Management Protocols and Functions

Terms you'll need to understand:

- ✓ NTP
- ✓ Stratum
- ✓ SNMP
- ✓ NMS
- ✓ MIB
- ✓ CDP
- ✓ AAA
- ✓ OTP

Techniques you'll need to master:

- ✓ Configuring NTP
- ✓ Configuring SNMP
- ✓ Configuring CDP
- ✓ Configuring AAA
- ✓ Configuring TFTP

We have finally arrived at a point where we can dig into the SAFE Blueprint itself. Up to now, we have focused on the material the Blueprint assumed you already knew and/or had available. Without that material fresh in your mind, many important aspects of the SAFE Blueprint would not have seemed reasonable. We'll start by covering the three network-management protocols important to securing a network: NTP, SNMP, and CDP.

The various network-management protocols have evolved along with everything else in networking. As a result, there have been several versions of each protocol we're going to discuss. Newer versions have more capabilities, and you should know which versions introduced significant new features, especially if those features are security related.

Network Time Protocol

The Network Time Protocol (NTP) often seems like a black box: You set the router or other host to get its time updated and never have to worry about it again. But that begs the question: Why should you care that much about the time on any host?

Why Time Matters

Time matters in the network in a number of ways. For starters, the time that information arrives or a particular action occurs might be important. Examples of these situations are common in the financial industry (time-stamping transactional data), in medical situations (time-stamping medical data), and in many matters involving litigation (what did she know and when did she know it?).

But every network has other tasks that are time sensitive. Most have time-scheduled jobs set to run at certain times (usually known as *cron jobs*, after the Unix daemon, crond, which manages them). Suppose that your cron job timing is set to use off-peak hours on the network and the jobs are staggered so they never overwhelm your bandwidth. If one or two devices have an incorrect time setting, their cron jobs will execute at a different time than planned. This could cause serious network problems or conceivably cause one or more of the cron jobs to fail.

System logs are referenced by time. That simple statement has profound implications that ripple through the SAFE Blueprint repeatedly. Almost all troubleshooting starts with a problem report, and efforts are made to trace back through time to learn the origin (as well as having someone work to mitigate the current problem, if necessary). If the logs are thick with entries,

the easy way to find relevant data is to search for a time stamp close to the problem time and work from there.

If the device's time is wrong, the logs that it creates will have incorrect time stamps on every entry. Finding the right entry will be far, far more difficult.

Now consider that the problem you're troubleshooting is more than simple network connectivity; instead, the problem is that an intruder has somehow gained entry into the customer database server. To disguise his steps, he changes the system time before he alters certain settings or creates an account, to have an available back door at his convenience (perhaps he set the year to be two years earlier). When he's finished, he resets the time to what it should be. Now when you review the logs to figure out what went wrong (probably much later, unless something else happens to make you look hard at that server), you search the log by time stamp and do not notice the account creation.

As a third problem involving time, think a minute about operations that are time based or that have time as a component in changing things. An example might be terminating the lifetime of an IPSec Security Association (SA). A separate timer does not "tick away" the lifetime set for every time-limited process; instead, an expiration date/time is set, based on the designated lifetime duration and start date/time. Suppose that the goal was to disrupt VPN tunnels: If the date and/or time on one tunnel endpoint is advanced (moved forward in time), that device will believe that the tunnel's lifetime is expiring before the other endpoint. One end of the tunnel will renegotiate the SA when it believes that the lifetime has reached 30 seconds remaining, and traffic passage will be interrupted while this occurs. A simple script to reset a device's clock every few minutes (perhaps backward as well as forward) could disrupt a VPN tunnel: Traffic flows and then it doesn't, then it flows again and then it doesn't again. The tunnel would seem to flap as though there were a bad circuit between the two endpoints.

While employed by a major equipment vendor, I used a laptop that began losing time (several minutes per day). The time stamps on my copy of files that were exchanged as part of a collaborative project became less reliable. After much maintenance (including a new battery, which failed to resolve the problem) and lost productivity, the laptop was simply replaced.

Finally, of course, access lists based on time of day can be written and applied to interfaces when it is important to allow access at certain times and deny it—even to the same source-destination-protocol sets—at others. System time matters, and it must be system time that you can trust.

Multiple time sources actually are available to a networking device, including its internal battery-powered clock, VINES, and manual configuration. If NTP is available, it is always considered more authoritative, and its value overrides that provided by any other source.

Using NTP

If used, NTP can prevent many of these time problems. It is based on a hierarchical system of time sources, identified by their *stratum*. This number designates both the level in the time server hierarchy and the accuracy and precision of the time it provides. Stratum 1 sources are the most precise and reliable sources available; they are the root-level time servers. Stratum 1 servers have direct access to an atomic clock or radio source (such as the Global Positioning System, GPS). Cisco devices do not usually have such access and usually do not offer stratum 1 service (although the capability is being added on certain high-end devices). Lower stratum levels are designated by the number of hops away from a stratum 1 source. In addition, some telecom vendors have stratum 3 or even stratum 2 clocking built into their core switches; with periodic checks to a stratum 1 server, these provide time service to connected networks.

Devices running NTP automatically choose to connect with the lowest-stratum server available to them (known to them by configuration); this provides them with the most accurate basis for their clock values. Because time must elapse between the time server sending the update and a time client receiving it, NTP uses UDP over IP (because UDP has less latency than TCP) on port 123. All NTP times are given as UTC, Coordinated Universal Time. Don't forget to configure your time zone and whether to update for daylight saving time (or summer).

NTP peerings are statically configured. Messages are normally unicast between peers, but you can configure devices to send or receive NTP messages via broadcast (although Cisco warns that this reduces the accuracy of the time values used because of the one-way nature of the traffic).

Configuring NTP

NTP support can be configured on both routers and some switches. Some of the newer switches use the IOS commands, while others use the Catalyst commands. Both types of commands are given in the examples that follow.

Router

In configuring NTP on a router, you must specify whether the relationship will be one of peers (time synchronized with each other) or whether the router will treat the other as a server (the router will synchronize to the other's settings). The syntax is

```
ntp peer ip-address [version number] [key keyid] [source interface] [prefer]
```

or

```
ntp server ip-address [version number] [key keyid] [source interface]
[prefer]
```

The importance of the optional version numbers, keys, and preference settings is described shortly. You need to configure only one end of the association, whether peer or server. If the other device is NTP-capable, it automatically responds appropriately and forms the association when this device sends an NTP packet.

If you think about that last statement, you begin to realize why NTP must be controlled on the network in some way: Any device that can contact your router could form an NTP association with it and affect your time settings (especially because NTP is more authoritative than any other source on your router or switch). That's why the SAFE Blueprint spends time (if you'll pardon the expression) worrying about NTP.

Switch

Switches, such as those in the Catalyst series, are NTP clients only. They can receive NTP updates as broadcast clients, or they can actively solicit NTP updates from a server. To receive broadcasts, the commands are as follows:

```
set ntp broadcastclient enable

set ntp broadcast delay microseconds
```

The latter command is a manual adjustment to compensate for the delay you believe to be present.

If you want your switch to request NTP updates, use these commands:

```
set ntp server ip_addr

set ntp client enable
```

Securing NTP

Because time can so easily be manipulated, it is important to control which devices affect the time settings on your network. Cisco recommends two methods: access lists (ACLs) on a router and authentication on both routers and switches.

Access List

To secure NTP with an access list, define the acceptable peers or servers with a standard access list, and then apply the access list to NTP via the `access-group` command:

```
access-list 77 permit host ip_addr1
access-list 77 permit host ip_addr2
access-list 77 deny any log
```

This is applied via the command:

```
ntp access-group peer 77
```

Authentication

IP addresses in headers can be spoofed, so it is better to use authentication. Cisco uses MD5 for the NTP authentication method. For both routers and switches, the authentication process must be turned on, at least one key must be established and paired with another NTP device (which, of course, must have the same key), and, in the case of a switch, the NTP client process must have authentication turned on.

For a router, the process looks like this:

```
ntp authenticate
ntp authentication-key number md5 value
ntp trusted-key number
```

The key numbers should match, of course, and the `md5` value is the actual key that is used when that key number is invoked.

For a switch, the process looks like this:

```
set ntp authentication enable
set ntp key public_key trusted md5 secret_key
set ntp server ip_addr key secret_key
set ntp client enable
```

Again, the key numbers should match (the `public_key` value), and the `secret_key` is the actual key used.

NTP Versions

NTP has been available since the late 1980s. Version 2 was the first version that included a limited capability to authenticate NTP peers (per RFC 1119); however, a full authentication capability was not available until NTPv3 (and the SAFE SMR Blueprint cites version 3 as the first to support a cryptographic authentication mechanism). Version 3 also offers improved technical performance of the timekeeping functionality; the features involved are potentially important to more modern, higher-speed links such as Gigabit Ethernet and faster.

Simple Network Management Protocol

When you look at the names of some of the variables used, "Simple" doesn't seem to quite describe this network-management protocol. Yet SNMP is simple in principle, though the details can become quite complex. SNMP is the protocol used by network-management systems (NMS).

SNMP operates at the application layer, over TCP or UDP over IP. An SNMP manager (a software package on the NMS station) works with SNMP agents (also software packages) deployed on devices in the network. Both managers and agents use a Management Information Base, or MIB. Each MIB contains data objects (often also called MIBs) that are used to track activities and make configuration changes on some devices. MIBs are listed in hierarchical order, with dots separating levels (much like the dotted notation in fully qualified domain names).

SNMP hosts are organized into communities, and the community string serves as an identifier (an *extremely* weak sort of password, not unlike a Windows workgroup name). One other aspect of community names is worth mentioning here: The default community name is `public` and is very well known to hackers. Remember, this is a network-management protocol—it is used to configure and gather data from networking devices. That makes it very useful to hackers willing to work through the naming hierarchy to the message they want (and their scripts can be used by less capable hackers who really make a hash of things). Do not use `public` as your community string! Furthermore, be more creative than some network managers, and do not use `private`, either. Use a string that, like a strong password, is not subject to a dictionary attack (it uses letters *and* numbers *and* special characters, without a partial dictionary word sequence).

Members of the community can use traps (unsolicited informational messages) or gets (requests for information) and sets (command messages). SNMP messages can use UDP or TCP; gets and sets use port 161, while traps use port 162. Informational messages (traps and the replies to gets) require only a read-only capability, but issuing sets requires both read and write capability. Cisco takes advantage of this in its recommended configurations in SAFE.

Configuring SNMP

The SAFE Blueprint recommends using two separate SNMP communities, one a read-only community and one with read-write capability (if you must have a read-write capability—it might not be necessary). Again, routers and switches have slightly different commands.

Router

You designate the community string and its type with this command:

```
snmp-server community string {ro|rw} number
```

The string is the (hopefully well-constructed) community name, and the number used in this command is the number of the standard access list that identifies the acceptable host addresses (hosts that are allowed to access the community). Such a command and its associated access list might look like this:

```
snmp-server community yhG2b@&sm! ro 42
access-list 42 permit 172.28.42.12
access-list 42 permit 172.18.42.10
access-list 42 deny any log
```

Switch

The switch SNMP commands look like the "switch versions" of the router commands:

```
set snmp-server community yhG2b@&sm! ro
set ip permit enable snmp
set ip permit 172.28.42.12 snmp
```

One difference between the router and switch versions of the `snmp-server` command is that the switch using the Catalyst OS has a third possibility:

- `read-only`—Read-only access to all MIB objects, but not the community string
- `read-write`—Read and write access to all MIB objects, but not the community string

- `read-write-all`—Read and write access to all MIB objects, including the community strings

Practicality

You might note in the sample configurations in Appendix A of the SMR SAFE Blueprint that the SNMP servers are also the logging servers. That is not a requirement, by any means, but it is a good economy of resources: Logging servers and SNMP servers should both be carefully protected and monitored. By having two of each, you gain redundancy; by placing both functions in those two hosts, you don't have an excessive number of devices to configure, protect, and monitor. Remember, the SAFE Blueprint is intended to be practical.

SNMP Versions

SNMP has reached version 3. The original version (usually just called SNMP) is rarely seen now, for good reason. SNMP did an adequate job, but SNMPv2 offered more advanced monitoring and configuring capabilities. Unfortunately, security features were not added until SNMPv3, which also includes a new message format. SNMPv3 is backward compatible with SNMPv2 (so that messages can be exchanged). However, it is strongly recommended that networks transition to SNMPv3 because it includes authentication between devices and message security.

Cisco Discovery Protocol

The Cisco Discovery Protocol, or CDP, is a proprietary protocol usable with Cisco networking devices for topological discovery. We've all had the happy experience of being called upon to work on a network whose topology we don't know—sometimes, in fact, part of the job is to figure that out.

CDP is media and protocol independent; it works with SNMP, using a set of MIBs known as CISCO-CDP-MIB, which depends on the presence of a number of other MIBs. Although it can report Layer 3 information, it is not routable. Instead, CDP runs over Layer 2 using the Subnetwork Access Protocol (SNAP). Based on configurable timers, CDP devices send periodic updates to a multicast address; included in the configurable timers is a hold time (or time to live) for the update. Each update includes information about at least one interface on the reporting device that can receive SNMP traffic.

By operating at Layer 2, the CDP information goes only to the immediately adjacent Cisco device; CDP packets are not forwarded among CDP devices.

However, by working one's way through the devices, a patient person can derive the network topology. This is because the information given in a CDP update is extremely useful (to a legitimate network administrator or to a hacker). What kind of information depends on whether the user requests a simple `show cdp neighbors` or the more informative `show cdp neighbors detail`.

The first command (`show cdp neighbors`) yields this information about every connected CDP-enabled neighbor:

- Device ID (name on the network)
- Local interface (the connected port on this device)
- Hold time (how long the information remains valid)
- Capability (reported as a code for router, switch, trans bridge, source route bridge, host, IGMP, or repeater)
- Platform (the device hardware type, such as 7206VXR)
- Port ID (the port on the distant device, such as Fas0/0/0)

With the second command (`show cdp neighbors detail`), in addition to the previously listed information, you (or a hacker) can learn this:

- IP address of the distant interface
- Duplex setting
- CDP version in use
- Software version running on the device

Although this is extremely useful if you have just inherited a network whose diagrams are dated (if anyone can find diagrams), it is also a wonderful exploratory tool for a hacker. Without going into too much detail, suppose that you have a simple network topology such as the one in Figure 5.1.

Figure 5.1 shows redundant access to the Internet via two perimeter routers (PR1 and PR2), with redundant distribution into the network core via two distribution routers (DR1 and DR2). Let's think about what you can learn with access to PR2.

From PR2, you can learn about the existence, name, software version, and so on, of both DR1 and DR2. You can also learn the IP address to try to Telnet in. Because these routers are inside the perimeter, they might very well allow a Telnet connection from a fellow network member. From either of those, you can learn everything about the other DR, PR1, and core devices (probably two or more core routers). From any of the core routers, you can learn about their neighbors, and so on through the network.

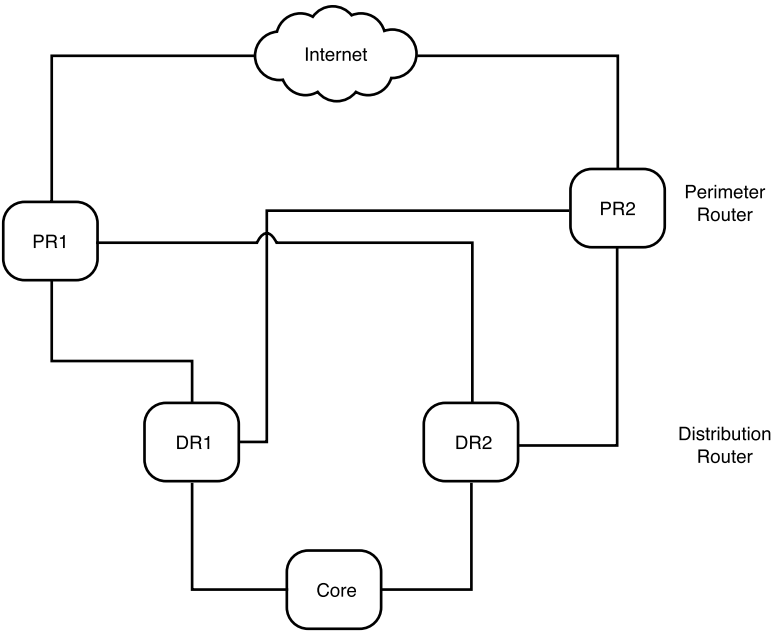

Figure 5.1 A simple network topology.

Configuring CDP

From this terribly simple example, you can appreciate how much information can be gleaned about your network just from a weakness in one device. That's why the SAFE Blueprint recommends turning off CDP. If you need to use CDP inside your network, disable it on your edge devices and on the interfaces of the devices that connect to your edge devices. To disable CDP globally on a router, use this command in global configuration mode:

```
no cdp run
```

To disable it on a particular router interface, use this command in interface configuration mode:

```
no cdp enable
```

On a switch, you can disable CDP globally with this command:

```
set cdp disable
```

On a given port, use this command:

```
set cdp disable [mod_num/port_num]
```

The last item designates the module and port (or range of ports) to disable.

The switch command syntax seems quite straightforward, but the router commands might seem backward. Nonetheless, you should get all these commands straight. You might need to know which command is used where for more than one Cisco certification exam.

CDP Versions

CDP has only the original version (CDP) and a second version (CDPv2). The second version is the default on newer releases (beginning with 12.0T). It added the capability to exchange information on VTP management domain name, native VLAN, and full/half-duplex status (which can prove useful in switch troubleshooting because VLAN and duplex mismatches are displayed).

Authentication, Authorization, and Accounting

Authentication, Authorization, and Accounting, more commonly simply abbreviated AAA, is a major part of securing network management. Even though you undoubtedly saw a number of questions regarding it on the SECUR exam (or its predecessor, the MCNS exam), you should be prepared to see questions about it again on the CIS exam. AAA is a significant piece of the SAFE Architecture. In case it has been a little while since you've reviewed the material, we include a brief overview of each function and how it is implemented in the SAFE Blueprint.

Authentication

"But Granny, what big ears you have," said Little Red Riding Hood.

"The better to hear you with, my dear," the wolf replied.

Nursery rhymes and other children's fables come down through the generations to teach fundamental lessons (often gruesomely, of course). In this case, the lesson is all too clear: Not everyone is whom they appear to be. (Of course, we could also cite "The Three Little Pigs," on keeping intruders at bay.)

The whole point of authentication is to verify rather than assume the identity of those attempting to access critical information resources. Authentication might be as simple as storing a username and password in a

local database on the device, or it might use a stronger program operating on a server inside the protected perimeter. Although Cisco supports servers using TACACS+, RADIUS, and Kerberos, the SAFE Reference Implementation in the Validation Lab uses TACACS+, and this is strongly preferred.

Briefly, the possible forms of authentication are as follows:

- None (complete trust)
- Username and static password
- Username and aging password
- One-time password (OTP)
- Token cards or soft tokens (OTP)

If only a few users are to be authenticated against only a few devices, Cisco states that using local username and password databases on the individual network devices is acceptable. However, when you have more than a few of either (and the meaning of "a few" is not explicit, but most people have a feel for it), it is better to use a dedicated database on a separate server. In that way, the database is more likely to be managed properly—kept current and with appropriate permissions.

Back Doors

Cisco recommends securing access to your routers and switches via authentication. However, Cisco also knows that things happen and thus recommends setting a back door to get in, just in case (SAFE is about practical security because, if it's not practical, it won't be used). That means it's useful to have at least a limited local user database, even if you're using a AAA server for normal authentication.

Configuring Authentication

If authentication is against a local database, you must add username and password combinations to the device. If you use a AAA server, this local database is still available; it can be one of the methods used for a type of authentication. On a router, in global configuration mode, use this to add a user and password to the local database:

```
username name password password
```

On a switch, use this command in global configuration mode:

```
username name secret {0|5} password
```

The option `0` on the secret (the password) means that the password is stored without encryption, while the option `5` means that the password is encrypted with the MD5 algorithm. Needless to say, encryption (`5`) is safer.

If you are using a local database instead of an AAA server, dialup sessions to a router carried by PPP need to be authenticated using PAP, CHAP, or MS-CHAP. This requires you to turn on (enable) PPP encapsulation, after which you can specify the authentication method. In line configuration mode, use these commands:

```
encapsulation ppp
ppp authentication {pap|chap} {default|list-name}
```

Using a list-name is permitted only with AAA. Be aware that passwords are sent in clear text when using PAP, but CHAP involves sending a challenge and then comparing the hashed response returned against the hash calculated locally. With CHAP, passwords are not sent in the clear, so it is strongly preferred if you not use an AAA server. Worth noting is that Cisco does not recommend ever using PAP, if it can possibly be avoided.

MS-CHAP operates a little differently; it is a Microsoft extension of the CHAP protocol. To use it, again in line configuration mode, you enter these commands:

```
encapsulation ppp
ppp authentication ms-chap
```

If you use AAA instead of a local database, you must enable AAA on the device and then establish the method(s) to be used for it—the sources to be checked, in the order in which they will be checked. Note that the first actual response (whether accepting or rejecting the login) ends the authentication process (this does not include an ERROR response, which means that the server could not evaluate the authentication). At some point in the AAA configuration, you must also identify the server holding the database (including the server type as well as its IP address), so it is included here as well. On a router, in global configuration mode, enter these commands:

```
aaa new-model
aaa authentication login {default|list-name} [method1 [method2…]
tacacs-server host ip_addr
tacacs-server key key
```

Remember, the `aaa new-model` command is necessary only when creating AAA when it is not already present. You might have several lines of authentication methods (method lists): a default list (to be used wherever a method list is not specified), a named list for incoming lines (such as vty or console), and a

named list for PPP. Note that, with the PPP method list, the optional command `if-needed` is available. This refers to the fact that the methods that follow will be used only if other authentication has not already been made.

Enabling AAA and setting the methods on a switch is just as simple. For TACACS+, use the following commands:

```
set tacacs server ip_addr
set tacacs key key
set authentication login tacacs enable telnet
set authentication login local disable telnet
```

The second pair of commands show that you're willing to allow Telnet if the party is authenticated by the TACACS+ server; local database Telnet authentication is not allowed. Remember, local login and enable authentication are both enabled by default; you are changing that to permit Telnet login only if TACACS+ authentication has been met.

Authorization

Authorization is granting permission to do something. Especially when under stress during the exam, it's easy to confuse authentication and authorization (it doesn't help that the words look so much alike at their beginnings and ends, which reinforces the need to read every question carefully).

If you are using a local database on a router, it is useful to assign the various commands to specific privilege levels (remember that they range from 0 to 15, with 0 being user EXEC and 15 being privileged EXEC). The command to assign users privileges (in global configuration mode) is as follows:

```
username name privilege level
```

On a switch, the commands are similar, though more limited (to `config`, `enable`, and `all`), and the authorization can be applied to `console`, `telnet`, or `both` (the default):

```
set authorization commands enable {config|enable|all} [console|telnet|both]
```

Again, this is useful only if you have a few users and a few devices to be configured; otherwise, you should use AAA.

To use AAA for your authorization, on a router, use the appropriate commands to authorize exec privileges, network access (for PPP, SLIP, and ARAP, the AppleTalk Remote Access Protocol), commands, and reverse-access. The same command must authorize a source for authorization. The source might include `if-authenticated`, which means that authorization is granted automatically if the user has been authenticated.

Much like the SECUR or MCNS exams, you should know which options are available on an authorization command compared to authentication. There are differences.

This command invokes AAA for authorization on a router (you can enter it several times, with a different name each time, for different services):

```
aaa authorization {network|exec|commands level|reverse-access}
{default|list-name} {if-authenticated|local|none|radius|tacacs+|krb5-
instance}
```

Notice that you have three sets of choices, and you must choose at least one from each of them.

On a switch, the choices are somewhat different:

```
set authorization {commands|enable|exec} {tacacs+|if-authenticated|none}
{tacacs+|if-authenticated|deny|none} {console|telnet|both}
```

This time, you must choose from four sets, but the principle is the same. The second set of options is the primary authorization method; the third set is the fallback option. Notice that although a router can use RADIUS and Kerberos, a switch running the Catalyst OS cannot.

Accounting

Accounting is boring, is a pain, and is where security violations often are first detected. If you don't appreciate that, read the excellent book *The Cuckoo's Egg*, by Clifford Stoll. Accounting is your opportunity to track not only accesses (including when they began and ended), but also which services were used—and all this can be tracked by user.

To configure accounting on a router, use the following command (again, you can enter the command several times, depending on the options you want to use):

```
aaa accounting {auth-proxy|system|network|exec|connection|commands level}
{default|list-name} {start-stop|stop-only|wait-start|none} [method1
[method2]]
```

The `list-name` option includes the following choices: `auth-proxy`, `commands`, `connection`, `exec`, `network`, and `resource`. The timing options include accounting for both the start and stop of activity (`start-stop`); only the stop (`stop-only`), the start and the stop, with the start beginning after accounting is initialized (`wait-start`); and `none`. Notice that a maximum of two methods is allowed; this is different from authentication and authorization:

Authentication allows up to four methods, and authorization allows one method per authorization command.

On a switch, you can see the now-familiar pattern of the same idea, expressed in Catalyst OS syntax. However, depending on what you want to do, you might need one or more commands to get the job done:

```
set accounting commands enable {config|enable|all} [stop-only] tacacs+
set accounting {connect|exec|system} {start-stop|stop-only} (tacacs+|radius}
```

Notice that the accounting for `commands` is a separate command, with different options than those for outgoing connections from the switch (`connect`), `exec` activities, and `system` activities.

We've covered a lot of possibilities, especially when it comes to AAA. Having gone over this, it is a good idea to go back to the SMR SAFE Blueprint and look at Appendix A, the configurations used in the Validation Lab. You'll see the pieces all put together in the router and switch configurations used.

Trivial File Transport Protocol

The Trivial File Transport Protocol (TFTP) is often used to transfer software images (for system upgrades) and configuration files from a central storage location to a networking device. Because it is so useful in that regard (especially because of its low overhead—it runs over UDP over IP and actually has a message-confirmation process, although it is not as sophisticated as that of TCP), Cisco does not recommend not using it. However, no usernames and passwords are required between a TFTP server and client, and the information is sent in the clear—which could potentially expose your configuration files to a packet sniffer en route.

Normally, TFTP uses UDP port 69 along with UDP ports greater than 1023 for the data stream itself. These ports, too, are well known to hackers, along with the typical uses of TFTP in the network. Therefore, Cisco recommends that TFTP sessions be run through IPSec tunnels whenever possible. The payload should be encrypted in the tunnel.

Summary

You should know these network-management protocols to successfully face the CSI Exam: NTP, SNMP, CDP, AAA, and TFTP. When you understand these, you're ready to look at the SAFE Blueprint and how the network pieces are grouped together and then secured.