ip domain-lookup** means that commands to your router will require the use of IP addresses (which it knows how to reach, of course). Unless a stranger knows your addressing scheme and has set aside an address somehow for nefarious purposes, turning off name resolution *on your routers* means that they can't be reconfigured by pointing them to a named host.

Similar to acting as a BOOTP server, what legitimate need is there for your router to act as an HTTP server? Disable that capability (**no ip http server**) to protect your router from accepting and perhaps operating on HTTP requests (which need not be Web page requests, of course).

Source routing is a means of tracing the path that a packet took to get where it did (not the same as the **traceroute** command, although the purpose might be similar). Do you really think it wise to allow someone to be able to find a way to your router's interfaces reliably? Not in a secured network. Disable IP source routing (**no ip source-route**).

The last three items on the list are services: **no service finger**, **no service tcp small-servers**, and **no service udp small-servers**. The finger service (which, by the way, is available at the Windows command line as well as Unix and its derivatives) is a means to learn who is logged on to a system. Even more dangerous, perhaps, than revealing usernames (*now I only have to crack the passwords!*) is that the finger service on a router reveals the processes running on the system, the line number, the connection name, the idle time, and the terminal location. That gives a hacker far too much information about your router and how to access it. The small servers are actually a cluster of services: echo, discard, chargen (character generator), and daytime. They, too, are relics of the early TCP/IP networks, and they do not need to run on your routers.

You might want to disable other services and "features" on interfaces, such as **no ip route-cache** and **no ip mroute-cache**, plus **no mop enable** to disable the old Maintenance Operating Protocol (which is used on DECnet). Also on your interfaces, you should enable MD5 authentication of routing protocol updates, and you should restrict to whom you advertise those updates by applying a distribute list.

That leaves you with the network-maintenance protocols that you do want to use, which should include NTP and might include SNMP. You want to use versions with authentication (v3, as it happens, in both cases). If you use SNMP, use a read-only community unless you actually need read-write capability. You should also set up logging, with the logs going to a dedicated server. For the

logs to be useful, you should have turned on time stamping (**service timestamps log datetime localtime msec**). In all three of these cases (logging, NTP, and SNMP), be sure to use an access list that designates the specific host(s) for that purpose.

When you read over the configurations used in the validation, you'll note an ACL for each of the functions rather than one ACL (even though often more than one management function resides on a single device). This provides flexibility in the configuration in case you need to move a function (but not all of the functions) from one host to another: Change the IP address in the **permit** statement of that particular ACL, and you are done. You should also note that all the access lists that permit management functions on the routers include a specific **deny any log** statement at the end. Even though there is an implicit **deny** at the end of every access list, the explicit denial enables you to log the failed attempts, which you would otherwise never know about.

Switches

Switches are simpler to configure, in part because they are less sophisticated and have fewer functions to manage and secure. Nonetheless, again it is a case of securing access and securing services and management, plus the added task of securing ports.

Securing Access

As with your routers, all switches should use strong passwords and, wherever possible, require AAA for authentication and authorization (and use accounting to record who logs in and does what). One of the differences between the router and switch security settings is that, with switches, you can enable Telnet when (for instance) the user is authenticated by TACACS+, but you can disable Telnet when the user is authenticated against the switch's local database. You should also specify the addresses permitted to create a Telnet connection (along with the addresses permitted for any other type of management connection, for that matter).

Securing Services and Management

Fewer unneeded services run by default on a switch. Therefore, turning them off requires fewer commands: **set cdp disable** and **set ip http server disable**. The rationale is no different from that on the routers; there is simply no need for these devices to offer these services.

You do need to enable the management protocols that you intend to use: Again, these are NTP and (possibly) SNMP—as always, with the latter, use a read-only community. The SNMP community name for your network management should follow the same rules as for a strong password: at least eight characters, mixed uppercase and lowercase alphabetic characters, and use of numbers and special characters. In this case, like all the others we have mentioned, specify the server's IP address to be sure that you are the one managing the switch and getting its reports.

You should turn on logging and specify the server's address, and include time stamping. Finally, as on your routers, you should create an opening banner designating the switch as your asset, for business use only.

Securing Ports

Of course, one of the big differences between a router and a switch is the number of ports. But if a hacker who gains physical access to a switch drops one more cable into an open port, would someone notice? Possibly not. That's why the SAFE Blueprint strongly advises disabling all unused ports in the switch configuration file.

Another very useful feature of switches is the capability to group ports into a virtual LAN, a VLAN. Conversation between hosts in the same VLAN occurs at Layer 2, whereas outside the VLAN it must go through decapsulation and recapsulation at Layer 3. However, traffic from multiple VLANs can share one port via trunking (for transit to a router, for instance, where Layer 3 filtering can take place). If so, that link (a trunk) now can carry many hosts' traffic and is a prime candidate for sniffing or carrying injected traffic. Switch ports can be set to automatically respond to a request to trunk, which could defeat part of your security design. For security purposes, they should be set to have trunking off (instead of on automatic) unless you specifically need it enabled. This ensures that you carry only multiple VLAN traffic when you choose to do so.

Also, especially on every switch that attaches to servers, ensure that port redirection is prevented. Each VLAN can be protected via the **no ip redirects** command. We discussed using private VLANs in Chapter 6, "The SAFE Security Blueprint," if you need to review this.

IDS

You might have noticed in the figures in Chapter 6, "The SAFE Security Blueprint," and Chapter 7, "The Extended SAFE Blueprints," that many of

the switches, even inside the campus, included NIDS; in addition, all servers in both the Management module and the Server module had HIDS. You need to understand how IDS works because it is a significant element of the SAFE Blueprint. Much of this was covered in the SECUR (or MCNS) Exam, and you should have gotten into it in quite a bit of detail in the IDS exam; we only review it here.

An IDS inspects each packet as it passes through the device (if it can—if the load is too great, it inspects as many as it can, but some might pass through uninspected—size the IDS's capability to match the device's throughput). If a packet matches the characteristics of a known attack, the IDS reacts according to its configuration: It can generate an alarm, drop the packet, and/or reset the connection (if the connection is TCP). The packet characteristics can indicate an info (reconnaissance) profile or an attack profile. This depends on the nature of the packet and how many packets with what header characteristics are being detected.

Typically, a network IDS (NIDS) placed on a switch sends an alarm and possibly resets the connection (depending partly on whether the match is to an info or attack profile). A host IDS (HIDS), on the other hand, is configured to be more aggressive; generally, it sends an alarm, drops the packets, and sends a reset.

NIDS will probably see more attacks (because you will place it to monitor choke points in the network, where all traffic passes through a switch), so its alarms will give you a better sense of what is actually happening in terms of where in your network an attack is being attempted (one host? one segment? an entire module?). HIDS, however, will see fewer attacks but could give you a deeper perspective on the nature of the attack. The two systems complement each other.

NIDS Configuration

As a system, NIDS requires the placement of sensors (which are typically software packages on a networking device, such as a switch or a router with the IOS Firewall Feature Set) and then linking them to an IDS Director. Some points to remember about how the IDS works are as follows:

- Cisco has a total of 59 intrusion signatures in its IOS IDS database; they are categorized as info or attack, and as atomic or compound (for a set of four possibilities: info, atomic; info, compound; attack, atomic; attack, compound).
- You create an audit rule via **ip audit name** ***audit-name*** **{info | attack} [list** ***standard ACL*****] [action [alarm] [drop] [reset]]**. Remember to use

a standard IP access list—you are concerned with only source IP addresses (which can be any).

- You apply the audit rule to an interface with a direction. Remember that if you want to know about all attacks, apply the audit rule to incoming traffic because it will be inspected before any ACL processing. If this is too much, apply the rule to outgoing traffic on an internal-facing interface (and it will apply after unwanted traffic is dropped by the ACLs).
- Auditing starts with the IP header and proceeds to the ICMP/TCP/UDP header (depending on the packet's upper-layer protocol); then the application-layer protocol is audited. It might help to think of the audit rule as progressively decapsulating the packet.
- If a match is found, the action specified by the rule is taken.
- Don't forget to configure logging with the **ip audit notify log** command. Note that **log** in this command indicates to use the syslog format (instead of the NetRanger format); it does not indicate a server, per se, nor is it followed by an IP address.

HIDS Configuration

The Cisco Security Agent (CSA) endpoint security software package is the HIDS that Cisco offers; SAFE also works, of course, with HIDS from other vendors (HIDS from Entercept was used in the validation lab, for instance). The Security Agent is a new product, and it offers intrusion *prevention* as well as intrusion detection (you might recognize it as the product set developed by Okena, Inc., which Cisco purchased in early 2003). Prevention does not rely on previously known signatures but is based instead on packet-level behavioral analysis; this allows it to protect against *zero-day attacks*. These are previously unknown attacks, usually exploiting a previously unknown vulnerability. They are also usually a nightmare for the unlucky victim because no one yet knows how to handle the mystery attack.

The CSA is monitored and managed by a Management Center running on the CiscoWorks VPN/Security Management Solution (VMS, discussed in Chapter 9). Configuration is performed from the VMS. Management is done through the Web-based VMS and uses policies (it has 20 default policies available, and you can design custom policies as well). The CSA is available in two types, the Server Agent and the Desktop Agent. The Server Agent is supported on these platforms:

- Windows 2000 Server and Advanced Server
- Windows NT 4.0 Server and Enterprise Server (SP 5 or later required)

➤ Solaris 8 Service Pack ARC Architecture (64-bit kernel)

The Desktop Agent is supported on these platforms:

➤ Windows NT 4.0 Workstation (SP 5 or later required)

➤ Windows 2000 Professional

➤ Windows XP Professional

The Management Center is supported on Windows 2000 Server and Advanced Server (SP 3 required).

CiscoSecure Access Control Server

The CS ACS (as it is usually known) is a software package for Windows 2000 Server or Unix (Solaris) servers. System requirements are as follows:

➤ *Windows 2000*—CS ACS 3.2 is supported on Windows 2000 Server with SP3 installed. Previous releases of CS ACS for Windows (2.5 and 2.6 for instance) were also supported on Windows 2000 Advanced Server and Datacenter Server (without Microsoft Clustering Services); those releases are now at End of Life.

➤ *Unix*—CS ACS 2.3(6) is supported on an Ultra 1 or compatible workstation running Solaris 2.51, 2.6, 7, or 8. CS ACS for Unix is being discontinued in 2003.

The CS ACS supports both RADIUS and TACACS+ for AAA (and it supports their simultaneous use). It is a key element of the Cisco Identity-Based Networking Services (IBNS). With release 3.2, Cisco began offering the CiscoSecure Solution Engine, a 1-Rack Unit security-hardened appliance with CS ACS installed. (Think of it the same way as you think of a PIX versus a router with a firewall—remember to make your choice based on performance of the dedicated appliance versus your needs, and then compare to the advantages of a software-only solution integrated on another device.)

Actually configuring the CS ACS is not in the list cited at the beginning—remember, you might be asked about configuration for "IOS routers, PIX Firewalls, VPN Concentrators, Cisco IDS Sensors, Cisco Host IDS, and the Cisco VPN Client." However, you should understand how the supporting devices, such as the CS ACS, work, even though you might not have to know their configuration commands.

The CS ACS runs as several modules, all of which can be started or stopped individually:

- *CSAdmin*—Provides the HTML management interface
- *CSAuth*—Provides the authentication service
- *CSDBSynch*—Provides synchronization of the CS ACS database with an external RDBMS database
- *CSLog*—Provides logging services (accounting and system activity logging)
- *CSMon*—Provides monitoring, recording, and notification of the CS ACS's performance, and includes automatic responses in some scenarios
- *CSTacacs*—Provides communication between TACACS+ AAA clients and the CSAuth service
- *CSRadius*—Provides communication between RADIUS AAA clients and the CSAuth service

Summary

These are the Campus Module Cisco products that you should know and be able to configure to pass the CSI exam. Although in some instances a router with the IOS Firewall Feature Set is used in the campus, you more likely will find it in the edge, so we discuss it there along with the PIX firewall.

Products used in the edge are the subject of Chapter 9. With that under your belt, you'll be able to put the pieces together for how the products are actually used in a SAFE architecture and what they achieve there—and especially how they interoperate with each other.