Threats

Terms you'll need to understand:

- ✓ Script kiddies
- ✓ Reconnaissance
- ✓ Ping sweep
- ✓ Port scan
- ✓ Target discovery
- ✓ Eavesdropping
- ✓ Packet sniffer
- ✓ Social engineering
- ✓ Resource overload
- ✓ Distributed denial of service

Techniques you'll need to master:

- ✓ Finding threat information
- ✓ Analyzing threat types

If nothing could ever go wrong, we wouldn't need to worry about protecting those network assets. But, of course, things do go wrong, and many of those things are related to security threats. In Chapter 2, "Information Assets," we said that you need to be able to identify just what your information assets are, what you need to protect them from, and what tools you might have available to do that job—and do it while keeping the network usable by the ordinary user. This is where we look at what you're protecting those information assets from. Cisco breaks out threats by their origin and by their type.

Origin

Threats can originate from inside or outside your network. That is actually not always an easy line to draw, though, as we'll see.

Internal Threats

When the SAFE Blueprint discusses the origin of threats (internal versus external), it repeats the perception that most threats actually originate inside the network instead of penetrating your perimeter from the outside. Although that has historically been true, it could be changing. Every year, the FBI and the Computer Security Institute conduct and then publish a survey on the threats large organizations actually faced in the previous year. The percentage of incidents that originate outside the network is now essentially equal to the number originating inside the network. You might have heard about this or seen it written in industry publications. For the purposes of taking the test, however, you should be prepared to say that the majority of threats originate inside the network rather than coming from external sources.

What or who are these internal threats? People, of course, but it helps to narrow the most likely candidates rather than simply assuming that all people inside your network are threats. Of course, in some very security-conscious networks, everyone must be considered a potential suspect, but even there, some people are more likely candidates than others. Who among these people on the inside are likely to cause problems?

- Current employees with dubious intentions
- Current employees with unauthorized activities
- Employees who mismanage their environment
- Contractors who fit these same descriptions

Bad Intentions

Why would employees or contractors want to hurt the company, especially when jobs are tight? There are as many reasons as there are people: Someone might have a grudge for a promotion that he felt was deserved but went to someone else; another person might think that by creating a problem, she'll be a hero for finding and fixing it. At least in this category, you can confidently say that the person intended to do the enterprise harm (even if the goal might have been to fix it later). Likewise, someone who has done something wrong (embezzling or stealing from inventory, for instance) might want to limit the visibility of that wrong by removing evidence of the actions. Some employees, of course, will never be satisfied; no matter what management does to accommodate them, they will remain disgruntled.

Unauthorized Activities

So many stories have arisen of employees or departed contractors hosting illicit Web sites, or even web-based businesses, on a company's network that it's easy to become blasé about the entire idea. But it remains true: People do use corporate resources to host pornographic Web sites and to host music and movie files for peer-to-peer swapping. People do use their corporate email accounts to buy or sell items on EBay or other auction sites. As this was written, yet another article appeared on IDG.net reporting that more than three quarters of business networks checked had unauthorized peer-to-peer networking software installed, and no company with more than 500 PCs had none. Unauthorized uses also include hosting other businesses, some of which might be legal under authorized circumstances, or hosting personal sites.

Outside audits regularly uncover evidence of these activities, and people are even fired for having done them. Yet the next audit might find that another entrepreneur has taken the departed first business operator's place, with a new and improved set of activities. The problem here is less the intent to do harm (because harm raises interest in what's going on and draws unwelcome management attention) than it is that these activities introduce code that IT does not know is operating. The code might have vulnerabilities that can be exploited if the customers—or even browsers—include hackers.

One other related factor to remember is this: Allowing unauthorized hosting makes a business look incompetent in managing its own affairs, which is very bad for its image in front of the public. If that unauthorized activity includes illegal business, such as pornography or peer-to-peer file sharing, the business can be held legally liable for allowing it to happen. That could prove very expensive.

Mismanagers

These are not the pointy-haired bosses of Dilbert fame; they are otherwise well-intentioned persons (employees or contractors) who make changes to their operating environment. Those changes can introduce holes in an otherwise well-guarded network. An example is an employee who likes to get a little more work done after hours from home and installs a package such as pcAnywhere for operating his desktop remotely. pcAnywhere is a commercial product, not malware, but if it is operating and IT doesn't know about it, it can create an opening in perimeter security that a hacker can exploit. Many employees, including less-experienced system administrators who should know better, or contractors use Instant Messaging or Internet Relay Chat without authorization. This, too, creates openings for malware. Many worms are now entering networks via chat because antivirus packages do not scan every object that enters; they scan only those that enter via email. Full-system virus scans will eventually catch the malware (if definition files are kept current), but cleanup is much harder than prevention. Again, there is probably no intent to cause harm, but an exposure is created by the addition of unmaintained or unauthorized software. That doesn't begin to address those who add a modem to dial in….

External Threats

If internal threats are people inside the network, external threats must be people outside the network, right? Remember, however, that when you break things down simplistically like this, much depends on where you draw the network boundary. For instance, if you draw the boundary at your edge, remote users are external. Even if they tunnel in, you might not necessarily extend the network boundary to their devices, especially if they are connecting via the Internet. You might want to keep thinking of them as external.

In this case, though, the external threat is not directly the person, who might or might not be the kind of person we would say fits the internal threat category (if accessing the network from inside). Instead, the external threat is the fact that the device used (whether a laptop for a mobile worker or a desktop for a teleworker) is significantly exposed to the outside world, especially the Internet. Unlike a host inside your perimeter (in your campus), this host might spend much of its time on the Internet without necessarily going through your security precautions. (There's a way around that, which we'll discuss when we cover some design alternatives in Chapters 11, "The Medium Network Implementation," and 12, "The Remote-User Design," but there are always disadvantages as well as advantages associated with choosing the alternatives.)

The courseware for Cisco's SAFE Implementation course also categorizes external threats as structured or unstructured. "Structure," in this context, refers to the degree of organization and planning, or the amount of method applied in the attack, as opposed to haphazard efforts that might seem almost random to an observer. Note that both structured and unstructured threats can be malicious in intent or can be the result of human clumsiness or error.

More conventional external threats are people outside your organization. Cisco categorizes them as follows:

- Thrill-seekers
- Competitors
- Enemies
- Spies
- Thieves
- Hostile former employees
- Others

The thrill-seekers are often simply engaging in a social activity—seeing what they can find and/or trying to impress their friends; they generally pose an unstructured—but still dangerous—threat. Thrill-seekers might or might not have substantial skill; they are often (but not always) *script kiddies*: relatively unskilled users running scripts developed by skilled users that the script kiddies often do not understand. The clumsiness and ignorance of these thrill-seekers can cause significant damage if they manage to penetrate a network. Some of the more well-known scripted tools are L0phtcrack for password cracking and BackOrifice for exploiting vulnerabilities in Microsoft's Office suite of products.

Competitors, of course, exist everywhere in economic life, but business competitors can have a significant incentive to snoop in your network: It can save them millions of dollars if they can learn the lessons of your development without spending the money it cost you to learn them. Most businesses maintain a group to analyze their competition, using whatever information becomes available.

Spies are a threat to businesses as well as governments. Because of the high cost of developing new products and the intensity of competition, which leads to lower prices, corporate espionage is a problem to protect against. If you don't think corporate espionage really happens, consider first that Cisco thinks that it is serious (which makes it serious for the exam, of course).

Second, take some time to read a few of the reference books listed at the end of the chapter. The stories in them have been sanitized to avoid lawsuits, but they are otherwise real.

So what exactly is the difference between competitors and spies? Cisco doesn't really say, but this might help: Competitors are in the same line of business (pharmaceuticals, mufflers, batteries, and so on), while spies are in the information business. Spies are usually third parties that obtain information for others; competitors are trying to obtain it for themselves. Either way, the hackers here generally pose a structured threat due to their greater skill and more organized effort.

Thieves are another group that has plagued business since there was such a thing as business. And the crime must pay (or, at least, be expected to pay) often enough to make it worthwhile to keep trying theft. What can be stolen via a network? Information such as credit card numbers or other data for perpetrating identity theft is always valuable. Surprisingly, information about the network can be valuable: If you can learn enough about the network devices, you might be able to control them and the traffic they carry. In short, if it can be used to create value for someone, it can be expected to be stolen at some point.

Hostile former employees (or contractors), such as current employees with a grudge, seek to damage the network or information assets for revenge. Sometimes they want to "get even" for whatever affronted them by stealing and selling information. What makes them different from outsiders is the likelihood that they have at least some inside information about the network—they start with an advantage over other outside threats.

Finally, Cisco provides the catchall category of "other." As one policeman said, whenever you think you've seen it all, you wake up one morning and realize that you haven't seen it all. A time will come when you will find a network threat that doesn't exactly fit any of the specific categories; that will be your example of the "other" group.

Threat Types

Whether it comes at you from the outside or from within, the threat to your network can take one of four broad forms:

- Reconnaissance
- Unauthorized access
- Denial of service (DoS)
- Data manipulation

You can expect to see a number of questions related to threat types when you take the actual exam. The questions might require you to recognize the category in which a threat belongs, as well as know the names of the types. You should review this section's key points before taking the exam.

Reconnaissance

Reconnaissance is a word that evokes thoughts of war and military observers scanning for targets through binoculars and then placing the acquired target information on maps used to plan the attack. The analogy is apt; network reconnaissance is indeed the unauthorized discovery, mapping, and monitoring of the systems and services in a network, along with probing for their vulnerabilities. To do all this, an unauthorized person uses target discovery, network commands, ping sweeps and port scans, eavesdropping, and information theft.

Target Discovery

Finding out what is available to be attacked starts at the very broad level: domain names and IP address blocks. The discovery process gradually narrows the target definition, expanding the names and narrowing the address ranges until the person has individual hostnames and addresses.

Certain targets are easier to discover than others because some are required to be publicly listed. For instance, DNS records must list the publicly accessible servers and their addresses so that the public can access them (name resolution must work). But if you create the hypothetical BillyBong Corporation and obtain a registered block of addresses, that block can be learned via the Registry. (Regional Internet Registries are ARIN, RIPE, APNIC, and LACNIC; AfriNIC has been proposed but not yet accepted by ICANN at this time.)

What Can Hackers Really Learn?

How much can a hacker really learn about you, starting from an Internet Registry? Let's take a look. From my firewall log, I see that 12.237.32.178 has been trying to connect to my network on port 445 (microsoft-ds, one of the most frequently tested ports). Because I'm in North America, I first try ARIN, **www.arin.net**, and enter the address in the **whois** search window.

From the output, I see that the address belongs to AT&T Worldnet Services, which owns the entire 12/8 (Class A) address block. I see the names of four name servers (DNS servers). Aha! I ping the name of one of them and discover that it uses an address in the 199.191.128.0 block. This means that AT&T has another address block for me to investigate. Pinging the other three name servers shows me that AT&T runs dual name servers on two different networks, clearly using redundancy.

The ARIN information listing also gives me names, telephone numbers, and addresses, always a help if I'm inclined toward social engineering (more on that coming up).

If I'm running a DNS server myself, as many skilled hackers do, I can interrogate AT&T's name server and try for more information. If I'm lucky, I can even manage a zone transfer and pull in the whole set of public addresses for the servers. Alternatively, I can open a Telnet session to a mail server on port 25 (mail servers tend to use fairly standardized names, so I can guess at that, if I have to). The server obligingly replies with an SMTP informational message stating its name and the name and version of the software it is running (assuming that the banner it sends is real—I know of one service that deliberately lies in its banner, to misdirect hackers).

By the way, opening a Telnet session on port 25 or port 110 was the troubleshooting advice that an ISP's help desk gave me when I called to complain about problems retrieving my mail. They assumed that I really had an email account with them, I suppose.

A wealth of information really is obtainable about publicly accessible networks. And the reasonably skillful hackers start by getting as much of it as they can.

Network Commands

When a hacker has a notion of addresses and names, it's time to do some serious probing. Simple utilities built into the TCP/IP stack are where most hackers start:

- ping (more on this perennial favorite coming up)
- traceroute (or tracert, for Windows)
- whois
- finger
- telnet
- dig
- nmap
- nslookup
- rusers (and all the other Unix "r commands," such as `rlogin`, `rcp`, and `rexec`)
- rpcinfo

Some of these utilities are not found in systems running the various versions of Microsoft Windows, but they are generally present in Unix and Unix-derived systems (such as Linux, OpenBSD, or FreeBSD).

Ping Sweeps and Port Scans

Network mapping is much easier when it can be automated. People become bored and make silly mistakes manually repeating a command, but, of course, a computer never gets bored and (essentially) never makes mistakes (although code can become corrupted). Ping sweeps check for a reply (an ICMP Echo) from a series of addresses, and scripts to run such a sweep are easily located. Commercial versions are available for network management (as a way to discover unauthorized hosts surreptitiously added to your network); ping sweep scripts also are available at hacker Web sites (run an Internet search for "warez," for example).

Port scans test a given address across a range of commonly open ports or across a set range of ports (or even across all 65,535 TCP ports and 65,535 UDP ports, for those so inclined). Most hackers do not scan all possible ports, for two logical reasons. First, it takes some time, and repeated probes for an open port from a single address are likely to get the ISP a phone call, costing the hacker her access. Second, not every port is vulnerable, whereas some ports are almost always open—and putting the effort where the reward is likely to be found is as typical of a hacker as it is of anyone else.

Commonly probed ports are those used for networking, for typically installed services, and for services left operating by default installations of operating systems or applications. The Internet Storm Center is one site that reports on attack statistics. As this is being written, its most commonly probed ports and associated protocols are as follows (this list rarely changes):

- 135—epmap
- 1434—ms-sql-m
- 137—netbios-ns
- 80—www
- 445—microsoft-ds
- 1433—ms-sql-s
- 554—rtsp
- 139—netbios-ssn
- 21—ftp
- 1080—socks

The proportion of probes for each of these ports in North America might be different from the proportion on other continents, but the same ports

are generally probed everywhere. The end of the chapter includes a reference where you can check for the most current information in your area of interest.

Eavesdropping

Children are generally taught that eavesdropping is not polite, but they learn from experience that it can be a good way to learn things that you weren't supposed to know. That's true of networks as well. Just as with eavesdropping on a spoken conversation, network eavesdropping is the passive monitoring of information flowing back and forth. It is often performed by software collectively known as *packet sniffers*.

Packet sniffers work very simply: They place the network connection on their host in promiscuous mode, where it processes every packet on the wire instead of only those addressed to the physical address (or a configured multicast address). In a shared wire, such as classical Ethernet or a hubbed network, the traffic of many hosts can be observed. Sniffers can filter traffic to capture only those packets bound for certain ports where interesting information can be discovered (for example, traffic bound for POP3 connections might carry plain-text email passwords). Figure 3.1 shows the output of a popular sniffer, Ethereal.

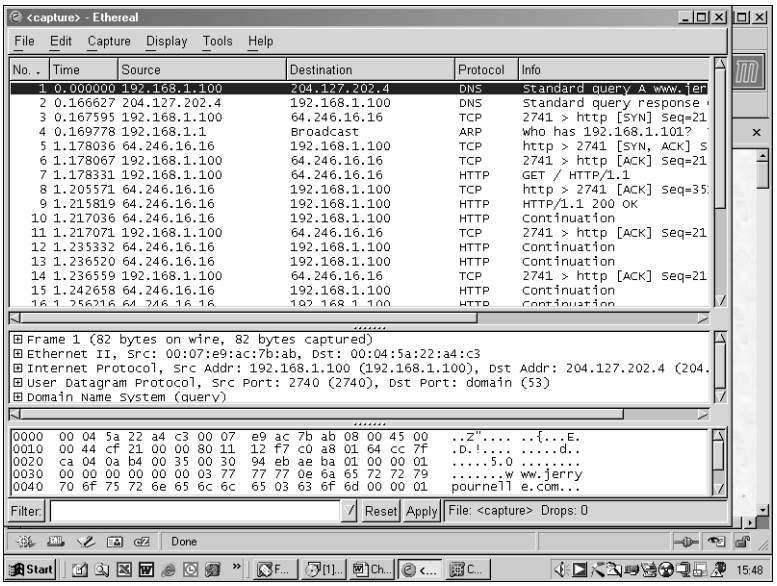

Figure 3.1 Web traffic as seen by Ethereal.

Notice that you can pick out a particular packet and examine it in more detail. The middle pane shows this to be a DNS request; notice that the "layering" of information is a reverse of the OSI model (the bottom is the application layer, while the physical-layer information is at the top). I also turned on name resolution so that, in the upper pane, I can see the hostname of the destination instead of the IP address.

All of this is a simple example of an extremely powerful tool and one that you might well see reference to on the exam. That is because sniffing traffic is a favorite technique of hackers. Packet sniffers are relatively small software packages, with low overhead, which makes them generally unnoticeable because they have little performance impact. This means that any compromised host on the network could become an unnoticed spy if the package could be installed surreptitiously.

Some mail clients actually send the mail server all information, including the account password, in plain text. Because many people use the same password for multiple purposes, this is a useful little item to capture. That's especially true because, in many cases, the email account name matches the logon name. If so, a simple packet capture reveals the username and password with which a hacker can log on, not just read someone's mail (as if that weren't bad enough).

Information Theft

Having discovered some useful information, such as hostnames and IP addresses, usernames and passwords, and possibly other information, it becomes time to make use of that information. Posing as a legitimate user, the hacker can copy confidential data, make changes (even worse—industrial espionage could become industrial sabotage), use the account to create another user account known only to the hacker, and so on. The last possibility means that the hacker now can steal information even if the current user changes his password or that user account is terminated.

Unauthorized Access

That leads to Cisco's second type of security threat: unauthorized access. However an unauthorized person gains access, when that person has even the most basic access, it is common to make efforts to escalate the access into that with more privileges. One means is to crack the master password list, such as the Unix/Linux /etc/shadow or the Windows NT SAM hive (actually, the passwords are not decrypted; instead, the cracking routine matches a retrieved hash by creating one using the same encryption algorithm known to be used by the OS). If that seems farfetched, utilities are available in the hacker community to do this for either OS. For passwords to offer real

instead of phantom protection, the organization must have and enforce a strong password policy.

The Unix "r commands"—`rlogin`, `rsh`, `rcp`—can be used to exploit trusted computing relationships among systems, just as trust relationships among Windows domains can be exploited. After getting into a system via a compromised account of exploited trust relationship, a hacker seeks to gain access to an account with more privileges (greater access, in terms of both scope of access and depth of access). This is known as *escalating privilege*. Escalating privilege, whether through obtaining the password or exploiting trust, is intended to gain the hacker control: the privileges of the Unix/Linux root account or the Windows Administrator account.

The CSI exam is neutral between the Windows operating system and the Unix and Unix-derived systems. Therefore, you need to be familiar with the security provisions of Windows, especially the servers, and those of the Unix community. Likewise, you need to be aware of the weaknesses of each and which protocols are more likely to be found working with which OS.

Another means of gaining unauthorized access is *social engineering*, the process of obtaining the information needed because people give it up on their own (sometimes just because they were asked for it). This exploits trust, too, but it is the trust we have of people rather than the trust among computer systems. Social-engineering attacks will never be prevented by technology because no technology is really involved.

Many of the protocols used in a network are insecure; they operate on an assumption of trust and do not protect the information they pass. Protocols such as Telnet, NTP, SNMP, and CDP can be exploited to manipulate the network devices, which are the nervous system of the organization. Cisco strongly recommends that, as much as possible, insecure protocols should be replaced with secure ones, such as replacing Telnet with SSH (Secure Shell). Network-management protocols should use the strongest version available, with authentication and/or encryption. Remember, unauthorized access to networking devices provides the hacker with far more opportunities for mischief than access to a single host. Therefore, control over access to routers and switches often needs to be stronger than that used to protect the average host.

Denial of Service

Of course, if the hacker can't get in, he can try to ensure that no one else can, either. This is the denial-of-service attack, usually just known as DoS. We

tend to think of DoS as flooding a line with packets; that is indeed one method of denying service. Others are to fill a buffer (such as filling a mail queue with spam) or to request more processes than the maximum allowed for a service, thereby cutting off that service from real users. These are all instances of *resource overload*; when a critical resource is overloaded, other (legitimate) users of the resource must be denied. Typical DoS attacks are listed here:

- TCP SYN flood
- Ping of death
- Tribe Flood Network (TFN) and Tribe Flood Network 2000 (TFN2K)
- Trinoo
- *Stacheldraht* (German for "barbed wire")
- Trinity

Although DoS attacks do not try to obtain information directly, bear in mind that, depending on which item crashes under the load, their effect can expose other resources that were previously protected.

One average desktop system can output a surprising number of packets with which to attack; combining attacks from several systems can overwhelm even robust networks. The combined attack is known as *distributed denial of service*, or DDoS. A simple form of DDoS attack sends a series of pings from several systems to a network broadcast address. The source IP for these pings is forged to appear to be that of the machine that is the actual DDoS target. The members of the network all obligingly reply to each ping they receive, and the TCP/IP stack of the target is overwhelmed with replies to a request that it did not send (but it must process each reply somewhat just to learn what it was, when the address reveals that this system is indeed the proper recipient of each packet). And that forging leads to the final threat type: data manipulation.

Data Manipulation

Data manipulation can take many forms. Because targets are not likely to take attacks upon them kindly, it is common to manipulate the source IP address of a packet sent by a hacker. This is surprisingly easy to do; Figure 3.2 is a screenshot of NMapWin, a Windows version of the nmap tool. Notice the opportunity (in the middle right of the GUI) to forge the source IP address.

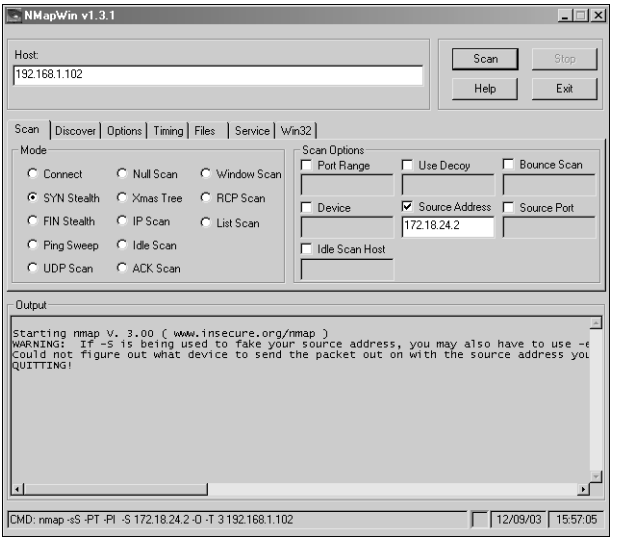

Figure 3.2 Spoofing the source IP address.

Conveniently, NMapWin even tells you the command-line syntax to use to implement this the old-fashioned way (look at the status bar along the bottom). IP headers are not the only address elements forged; much spam is created with false email headers to make tracing the spammer more difficult.

Another form of data manipulation is the man-in-the-middle attack, in which a system interposes itself between the two parties in a dialogue, pretending to each of them that it is actually the other party rather than a third party. In this manner, the third party receives both sides of the conversation and can manipulate what each intended recipient sees. The idea behind this is shown in Figure 3.3.

Somewhat similar is the *session replay* or *session hijacking* form of data-manipulation attack. The difference here is that the hacker is the endpoint of your conversation, even though he pretends to be the other party (the other party isn't actually involved). A session replay reuses information, some of which can be altered, to create a new action (a new payment, for instance, in a larger amount or to a new payee). Session hijacking redirects traffic in an actual system data exchange, diverting it from the real other party to a hacker posing as that party. This can be done by noting the pattern of TCP sequence numbers and their change, and then interposing a packet with the predicted sequence number.

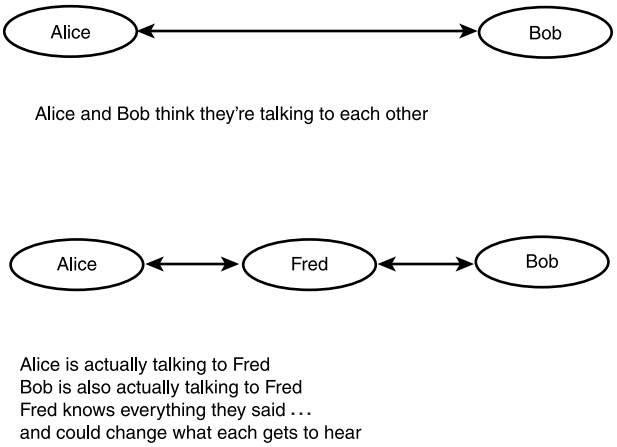

Figure 3.3 Man-in-the-middle attack.

Finally, data can be diverted from its proper destination to a bogus one, such as a Web browser session being diverted to a good forgery of a Web page (*rerouting*); likewise, one party to an actual transaction might deny that the transaction ever took place. The latter is called *repudiation* and can be done to prevent a third party (such as the debt collector) from proving that the transaction occurred. Although the data itself is not necessarily manipulated, its validity as it currently exists is questioned because it cannot be proven that it was not manipulated. This makes repudiation a case of requiring proof of a non-event, that there was no attack. This is much more difficult to prove, and the methods for handling nonrepudiation ensure that it is reasonably provable.

Summary

Whether you prefer to classify threats by their location—internal or external—or by their type—reconnaissance, unauthorized access, denial of service, or data manipulation—depends on your preference as much as it depends on what you see happening in your network. When it comes time to do something about the threat, however, the first thing you need is a security policy, so that's where we'll turn next.