

Suspending Accounts

Another important practice in account management is to suspend accounts that are no longer needed. Every large organization eventually stumbles across one or more accounts that belong to users who are no longer part of the organization. In extreme cases, an organization discovers that a user who left several months ago still has privileged accounts. The unfettered presence of these accounts on our networks gives adversaries a powerful means to become seemingly legitimate users, which makes our job of detecting and repulsing them that much more difficult.

Accounts may become unneeded, and thus require suspension, for a variety of reasons, but perhaps the most common one would be that the user of the account was terminated or otherwise left the organization. Other reasons for suspension include reaching the account's default expiration date, and temporary, but extended, absences of employees (e.g., maternity leave, military deployment). Whatever the reason, we must ensure that the account of someone who is not present to use it is suspended until that person returns or the term of our retention policy is met.

Testing the administrative controls on suspended accounts follows the same pattern already laid out in the preceding two sections: look at each account (or take a representative sample of all of them) and compare it with the status of its owner according to our HR records. Alternatively, we can get a list of employees who are temporarily or permanently away from the organization and check the status of those accounts. It is important that accounts are deleted only in strict accordance with the data retention policy. Many investigations into terminated employees have been thwarted because administrators have prematurely deleted user accounts and/or files.

Backup Verification

Modern organizations deal with vast amounts of data, which must be protected for a variety of reasons, including disaster recovery. We have all been in at least one situation in which we have lost data and needed to get it back. Some of us have had a rude awakening upon discovering that the data was lost permanently. The specific nature of the backup media is not as important as the fact that the data must be available when we need it most.

Magnetic tapes are now able to hold over 180 terabytes of data, which makes this seemingly antiquated technology the best in terms of total cost of ownership. That being said, many organizations prefer other technologies for daily operations, and relegate tapes to the role of backup to the backup. In other words, it is not uncommon for an organization to back up its user and enterprise data to a storage area network (SAN) on a daily basis, and back up these backups to tape on a weekly basis. Obviously, the frequency of each backup (hourly, daily, weekly) is driven by the risk management process discussed in Chapter 1.

Whatever the approach to backing up our organizational data, we need to periodically test it to ensure that the backups will work as promised when we need them. There are some organizations that have faced an event or disaster that required them to restore some or all data from backups, only to discover that the backups were missing, corrupted, or outdated. This section discusses some approaches to assess whether the data will be there when we need it.



CAUTION Never back up your data to the same device on which the original data exists.

Types of Data

Not all data is created equal, and different types may have unique requirements when it comes to backups. The following sections discuss some of the major categories of data that most of us deal with and some considerations when planning to preserve that data. Keep in mind, however, that there are many other types of data that we will not discuss here for the sake of brevity.

User Data Files This is the type of data with which most of us are familiar. These are the documents, presentations, and spreadsheets that we create or use on a daily basis. Though backing up these files may seem simple, challenges arise when users put “backup” copies in multiple locations for safekeeping. Users, if left to their own devices, may very well end up with inconsistently preserved files and may even violate retention requirements. The challenge with this type of data is ensuring that it is consistently backed up in accordance with all applicable policies, regulations, and laws.

Databases Databases are different from regular files in that they typically store the entire database in a special file that has its own file system within it. To make sense of this embedded file system, your database software uses metadata that lives in other files within your system. This architecture can create complex interdependencies among files on the database server. Fortunately, all major database management systems (DBMSs) include one or more means to back up their databases. The challenge is in ensuring that the backup will be sufficient to reconstitute the databases if necessary. To verify the backups, many organizations use a test database server that is periodically used to verify that the databases can be recovered from backup and that the queries will execute properly from the restored data.

Virtualization as a Backup and Security Strategy

Many organizations have virtualized their server infrastructure for performance and maintenance reasons. Some are also virtualizing their client systems and turning their workstations into thin clients on a virtualization infrastructure. The next step in this evolution is the use of virtual machine (VM) snapshots as a backup strategy. The main advantage to this approach is that restoration is almost instantaneous. All you typically have to do is click a button or issue a scripted command and the VM will revert to the designated state. Another key advantage is that this approach lends itself to automation and integration with other security systems so that if, for example, a workstation is compromised because the user clicked a link and an IDS detected this incident, then the VM can be instantly quarantined for later analysis while the user is dropped into the most recent snapshot automatically with very little impact to productivity.

Mailbox Data By some estimates, as much as 75 percent of an average organization's data lives in its mailboxes. Depending on the mail system your organization is running, the backup process may be very different. Still, some commonalities exist across all platforms, such as the critical need to document in excruciating detail every aspect of the configuration of the mail servers. Most medium-sized to large organizations have multiple mail servers (perhaps backing each other up), so it is a good idea not to back them up at the same time. Finally, whatever backup mechanism you have in place for your mail servers should facilitate compliance with e-discovery.

Verification

Having data backups is not particularly helpful unless we are able to use them to recover from mistakes, accidents, attacks, or disasters. Central to verifying this capability is understanding the sorts of things that can go wrong and which of them would require backups. Recall from our discussion on threat modeling in Chapter 9 that an important step in understanding risk is to consider what can happen or be done to our systems that would destroy, degrade, or disrupt our ability to operate. It is helpful to capture these possibilities in scenarios that can then inform how we go about ensuring that we are prepared for the likely threats to our information systems. It is also helpful to automate as much of the testing as possible, particularly in large organizations. This ensures that we cover the likely contingencies in a very methodical and predictable manner.

Some tests may cause disruptions to our business processes. It is difficult to imagine how a user's backups can be fully tested without involving that user in the process to some extent. If, for instance, our users store files locally and we want to test Mary's workstation backup, an approach could be to restore her backup to a new computer and have Mary log into and use the new computer as if it were the original. She would be in a better position than anyone else to determine whether everything works as expected. This kind of thorough testing is expensive and disruptive, but it ensures that we have in place what we need. Obviously, we have to be very selective about when and how we impact our business processes, so it becomes a trade-off.

However you decide to implement your backup verification, you must ensure that you are able to assert that all critical data is backed up and that you will be able to restore it in time of need. This means that you probably have to develop an inventory of data and a schedule for testing it as part of your plan. This inventory will be a living document, so you must have a means to track and document changes to it. Fortunately, major items such as mail and database servers don't change very frequently. The challenge is in verifying the backups of user data.

This brings us back to our policies. We already discussed the importance of the organization's data retention policy, but an equally important one is the policy that dictates how user data is backed up. Many organizations require their staff to maintain their files on file shares on network servers, but we all know that users don't necessarily always do this. It is not uncommon for users to keep a local folder with the data that is most important to them. If the local files are not being backed up, then we risk losing the most critical files, particularly if backups can be disabled by the user. The point of this is that policies need to be carefully thought out and aggressively enforced if we are to be ready for the day when things go badly for us.

Testing Data Backups

It is important to develop formal processes for testing your data backups to ensure they are available when needed. The following are some elements that should be included in these processes:

- *Develop scenarios* that capture specific sets of events that are representative of the threats facing the organization.
- *Develop a plan* that tests all the mission-critical data backups in each of the scenarios.
- *Leverage automation* to minimize the effort required by the auditors and ensure tests happen periodically.
- *Minimize impact on business* processes of the data backup test plan so that it can be executed regularly.
- *Ensure coverage* so that every system is tested, though not necessarily in the same test.
- *Document the results* so you know what is working and what needs to be worked on.
- *Fix or improve* any issues you documented.

Security Training and Security Awareness Training

As should be clear from the preceding discussions, having a staff that is well trained in security issues is crucial to the security of our organizations. The terms security training and security awareness training are often used interchangeably, but they have subtly different meanings. *Security training* is the process of teaching a skill or set of skills that enables people to perform specific security functions better. *Security awareness training*, on the other hand, is the process of exposing people to security issues so that they are able to recognize and respond to them better. Security training is typically provided to security personnel, while security awareness training should be provided to every member of the organization.

Assessing the effectiveness of our security training programs is fairly straightforward because the training is tied to specific security functions. Therefore, to test the effectiveness of a training program, all we have to do is test the performance of an individual on those functions before and after the training. If the performance improves, then the training was probably effective. Keep in mind that skills atrophy over time, so the effectiveness of the training should be measured immediately after it concludes. Otherwise, we are assessing the long-term retention of the functional skills.

We now turn our attention to the somewhat more difficult issue of assessing the effectiveness of a security awareness training program. As we broach this subject, keep in mind that the end state is to better equip our teammates to recognize and deal with

security issues that arise while they are performing their everyday tasks. This implies that a key measure of the effectiveness of the security awareness program is the degree to which people change their behaviors when presented with certain situations. If this change is toward a better security posture, then we can infer that the program was effective. In the following sections, we take a look at specific components of a security awareness training program that are common to many organizations.



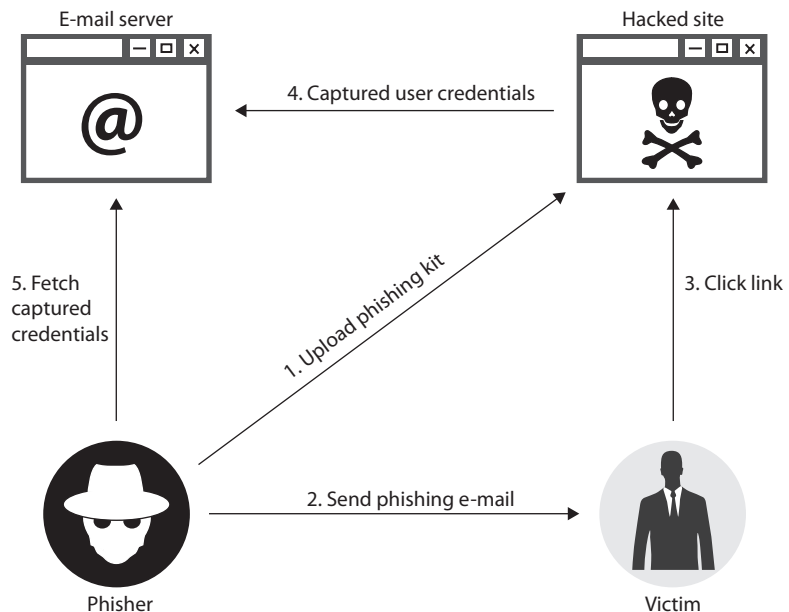
EXAM TIP Security awareness (and the training required to attain it) is one of the most critical controls in any ISMS. Expect exam questions on this topic.

Social Engineering

Social engineering, in the context of information security, is the process of manipulating individuals so that they perform actions that violate security protocols. Whether the action is divulging a password, letting someone into the building, or simply clicking a link, it has been carefully designed by the adversaries to help them exploit our information systems. A common misconception is that social engineering is an art of improvisation. While improvising may help the attacker better respond to challenges, the reality is that most effective social engineering is painstakingly designed against a particular target, sometimes a specific individual.

Perhaps the most popular form of social engineering is *phishing*, which is social engineering conducted through a digital communication. Figure 19-2 depicts the flow of a typical e-mail phishing attack. (While e-mail phishing receives a lot of attention, text messages can also be used to similar effect.) Like casting a baited fishing line into a

Figure 19-2
Typical phishing attack



pond full of fish, phishing relies on the odds that if enough people receive an enticing or believable message, at least one of them will click an embedded link within it.

Some adversaries target specific individuals or groups, which is referred to as *spear-phishing*. In some cases, the targets are senior executives, in which case it is called *whaling*. In whatever variety it comes, the desired result of phishing is usually to have the target click a link that will take them to a website under the control of the attacker. Sometimes the website looks like the legitimate logon page of a trusted site, such as that of the user's bank. Other times, the website is a legitimate one that has been compromised by the attacker to redirect users somewhere else. In the case of a *drive-by download*, the site invisibly redirects the user to a malware distribution server, as shown in Figure 19-3.

Pretexting is a form of social engineering, typically practiced in person or over the phone, in which the attacker invents a believable scenario in an effort to persuade the target to violate a security policy. A common example is a call received from (allegedly) customer service or fraud prevention at a bank in which the attacker tries to get the target to reveal account numbers, personal identification numbers (PINs), passwords, or similarly valuable information. Remarkably, pretexting was legal in the United States until 2007, as long as it was not used to obtain financial records. In 2006, Hewlett-Packard became embroiled in a scandal dealing with its use of pretexting in an effort to identify the sources of leaks on its board of directors. Congress responded by passing the Telephone Records and Privacy Protection Act of 2006, which imposes stiff criminal penalties on anyone who uses pretexting to obtain confidential information.

So how does one go about assessing security awareness programs aimed at countering social engineering in all its forms? One way is to keep track of the number of times users fall victim to these attacks before and after the awareness training effort. The challenge with this approach is that victims may not spontaneously confess to falling for

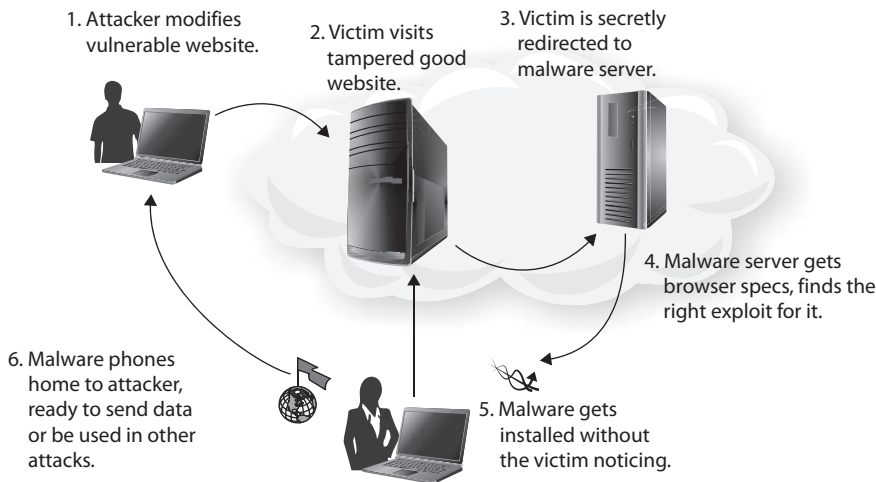


Figure 19-3 Drive-by downloads

these tricks, and our security systems will certainly not detect all instances of successful attacks. Another approach is to have auditors (internal or external) conduct benign social engineering campaigns against our users. When users click a link inserted by the auditors, they are warned that they did something wrong and perhaps are redirected to a web page or short video explaining how to avoid such mistakes in the future. All the while, our automated systems are keeping tabs on which users are most susceptible and how often these attacks are successful. Anecdotal evidence suggests that there is a group of users who will not respond to remedial training, so the leadership should decide what to do with individuals who repeatedly make the wrong choices.

Online Safety

Oftentimes users don't have to be tricked into doing something wrong, but willingly go down that path. This is often the result of ignorance of the risks, and the remediation of this ignorance is the whole point of the security awareness campaign. An effective security awareness program should include issues associated with unsafe online behavior that could represent risk for the organization.

Perhaps one of the most important elements of safe online behavior is the proper use of social media. A good starting point is the proper use of privacy settings, particularly considering that all major social media sites have means to restrict what information is shared with whom. The default settings are not always privacy-focused, so it is important for users to be aware of their options. This becomes particularly important when users post information concerning their workplace. Part of the security awareness program should be to educate users about the risks they can pose to their employers if their posts reveal sensitive information. Once posted, the information cannot be recalled; it is forevermore out there.

Sometimes it is not what goes out to the Internet but what comes in from it that should concern users. Simply surfing to the wrong website, particularly from a workplace computer, may be all it takes to bring down the whole organization. In the case of a drive-by download, the attack is triggered simply by visiting a malicious website. While the mechanisms vary, the effect can be the execution of malware on the client computer, with or without additional user interaction. While web filters can mitigate some of the risk of surfing to inappropriate sites, malicious websites sometimes are legitimate ones that have been compromised, which means that the filters may not be effective.

While some downloads happen without user knowledge or interaction, others are intentional. It is not unusual for naïve users to attempt to download and install unauthorized and potentially risky applications on their computers. Unfortunately, many organizations do not use software whitelisting and even allow their users to have administrative privileges on their computers, which allows them to install any application they desire. Even benign applications can be problematic for the security of our systems, but when you consider that the software may come from an untrusted and potentially malicious source, the problem is compounded.

Assessing the effectiveness of an awareness campaign that promotes users' online safety is not easy and typically requires a multipronged approach. Social media posts may be detected using something as simple as Google Alerts, which trigger whenever Google's

robots find a term of interest online. A simple script can then filter out the alerts by source in order to separate, say, a news outlet report on our organization from an ill-advised social media post. The software download problem (whether intentional or not) can be assessed by a well-tuned IDS. Over time, with an effective awareness campaign, we should see the number of incidents go down, which will allow us to focus our attention on repeat offenders.

Data Protection

We already covered data protection in Chapter 6, but for the purposes of assessing a security awareness program, it bears repeating that sensitive data must always be encrypted whether at rest or in transit. It is possible for users to circumvent controls and leave this data unprotected, so awareness is a key to preventing this type of behavior. Unencrypted data is vulnerable to leaks if it is stored in unauthorized online resources or intentionally (but perhaps not maliciously) shared with others. Another important topic is the proper destruction of sensitive data when it is no longer needed and falls out of the mandatory retention period (see Chapter 5).

Testing the degree to which our users are aware of data protection requirements and best practices can best be done by using tags in our files' metadata. The information classification labels we discussed in Chapter 5 become an effective means of tracking where our data is. Similarly, data loss prevention (DLP) solutions can help stop leaks and identify individuals who are maliciously or inadvertently exposing our sensitive information. This allows us to target those users either with additional awareness training or with disciplinary actions.

Culture

At the end of the day, the best way to test the security awareness of an organization may be by assessing its security culture. Do we have the kind of environment in which users feel safe self-reporting? Are they well incentivized to do so? Do they actively seek information and guidance when encountering a strange or suspicious situation? Self-reports and requests for information by users provide a good indicator of whether the organizational culture is helping or hindering us in securing our systems.

Disaster Recovery and Business Continuity

Most organizations cannot afford to be incapable of performing their business processes for very long. Depending on the specific organization, the acceptable downtime can be measured in minutes, hours, or, in some noncritical sectors, maybe days. Consequently, we all need to have procedures in place for ensuring we can go on working regardless of what happens around or to us. As introduced in Chapter 2, *business continuity* is the term used to describe the processes enacted by an organization to ensure that its vital business processes remain unaffected or can be quickly restored following a serious incident. Business continuity looks holistically at the entire organization. A subset of this effort, called *disaster recovery*, focuses on restoring the information systems after a disastrous event. Like any other business process, these processes must be periodically assessed to ensure they are still effective.

Often, the initial response to an emergency affects the ultimate outcome. Emergency response procedures are the prepared actions that are developed to help people in a crisis situation better cope with the disruption. These procedures are the first line of defense when dealing with a crisis situation. People who are up to date on their knowledge of these procedures will perform the best, which is why training and drills are very important. Emergencies are unpredictable, and no one knows when they will be called upon to perform their disaster recovery duties.

Protection of life is of the utmost importance and should be dealt with first before attempting to save material objects. Emergency procedures should show the people in charge how to evacuate personnel safely (see Table 19-1). All personnel should know their designated emergency exits and destinations. Emergency gathering spots should take into consideration the effects of seasonal weather. One person in each designated group is often responsible for making sure all people are accounted for. One person in particular should be responsible for notifying the appropriate authorities: the police department, security guards, fire department, emergency rescue, and management. With proper training, employees will be better equipped to handle emergencies and avoid the reflex to just run to the exit.



EXAM TIP Protection of human life is always the top priority in situations where it is threatened.

If the situation is not life threatening, designated staff should shut down systems in an orderly fashion, and remove critical data files or resources during evacuation for safekeeping. There is a reason for the order of activities. As with all processes, there are

Procedure: Personnel Evacuation Description	Location	Names of Staff Trained to Carry Out Procedure	Date Last Carried Out
Each floor within the building must have two individuals who will ensure that all personnel have been evacuated from the building after a disaster. These individuals are responsible for performing employee head count, communicating with the business continuity plan (BCP) coordinator, and assessing emergency response needs for their employees.	West wing parking lot	David Miller Michelle Lester	Drills were carried out on May 4, 2021.
Comments: These individuals are responsible for maintaining an up-to-date listing of employees on their specific floor. These individuals must have a company-issued walkie-talkie and proper training for this function.			

Table 19-1 Sample Emergency Response Procedure

dependencies with everything we do. Deciding to skip steps or add steps could in fact cause more harm than good.

Once things have approached a reasonable plateau of activity, one or more people will most likely be required to interface with external entities, such as the press, customers, shareholders, and civic officials. One or more people should be prepped in their reaction and response to the recent disaster so a uniform and reasonable response is given to explain the circumstances, how the organization is dealing with the disaster, and what customers and others should now expect from the organization. The organization should quickly present this information instead of allowing others to come to their own conclusions and start false rumors. At least one person should be available to the press to ensure proper messages are being reported and sent out.

Another unfortunate issue needs to be addressed prior to an emergency: potential looting, vandalism, and fraud opportunities from both a physical perspective and a logical perspective. After an organization is hit with a large disturbance or disaster, it is usually at its most vulnerable, and others may take advantage of this vulnerability. Careful thought and planning, such as provision of sufficient security personnel on site, enable the organization to deal with these issues properly and provide the necessary and expected level of protection at all times.

Ideally, we collect most of the data we need for assessing our disaster recovery and business continuity processes before any real emergencies arise. This allows us to ensure we are prepared and to improve the effectiveness of our organizational responses to these unforeseen events. Still, the best data is captured during an actual emergency situation. After any real or training events, it is imperative that we have a debriefing immediately after it. This event, sometimes called a *hot wash*, must happen while memories are still fresh. It is an ad hoc discussion of what happened, how we dealt with it, what went well, and how we can do better in the future. Ideally, it is followed by a more deliberate *after-action review (AAR)* that takes place later, once the stakeholders have had a chance to think through the events and responses and analyze them in more detail. Hot wash notes and AAR reports are excellent sources of security process data for disaster recovery and business continuity.

Reporting

For many security professionals, report writing is perhaps one of the least favorite activities, and yet it is often one of the most critical tasks for our organizations. While we all thrive on putting hands on keyboards and patch panels when it comes to securing our networks, we often cringe at the thought of putting in writing what it is that we've done and what it means to the organization. This is probably the task that best distinguishes the true security professional from the security practitioner: the professional understands the role of information systems security within the broader context of the business and is able to communicate this to both technical and nontechnical audiences alike.

It seems that many of us have no difficulty (though perhaps a bit of reluctance) describing the technical details of a plan we are proposing, a control we have implemented, or an audit we have conducted. It may be a bit tedious, but we've all done this at some

point in our careers. The problem with these technical reports, important though they are, is that they are written by and for technical personnel. If your CEO is a technical person running a technical company, this may work fine. However, sooner or later most of us will work with decision-makers that are not inherently technical. These leaders will probably not be as excited about the details of an obscure vulnerability you just discovered as they will be about its impact on the business. If you want your report to have a business impact, it must be both technically sound and written in the language of the business.

Analyzing Results

Before you start typing that report, however, you probably want to take some time to review the outputs, ensure you understand them, and then infer what they mean to your organization. Only after analyzing the results can you provide insights and recommendations that will help maintain or improve your organization's security.

The goal of this analysis process is to move logically from facts to actionable information. A list of vulnerabilities and policy violations is of little value to business leaders unless it is placed in context. Once you have analyzed all the results in this manner, you'll be ready to start writing the official report.

You can think of analyzing results as a three-step process to determine the following: What?, So what?, and Now what? First you gather all your data, organize it, and study it carefully. You find out *what* is going on. This is where you establish the relevant and interesting facts. For example, you may have determined the fact that 12 of your servers are not running on the latest software release. Worse yet, you may have found that three of those servers have vulnerabilities that are being exploited in the wild. The instinctive reaction of many would be to say this is a big deal that needs to be corrected immediately. But wait.

The second step in your analysis is to determine the business impact of those facts. This is the *so what?* Though we tend to focus on the technology and security aspects of our environments, we have a responsibility to consider facts in a broader organizational context. Continuing with the previous example, you may find that those 12 servers provide a critical business function and cannot be updated in the near term for perfectly legitimate operations reasons. You may also discover that you already have compensatory administrative or technical controls that mitigate the risk they pose. So maybe it's not that big of a deal after all.

The third step is to figure out the *now what?* The whole point of measuring security is to ensure it is sufficient or to improve it so that it is sufficient. The analysis process leads to results, and these are only valuable if they are actionable. They must point to one or more sound recommendations that address the broader organizational needs. In our example, you clearly don't want to leave those servers as they are indefinitely. Maybe you have considered two courses of action: either leave things as they are but reassess every 30 days or update the servers immediately despite the resulting business impact. You evaluate the alternatives using risk and business impact as decision criteria and ultimately decide that keeping an extra-close eye on the unpatched servers for a few more weeks is the better course of action. You put down a date for the next decision point and go from there. The point is that your decision is based on a sound analysis of the facts.

Remediation

Most assessments uncover vulnerabilities. While many cybersecurity practitioners think of vulnerabilities in terms of software defects to be patched, the reality is that most vulnerabilities in the average organization tend to come from misconfigured systems, inadequate policies, unsound business processes, or unaware staff. Correcting most of these vulnerabilities requires engagement by more than just the IT or security teams. Even the more mundane system patches need to be carefully coordinated with all affected departments within the organization. Vulnerability remediation should include all stakeholders, especially those who don't have the word "security" anywhere in their job titles.

The fact that you're leveraging a multifunctional extended team to remediate vulnerabilities highlights the need for the sound analyses described in the previous section. You'll need the support of everyone from the very top of the organization on down, which is why you want to educate them on your findings, why they are impacted, and what you must all do about them. It is likely that remediation will impact the business, so it is also critical to have contingency plans and be able to handle exceptional cases.

Exception Handling

Sometimes, vulnerabilities simply can't be patched (at least, not in any reasonable amount of time). Some of us have dealt with very big and expensive medical devices that require Food and Drug Administration accreditations that preclude their patching without putting them through an expensive and time-consuming recertification process. The solution is to implement compensatory controls around the problem, document the exception, and revisit the vulnerability over time to see if can be remediated directly at some point in the future. For example, a medical device may be micro-segmented in its own VLAN behind a firewall that would only allow one other device to communicate with it, and then using only a specific port and protocol.

The Language of Your Audience

You cannot be an effective communicator if you don't know your audience. Learning to speak the language(s) of those you are trying to inform, advise, or lead is absolutely critical. It has been said that accounting is the language of business, which means you can generally do well communicating in terms of the financial impacts of your findings. The fact that risks are expressed as the probability of a certain amount of loss should make this fairly easy as long as you have some sort of risk management program in place.

Still, in order to up your game, you want to be able to communicate in the language of the various disciplines that make up a business. Human resource leaders will care most about issues like staff turnover and organizational culture. Your marketing (or public affairs) team will be focused on what external parties think about your organization. Product managers will be very reluctant to support proposals that can slow down their delivery tempo. We could go on, but the point is that, while the facts and analyses must be unassailable, you should always try to communicate them in the language of...whoever it is you're trying to persuade.

Ethical Disclosure

Occasionally, security assessments lead to discoveries of vulnerabilities that were not known and which affect other organizations. Perhaps you were performing a code review on one of the products your company sells and you discovered a vulnerability, or maybe your pentesting team was conducting a pen test on a system your organization bought from one of its vendors and they found a previously unknown way to exploit the system. However you discover the vulnerability, you have an ethical obligation to properly disclose it to the appropriate parties. If the vulnerability is in your own product, you need to notify your customers and partners as soon as possible. If it is in someone else's product, you need to notify the vendor or manufacturer immediately so they can fix it. The goal of ethical disclosure is to inform anyone who might be affected as soon as feasible, so a patch can be developed before any threat actors become aware of the vulnerability.

More commonly, exception handling is required because something crashed while we were attempting to patch a system. Though we should always test patches in a sandbox environment before pushing them out to production systems, we can never be 100 percent certain that something won't go wrong. In those cases, particularly if the system is mission-critical, we roll back the patch, get the system back online as quickly and securely as we can, document the exception, and move on with remediation of other systems. We circle back, of course, but exception handling is typically a time-intensive effort that should not delay the larger remediation effort.

Writing Technical Reports

After analyzing the assessment results, the next step is to document. A technical report should be much more than the output of an automated scanning tool or a generic checklist with yes and no boxes. There are way too many so-called auditors that simply push the start button on a scanning tool, wait for it to do its job, and then print a report with absolutely none of the analysis we just discussed.

A good technical report tells a story that is interesting and compelling *for its intended audience*. It is very difficult to write one without a fair amount of knowledge about its readers, at least the most influential ones. Your goal, after all, is to persuade them to take whatever actions are needed to balance risks and business functions for the betterment of the organization. Simultaneously, you want to anticipate likely objections that could undermine the conversation. Above all else, you must be absolutely truthful and draw all conclusions directly from empirical facts. To improve your credibility, you should always provide in an appendix the relevant raw data, technical details, and automated reports.

The following are key elements of a good technical audit report:

- **Executive Summary** We'll get into the weeds of this in the next section, but you should always consider that some readers may not be able to devote more than a few minutes to your report. Preface it with a hard-hitting summary of key take-aways.

- **Background** Explain why you conducted the experiment/test/assessment/audit in the first place. Describe the scope of the event, which should be tied to the reason for doing it in the first place. This is a good place to list any relevant references such as policies, industry standards, regulations, or statutes.
- **Methodology** As most of us learned in our science classes, experiments (and audits) must be repeatable. Describe the process by which you conducted the study. This is also a good section in which to list the personnel who participated, dates, times, locations, and any parts of the system that were excluded (and why).
- **Findings** You should group your findings to make them easier to search and read for your audience. If the readers are mostly senior managers, you may want to group your findings by business impact. Technologists may prefer groupings by class of system. Each finding should include the answer to “so what?” from your analysis.
- **Recommendations** This section should mirror the organization of your findings and provide the “now what?” from your analysis. This is the actionable part of the report, so you should make it compelling. When writing it, you should consider how each key reader will react to your recommendations. For instance, if you know the CFO is reluctant to make new capital investments, then you could frame expensive recommendations in terms of operational costs instead.
- **Appendices** You should include as much raw data as possible, but you certainly want to include enough to justify your recommendations. Pay attention to how you organize the appendices so that readers can easily find whatever data they may be looking for.

If you are on the receiving end of this process, always be wary of reports that look auto-generated, which usually points to an ineffective auditing team. Also be careful about reports that, having failed to find any significant vulnerabilities, overemphasize the importance of less important flaws. If the security posture of the organization is good, then the auditors should not shy away from saying so.

Executive Summaries

Getting into the technical weeds with an audit report is wonderful for techies, but it doesn't do the business folks any good. The next step in writing impactful reports is to translate the key findings and recommendations into language that is approachable and meaningful to the senior leadership of your organization. After all, it is their support that will allow you to implement the necessary changes. They will provide both the authority and resources that you will need.

Typically, technical reports (among others) include an executive summary of no more than a page or two, which highlights what senior leaders need to know from the report. The goal is to get their attention and effect the desired change. One way to get a business leader's attention is to explain the audit findings in terms of risk exposure. Security is almost always perceived as a cost center for the business. A good way to show return on investment (ROI) for a department that doesn't generate profits is by quantifying how much money a recommended change could potentially save the company.

One way to quantify risk is to express it in monetary terms. We could say that the risk (in dollars) is the value of an asset multiplied by the probability of the loss of that asset. In other words, if our customer's data is worth \$1 million and there is a 10 percent chance that this data will be breached, then our risk for this data breach would be \$100,000. How can we come up with these values? There are different ways in which accountants value other assets, but the most common are the following.

- The *cost approach* simply looks at the cost of acquiring or replacing the asset. This is the approach we oftentimes take to valuating our IT assets (minus information, of course). How might it be applied to information? Well, if an information asset is a file containing a threat intelligence report that cost the organization \$10,000, then the cost approach would attach that value to this asset.
- The *income approach* considers the expected contribution of the asset to the firm's revenue stream. The general formula is value equals expected (or potential) income divided by capitalization rate. The capitalization rate is the actual net income divided by the value of the asset. So, for instance, if that \$10,000 threat intelligence report brought in \$1,000 in net income last year (so the capitalization rate is 0.10) and our projections are that it will bring in \$2,000 this year, then its present value would be $\$2,000 \div 0.10$, or \$20,000. As you should be able to see, the advantage of this approach is that it takes into account the past and expected business conditions.
- The *market approach* is based on determining how much other firms are paying for a similar asset in the marketplace. It requires a fair amount of transparency in terms of what other organizations are doing. For instance, if we have no way of knowing how much others paid for that threat intelligence report, then we couldn't use a market approach to valuating it. If, on the other hand, we were able to find out that the going rate for the report is actually \$12,000, then we can use that value for our report (asset) and celebrate that we got a really good deal.

So, as long as the life-cycle costs of implementing our proposed controls (say, \$180,000) are less than the risks they mitigate (say, \$1,000,000), it should be obvious that we should implement the control, right? Not quite. The controls, after all, are not perfect. They will not be able to eliminate the risk altogether, and will sometimes fail. This means that we need to know the likelihood that the control will be effective at thwarting an attack. Let's say that we are considering a solution that has been shown to be effective about 80 percent of the time and costs \$180,000. We know that we have a 10 percent chance of being attacked and, if we are, that we have a 20 percent chance of our control failing to protect us. This means that the residual risk is 2 percent of \$1,000,000, or \$20,000. This is then added to the cost of our control (\$180,000) to give us the total effective cost of \$200,000.

This is the sort of content that is impactful when dealing with senior leaders. They want to know the answers to questions such as these: How likely is this control to work? How much will it save us? How much will it cost? The technical details are directly

important to the ISMS team and only indirectly important to the business leaders. Keep that in mind the next time you package an audit report for executive-level consumption.

Management Review and Approval

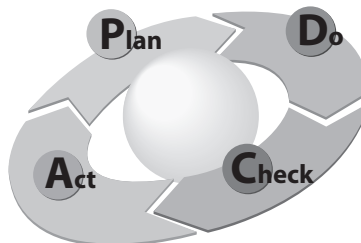
A management review is a formal meeting of senior organizational leaders to determine whether the management systems are effectively accomplishing their goals. In the context of the CISSP, we are particularly interested in the performance of the ISMS. While we restrict our discussion here to the ISMS, you should be aware that the management review is typically much broader in scope.

While management reviews have been around for a very long time, the modern use of the term is perhaps best grounded in quality standards such as the ISO 9000 series. These standards define a Plan-Do-Check-Act loop, depicted in Figure 19-4. This cycle of continuous improvement elegantly captures the essence of most topics we cover in this book. The Plan phase is the foundation of everything else we do in an ISMS, because it determines our goals and drives our policies. The Do phase of the loop is the focal point of Part VII of this book (“Security Operations”). The Check phase is the main topic of this chapter and the previous one. Lastly, the Act phase is what we formally do in the management review. We take all the information derived from the preceding stages and decide whether we need to adjust our goals, standards, or policies in order to continuously improve our posture.

The management review, unsurprisingly, looks at the big picture in order to help set the strategy moving forward. For this reason, a well-run review will not be drawn into detailed discussions on very specific technical topics. Instead, it takes a holistic view of the organization and makes strategic decisions, which is the primary reason why the management review must include all the key decision makers in the organization. This top-level involvement is what gives our ISMS legitimacy and power.

When communicating with senior executives, it is important to speak the language of the business and to do so in a succinct manner. We already discussed this style of communication when we covered reports in the previous section, but it bears repeating here. If we are not able to clearly and quickly get the point across to senior leaders on the first try, we may not get another chance to do so.

Figure 19-4
The Plan-Do-Check-Act loop



Before the Management Review

The management review should happen periodically. The more immature the management system and/or the organization, the more frequent these reviews should take place. Obviously, the availability of the key leaders will be a limiting factor during scheduling. This periodicity helps ensure that the entire organization is able to develop an operational rhythm that feeds the senior-level decision-making process. Absent this regularity, the reviews risk becoming reactive rather than proactive.

The frequency of the meetings should also be synchronized with the length of time required to implement the decisions of the preceding review. If, for instance, the leaders decided to implement sweeping changes that will take a year to develop, integrate, and measure, then having a review before the year is up may not be particularly effective. This is not to say that enough time must lapse to allow every single change to yield measurable results, but if these reviews are conducted too frequently, management won't be able to make decisions that are informed by the results of the previous set of actions.

Reviewing Inputs

The inputs to the management review come from a variety of sources. A key input is the results of relevant audits, both external and internal. These are, in part, the reports described earlier in the chapter. In addition to making the audit reports available for review, it is also necessary to produce executive summaries that describe the key findings, the impact to the organization, and the recommended changes (if any). Remember to write these summaries in business language.

Another important input to the review is the list of open issues and action items from the previous management review. Ideally, all these issues have been addressed and all actions have been completed and verified. If that is not the case, it is important to highlight whatever issues (e.g., resources, regulations, changes in the landscape) prevented them from being closed. Senior leaders normally don't like surprises (particularly unpleasant ones), so it might be wise to warn them of any unfinished business before the review is formally convened.

In addition to the feedback from auditors and action officers, customer feedback is an important input to the management review. Virtually every organization has customers, and they are normally the reason for the organization to exist in the first place. Their satisfaction, or lack thereof, is crucial to the organization's success. Chapter 18 mentioned real user monitoring (RUM) as one way of measuring their interactions with our information systems. Organizations are also increasingly relying on social media analysis to measure customer sentiments with regard to the organization in general and specific issues. Finally, we can use questionnaires or surveys, although these tend to have a number of challenges, including very low response rates and negative bias among respondents.

The final inputs to the management review are the recommendations for improvement based on all the other inputs. This is really the crux of the review. (While it is technically possible for a review to include no substantive change recommendations, it would be extremely unusual since it would mean that the ISMS team cannot think of any way to

improve the organizational posture.) The ISMS team presents proposed high-level changes that require the approval and/or support of the senior leaders. This is not the place to discuss low-level tactical changes; we can take care of those ourselves. Instead, we would want to ask for changes to key policies or additional resources. These recommendations must logically follow from the other inputs that have been presented to the review panel.

In setting the stage for the senior leaders' decision-making process, it is often useful to present them with a range of options. Many security professionals typically offer three to five choices, depending on the complexity of the issues. For instance, one option could be "do nothing," which describes what happens if no changes are made. At the other end of the spectrum, we could state an option that amounts to the solid-gold approach in which we pull out all the stops and make bold and perhaps costly changes that are all but guaranteed to take care of the problems. In between, we would offer one to three other choices with various levels of risk, resource requirements, and business appeal.

When we present the options, we should also present objective evaluative criteria for management to consider. A criterion that is almost always required in the presentation is the monetary cost of the change. This factor should be the life-cycle cost of the option, not just the cost of implementation. It is a common mistake to overlook the maintenance costs over the life of the system/process, disregarding the fact that these costs are often much greater than the acquisition price tag. Other factors you may want to consider presenting are risk, impact on existing systems or processes, training requirements, and complexity. But whatever evaluative factors you choose, you should apply them to each of the options in order to assess which is the best one.

Management Approval

The senior leadership considers all the inputs; typically asks some pretty pointed questions; and then decides to approve, reject, or defer the recommendations. The amount of debate or discussion at this point is typically an indicator of how effective the ISMS team was at presenting sound arguments for changes that are well nested within (and supportive of) the business processes. Obviously, the leadership's decisions are the ultimate testament to how convincing the ISMS team's arguments were.

Typically, senior management will decide to either approve the recommendation in its entirety, approve it with specific changes, reject the recommendation, or send the ISMS team back to either get more supporting data or redesign the options. Regardless of the outcome, there will likely be a list of deliverables for the next management review that will have to be addressed. It is a good idea to conclude the management review with a review of open and action items, who will address them, and when each is due. These all become inputs to the next management review in a cycle that continues indefinitely.

Chapter Review

Whereas the focus of Chapter 18 was assessing and testing technical controls, this chapter discussed administrative controls, analyzing results, and communicating them effectively. We also introduced a couple of tools that will make this effort a whole lot easier (and more effective) for you: security metrics, KPIs, and KRIs. Together with the topics

discussed in the previous chapter, we hope to have given you useful insights into how to measure and improve your ISMS, particularly when improvements depend on your ability to persuade other leaders in your organization to support your efforts. This all sets the stage for the next part of this book: “Security Operations.”

Quick Review

- A factor is an attribute of an ISMS that has a value that can change over time.
- A measurement is a quantitative observation of a factor at a particular point in time.
- A baseline is a value for a factor that provides a point of reference or denotes that some condition is met by achieving some threshold value.
- A metric is a derived value that is generated by comparing multiple measurements against each other or against a baseline.
- Good metrics are relevant, quantifiable, actionable, robust, simple, and comparative.
- An indicator is a particularly important metric that describes a key element of the effectiveness of an ISMS.
- A key performance indicator (KPI) is an indicator that is particularly significant in showing the performance of an ISMS compared to its stated goals.
- Key risk indicators (KRIs) measure the risk inherent in performing a given action or set of actions.
- Privileged user accounts pose significant risk to the organization and should be carefully managed and controlled.
- User accounts should be promptly suspended whenever the user departs the organization permanently or for an extended period.
- Data backups should not be considered reliable unless they have been verified to be usable to restore data.
- Business continuity is the term used to describe the processes enacted by an organization to ensure that its vital business processes remain unaffected or can be quickly restored following a serious incident.
- Disaster recovery focuses on restoring the information systems after a disastrous event and is a subset of business continuity.
- Security training is the process of teaching a skill or set of skills that enables people to perform specific functions better.
- Security awareness training is the process of exposing people to security issues so that they are able to recognize and respond to them better.
- Social engineering, in the context of information security, is the process of manipulating individuals so that they perform actions that violate security protocols.
- Phishing is social engineering conducted through a digital communication.

- A drive-by download is an automatic attack that is triggered simply by visiting a malicious website.
- Disaster recovery and business continuity processes both need to be evaluated regularly to ensure they remain effective in the face of environmental changes in and around the organization.
- Reports must be written with a specific audience in mind if they are to be effective.
- A management review is a formal meeting in which senior organizational leaders determine whether the information security management systems are effectively accomplishing their goals.

Questions

Please remember that these questions are formatted and asked in a certain way for a reason. Keep in mind that the CISSP exam is asking questions at a conceptual level. Questions may not always have the perfect answer, and the candidate is advised against always looking for the perfect answer. Instead, the candidate should look for the best answer in the list.

1. What is a key performance indicator (KPI)?
 - A. A value for a factor that denotes that some condition is met
 - B. The result of comparing multiple measurements
 - C. A significant indicator that shows the performance of an ISMS
 - D. A quantitative observation of a factor of an ISMS at a point in time
2. Which of the following is true about key risk indicators (KRIs)?
 - A. They tell managers where an organization stands with regard to its goals.
 - B. They are inputs to the calculation of single loss expectancy (SLE).
 - C. They tell managers where an organization stands with regard to its risk appetite.
 - D. They represent an interpretation of one or more metrics that describes the effectiveness of the ISMS.
3. All of the following are normally legitimate reasons to suspend rather than delete user accounts *except*
 - A. Regulatory compliance
 - B. Protection of the user's privacy
 - C. Investigation of a subsequently discovered event
 - D. Data retention policy
4. Data backup verification efforts should
 - A. Have the smallest scope possible
 - B. Be based on the threats to the organization
 - C. Maximize impact on business
 - D. Focus on user data

5. What is the difference between security training and security awareness training?
 - A. Security training is focused on skills, while security awareness training is focused on recognizing and responding to issues.
 - B. Security training must be performed, while security awareness training is an aspirational goal.
 - C. Security awareness training is focused on security personnel, while security training is geared toward all users.
 - D. There is no difference. These terms refer to the same process.
6. Which of the following is *not* a form of social engineering?
 - A. Pretexting
 - B. Fishing
 - C. Whaling
 - D. Blackmailing
7. When assessing the performance of your organization during a disaster recovery drill, which is the highest priority?
 - A. Safeguarding sensitive assets
 - B. Notifying the appropriate authorities
 - C. Preventing looting and vandalism
 - D. Protection of life
8. Which of the following is true about vulnerability remediation after an organizational security assessment?
 - A. All vulnerabilities uncovered must be remediated as soon as possible.
 - B. It entails applying patches to all vulnerable software systems.
 - C. Properly done, it should never impact the business.
 - D. It requires the support of everyone from the very top of the organization.
9. Which of the following is true of management reviews?
 - A. They happen periodically and include results of audits as a key input.
 - B. They happen in an ad hoc manner as the needs of the organization dictate.
 - C. They are normally conducted by mid-level managers, but their reports are presented to the key business leaders.
 - D. They are focused on assessing the management of the information systems.

Answers

1. **C.** Key performance indicators (KPIs) are indicators that are particularly significant in showing the performance of an ISMS compared to its stated goals. Because every KPI is a metric, answer B (the partial definition of a metric) would also be correct but would not be the best answer since it leaves out the significance and purpose of the metric.
2. **C.** Key risk indicators (KRIs) allow managers to understand when specific activities of the organization are moving it toward a higher level of risk. They are useful to understanding changes and managing the overall risk.
3. **B.** If the organization was intentionally attempting to protect the privacy of its user, suspension of the account would be a poor privacy measure compared to outright deletion.
4. **B.** The verification of data backups should focus on assessing the organization's ability to respond to the threats identified during the threat modeling and risk management processes. If the organization can't respond to these threats, then its backups may be useless.
5. **A.** Security training is the process of teaching a skill or set of skills that will enable people to perform specific functions better. Security awareness training, on the other hand, is the process of exposing people to security issues so that they are able to recognize and respond to them better. Security training is typically provided to security personnel, while security awareness training should be provided to every member of the organization.
6. **B.** The correct term for social engineering conducted over digital communications means is phishing, not fishing.
7. **D.** In any situation where loss or harm to human lives is a possible outcome, protection of life is the top priority. The other options are all part of a disaster recovery process, but are never the top priority.
8. **D.** Because most remediations will have some impact on the business, they require the support of everyone. This is particularly true of organizational (as opposed to system-specific) assessments because not all vulnerabilities will involve just a software patch.
9. **A.** Management reviews work best when they are regularly scheduled events involving the key organizational leaders, because this allows the subordinate leaders to plan and conduct the assessments, such as audits that provide inputs to the review.

This page intentionally left blank

PART VII

Security Operations

- **Chapter 20** Managing Security Operations
- **Chapter 21** Security Operations
- **Chapter 22** Security Incidents
- **Chapter 23** Disasters

This page intentionally left blank

Managing Security Operations

This chapter presents the following:

- Foundational security operations concepts
- Change management processes
- Configuration management
- Resource protection
- Patch and vulnerability management
- Physical security management
- Personnel safety and security

Management is keeping the trains running on time.

—Andy Dunn

Security operations is a broad field, but the image that comes to many of our minds when we hear the term is a security operations center (SOC) where analysts, threat hunters, and incident responders fight off cyberthreats day in and day out. That is, in fact, an important aspect of security operations, but it isn't the complete scope. A lot of other work goes into ensuring our spaces are protected, our systems are optimized, and our people are doing the right things. This chapter covers many of the issues that we, as security leaders, must tackle to create a secure operational environment for our organizations. Security operations is the business of managing security. It may not be as exciting as hunting down a threat actor in real time, but it is just as important.

Foundational Security Operations Concepts

Security operations revolves around people much more than around computers and networks. A good chunk of our jobs as CISSPs is to lead security teams who prevent teams of attackers from causing us harm; our computers and networks are just the battlefields

on which these groups fight each other. Sometimes, it is our own teammates who can become the enemy, either deliberately or through carelessness. So, we can't really manage security operations without first understanding the roles we need our teammates to fill and the ways in which we keep the people filling those roles honest.

Table 20-1 shows some of the common IT and security roles within organizations and their corresponding job definitions. Each role needs to have a completed and well-defined job description. Security personnel should use these job descriptions when assigning access rights and permissions in order to ensure that individuals have access only to those resources needed to carry out their tasks.

Table 20-1 contains just a few roles with a few tasks per role. Organizations should create a *complete* list of roles used within their environment, with each role's associated tasks and responsibilities. This should then be used by data owners and security personnel when determining who should have access to specific resources and the type of access. A clear and unambiguous understanding of roles and responsibilities across the organization is critical to managing security. Without it, ensuring that everyone has the right access they need for their jobs, and no more, becomes very difficult. In the sections that follow we look at other foundational concepts we all need to be able to apply to our security operations.

Organizational Role	Core Responsibilities
Cybersecurity Analyst	Monitors the organization's IT infrastructure and identifies and evaluates threats that could result in security incidents
Help Desk/Support	Resolves end-user and system technical or operations problems
Incident Responder	Investigates, analyzes, and responds to cyber incidents within the organization
IT Engineer	Performs the day-to-day operational duties on systems and applications
Network Administrator	Installs and maintains the local area network/wide area network (LAN/WAN) environment
Security Architect	Assesses security controls and recommends and implements enhancements
Security Director	Develops and enforces security policies and processes to maintain the security and safety of all organizational assets
Security Manager	Implements security policies and monitors security operations
Software Developer	Develops and maintains production software
System Administrator	Installs and maintains specific systems (e.g., database, e-mail)
Threat Hunter	Proactively finds cybersecurity threats and mitigates them before they compromise the organization

Table 20-1 Roles and Associated Tasks

SecOps

In many organizations the security and IT operations teams become misaligned because their responsibilities have different (and oftentimes conflicting) focuses. The operations staff is responsible for ensuring systems are operational, highly available, performing well, and providing users with the functionality they need. As new technology becomes available, they come under pressure by business leaders to deploy it as soon as possible to improve the organization's competitiveness. But many times this focus on operations and user functionality comes at the cost of security. Security mechanisms commonly decrease performance, delay provisioning, and reduce the functionality available to the users.

The conflicts between the priorities and incentives of the IT operations and security teams can become dysfunctional in many organizations. Many of us have witnessed the finger pointing and even outright hostility that can crop up when things go wrong. A solution that is catching on is *SecOps* (Security + Operations), which is the integration of security and IT operations people, technology, and processes to reduce risks while improving business agility. The goal is to create a culture in which security is baked into the entire life cycle of every system and process in the organization. This is accomplished by building multifunctional teams where, for instance, a cloud system administrator and a cloud security engineer work together under the leadership of a manager who is responsible for delivering agile *and* secure functionality to the organization.

Accountability

Users' access to resources must be limited and properly controlled to ensure that excessive privileges do not provide the opportunity to cause damage to an organization and its resources. Users' access attempts and activities while using a resource need to be properly monitored, audited, and logged. The individual user ID needs to be included in the audit logs to enforce individual responsibility. Each user should understand his responsibility when using organizational resources and be accountable for his actions.

Capturing and monitoring audit logs helps determine if a violation has actually occurred or if system and software reconfiguration is needed to better capture only the activities that fall outside of established boundaries. If user activities were not captured and reviewed, it would be very hard to determine if users have excessive privileges or if there has been unauthorized access.

Auditing needs to take place in a routine manner. Also, security analysts and managers need to review audit and log events. If no one routinely looks at the output, there really is no reason to create logs. Audit and function logs often contain too much cryptic or mundane information to be interpreted manually. This is why products and services are available that parse logs for organizations and report important findings. Logs should be monitored and reviewed, through either manual or automatic methods, to uncover suspicious activity and to identify an environment that is shifting away from its original

baselines. This is how administrators can be warned of many problems before they become too big and out of control.

When reviewing events, administrators need to ask certain questions that pertain to the users, their actions, and the current level of security and access:

- *Are users accessing information and performing tasks that are not necessary for their job description?* The answer indicates whether users' rights and permissions need to be reevaluated and possibly modified.
- *Are repetitive mistakes being made?* The answer indicates whether users need to have further training.
- *Do too many users have rights and privileges to sensitive or restricted data or resources?* The answer indicates whether access rights to the data and resources need to be reevaluated, whether the number of individuals accessing them needs to be reduced, and/or whether the extent of their access rights should be modified.

Need-to-Know/Least Privilege

Least privilege (one of the secure design principles introduced in Chapter 9) means an individual should have just enough permissions and rights to fulfill her role in the organization and no more. If an individual has excessive permissions and rights, it could open the door to abuse of access and put the organization at more risk than is necessary. For example, if Dusty is a technical writer for a company, he does not necessarily need to have access to the company's source code. So, the mechanisms that control Dusty's access to resources should not let him access source code.

Another way to protect resources is enforcing *need to know*, which means we must first establish that an individual has a legitimate, job role–related need for a given resource. Least privilege and need to know have a symbiotic relationship. Each user should have a need to know about the resources that she is allowed to access. If Mikela does not have a need to know how much the company paid last year in taxes, then her system rights should not include access to these files, which would be an example of exercising least privilege. The use of identity management software that combines traditional directories; access control systems; and user provisioning within servers, applications, and systems is becoming the norm within organizations. This software provides the capabilities to ensure that only specific access privileges are granted to specific users, and it often includes advanced audit functions that can be used to verify compliance with legal and regulatory directives.

Separation of Duties and Responsibilities

The objective of separation of duties (another of the secure design principles introduced in Chapter 9) is to ensure that one person acting alone cannot compromise the organization's security in any way. High-risk activities should be broken up into different parts and distributed to different individuals or departments. That way, the organization does not need to put a dangerously high level of trust in certain individuals. For fraud to take place, collusion would need to be committed, meaning more than one person would have to be

involved in the fraudulent activity. Separation of duties, therefore, is a preventive measure that requires collusion to occur for someone to commit an act that is against policy.

Separation of duties helps prevent mistakes and minimize conflicts of interest that can take place if one person is performing a task from beginning to end. For instance, a programmer should not be the only one to test her own code. Another person with a different job and agenda should perform functionality and integrity testing on the programmer's code, because the programmer may have a focused view of what the program is supposed to accomplish and thus may test only certain functions and input values, and only in certain environments.

Another example of separation of duties is the difference between the functions of a computer user and the functions of a security administrator. There must be clear-cut lines drawn between system administrator duties and computer user duties. These will vary from environment to environment and will depend on the level of security required within the environment. System and security administrators usually have the responsibility of installing and configuring software, performing backups and recovery procedures, setting permissions, adding and removing users, and developing user profiles. The computer user, on the other hand, may set or change passwords, create/edit/delete files, alter desktop configurations, and modify certain system parameters. The user should not be able to modify her own security profile, add and remove users globally, or make critical access decisions pertaining to network resources. This would breach the concept of separation of duties.

Privileged Account Management

Separation of duties also points to the need for *privileged account management* processes that formally enforce the principle of least privilege. A *privileged account* is one with elevated rights. When we hear this term, we usually think of system administrators, but it is important to consider that privileges often are gradually attached to user accounts for legitimate reasons but never reviewed again to see if they're still needed. In some cases, regular users end up racking up significant (and risky) permissions without anyone being aware of it (known as *authorization creep*).

More commonly, you will hear this concept under the label of *privileged account management (PAM)* because many organizations have very granular, role-based access controls. PAM consists of the policies and technologies used by an organization to control elevated (or privileged) access to any asset. It consists of processes for addressing the needs for individual elevated privileges, periodically reviewing those needs, reducing them to least privilege when appropriate, and documenting the whole thing.

Job Rotation

Job rotation means that, over time, more than one person fulfills the tasks of one position within the organization. This enables the organization to have more than one person who understands the tasks and responsibilities of a specific job title, which provides backup and redundancy if a person leaves the organization or is absent. Job rotation also helps identify fraudulent activities, and therefore can be considered a detective type

of control. If Keith has performed David's position, Keith knows the regular tasks and routines that must be completed to fulfill the responsibilities of that job. Thus, Keith is better able to identify whether David does something out of the ordinary and suspicious.

A related practice is *mandatory vacations*. Chapter 1 touched on reasons to make sure employees take their vacations. Reasons include being able to identify fraudulent activities and enabling job rotation to take place. If an accounting employee has been performing a "salami attack" by shaving off pennies from multiple accounts and putting the money into his own account, the employee's company would have a better chance of figuring this out if that employee is required to take a vacation for a week or longer. When the employee is on vacation, another employee has to fill in. She might uncover questionable documents and clues of previous activities, or the company may see a change in certain patterns once the employee who is committing fraud is gone for a week or two.

It is best for auditing purposes if the employee takes two contiguous weeks off from work, which allows more time for fraudulent evidence to appear. Again, the idea behind mandatory vacations is that, traditionally, those employees who have committed fraud are usually the ones who have resisted going on vacation because of their fear of being found out while away.

Service Level Agreements

As we discussed briefly in Chapter 2, a service level agreement (SLA) is a contractual agreement that states that a service provider guarantees a certain level of service. For example, a web server will be down for no more than 52 minutes per year (which is approximately a 99.99 percent availability). SLAs help service providers, whether they are an internal IT operation or an outsourcer, decide what type of availability technology is appropriate. From this determination, the price of a service or the budget of the IT operation can be set. Most frequently, organizations use SLAs with external service providers to guarantee specific performance and, if it is not delivered, to penalize (usually monetarily) the vendor.

The process of developing an internal SLA (that is, one between the IT operations team and one or more internal departments) can also be beneficial to an organization. For starters, it drives a deeper conversation between IT and whoever is requesting the service. This alone can help both sides get a clearer understanding of the opportunities and threats the service brings with it. The requestor will then better understand the tradeoffs between service levels and costs and be able to negotiate the most cost-effective service with the IT team. The IT team can then use this dialogue to justify resources such as budget or staffing. Finally, internal SLAs allow all parties to know what "right" looks like.

Whether the SLA is internal or external, the organization must collect metrics to determine whether or not it is being met. After all, if nobody measures the service, what's the point of requiring a certain level of it? Identifying these metrics, in and of itself, allows the organization to determine whether a particular requirement is important or not. If both parties are having a hard time figuring out how much scheduled downtime is acceptable, that requirement probably doesn't need to be included in the SLA.

Change Management

The Greek philosopher Heraclitus said that “the only constant in life is change,” and most of us would agree with him, especially when it comes to IT and security operations in our organizations. Change is needed to remain relevant and competitive, but it can bring risks that we must carefully manage. *Change management*, from an IT perspective, is the practice of minimizing the risks associated with the addition, modification, or removal of anything that could have an effect on IT services. This includes obvious IT actions like adding new software applications, segmenting LANs, and retiring network services. But it also includes changes to policies, procedures, staffing, and even facilities. Consequently, any change to security controls or practices probably falls under the umbrella of change management.

Change Management Practices

Well-structured change management practices are essential to minimizing the risks of changes to an environment. The process of devising these practices should include representatives for all stakeholders, so it shouldn't just be limited to IT and security staff. Most organizations that follow this process formally establish a group that is responsible for approving changes and overseeing the activities of changes that take place within the organization. This group can go by one of many names, but for this discussion we will refer to it as the change advisory board (CAB).

The CAB and change management practices should be laid out in the change management policy. Although the types of changes vary, a standard list of procedures can help keep the process under control and ensure it is carried out in a predictable manner. The following steps are examples of the types of procedures that should be part of any change management policy:

- **Request for a change to take place** The individual requesting the change must do so in writing, justify the reasons, clearly show the benefits and possible pitfalls of (that is, risk introduced by) the change. The Request for Change (RFC) is the standard document for doing this and contains all information required to approve a change.
- **Evaluate the change** The CAB reviews the RFC and analyzes its potential impacts across the entire organization. Sometimes the requester is asked to conduct more research and provide more information before the change is approved. The CAB then completes a change evaluation report and designates the individual or team responsible for planning and implementing the change.
- **Plan the change** Once the change is approved, the team responsible for implementing it gets to work planning the change. This includes figuring out all the details of how the change interfaces with other systems or processes, developing a timeline, and identifying specific actions to minimize the risks. The change must also be fully tested to uncover any unforeseen results. Regardless of how well we test, there is always a chance that the change will cause an unacceptable loss or outage, so every change request should also have a rollback plan that restores the system to the last known-good configuration.

- **Implementation** Once the change is planned and fully tested, it is implemented and integrated into any other affected processes and systems. This may include reconfiguring other systems, changing or developing policies and procedures, and providing training for affected staff. These steps should be fully documented and progress should be monitored.
- **Review the change** Once the change is implemented, it is brought back to the CAB for a final review. During this step, the CAB verifies that the change was implemented as planned, that any unanticipated consequences have been properly addressed, and that the risks remain within tolerable parameters.
- **Close or sustain** Once the change is implemented and reviewed, it should be entered into a change log. A full report summarizing the change may also be submitted to management, particularly for changes with large effects across the organization.

These steps, of course, usually apply to large changes that take place within an organization. These types of changes are typically expensive and can have lasting effects on an organization. However, smaller changes should also go through some type of change control process. If a server needs to have a patch applied, it is not good practice to have an engineer just apply it without properly testing it on a nonproduction server, without having the approval of the IT department manager or network administrator, and without having backup and backout plans in place in case the patch causes some negative effect on the production server. Of course, these changes still need to be documented. For this reason, ITIL 4 (introduced in Chapter 4) specifies three types of changes that follow the same basic process but tailored for specific situations:

- **Standard changes** Preauthorized, low-risk changes that follow a well-known procedure. Examples include patching a server or adding memory or storage to it.
- **Emergency changes** Changes that must be implemented immediately. Examples include implementing a security patch for a zero-day exploit or isolating the network from a DDoS attack.
- **Normal changes** All other changes that are not standard changes or emergency changes. Examples include adding a server that will provide new functionality or introducing a new application to (or removing a legacy one from) the golden image.

Regardless of the type of change, it is critical that the operations department create approved backout plans before implementing changes to systems or the network. It is very common for changes to cause problems that were not properly identified before the implementation process began. Many network engineers have experienced the headaches of applying poorly developed “fixes” or patches that end up breaking something else in the system. Developing a backout plan ensures productivity is not negatively affected by these issues. This plan describes how the team will restore the system to its original state before the change was implemented.

Change Management Documentation

Failing to document changes to systems and networks is only asking for trouble, because no one will remember, for example, what was done to that one server in the demilitarized zone (DMZ) six months ago or how the main router was fixed when it was acting up last year. Changes to software configurations and network devices take place pretty often in most environments, and keeping all of these details properly organized is impossible, unless someone maintains a log of this type of activity.

Numerous changes can take place in an organization, some of which are as follows:

- New computers installed
- New applications installed
- Different configurations implemented
- Patches and updates installed
- New technologies integrated
- Policies, procedures, and standards updated
- New regulations and requirements implemented
- Network or system problems identified and fixes implemented
- Different network configurations implemented
- New networking devices integrated into the network
- Company acquired by, or merged with, another company

The list could go on and on and could be general or detailed. Many organizations have experienced some major problem that affects the network and employee productivity. The IT department may run around trying to figure out the issue and go through hours or days of trial-and-error exercises to find and apply the necessary fix. If no one properly documents the incident and what was done to fix the issue, the organization may be doomed to repeat the same scramble six months to a year down the road.

Configuration Management

At every point in the O&M part of assets' life cycles (which we discussed in Chapter 5), we need to also ensure that we get (and keep) a handle on how these assets are configured. Sadly, most default configurations are woefully insecure. This means that if we do not configure security when we provision new hardware or software, we are virtually guaranteeing successful attacks on our systems. *Configuration management (CM)* is the process of establishing and maintaining consistent configurations on all our systems to meet organizational requirements.

Configuration management processes vary among organizations but have certain elements in common. Virtually everyone that practices it starts off by defining and establishing organization-wide agreement on the required configurations for all systems in the scope of the effort. At a minimum, this should include the users' workstations

and all business-critical systems. These configurations are then applied to all systems. There will be exceptions, of course, and special requirements that lead to nonstandard configurations, which need to be approved by the appropriate individuals and documented. There will also be changes over time, which should be dealt with through the change management practices defined in the previous section. Finally, configurations need to be periodically audited to ensure continued compliance with them.

Baselining

A *baseline* is the configuration of a system at a point in time as agreed upon by the appropriate decision makers. For a typical user workstation, a baseline defines the software that is installed (both operating system and applications), policies that are applied (e.g., disabling USB thumb drives), and any other configuration setting such as the domain name, DNS server address, and many others. Baselining allows us to build a system once, put it through a battery of tests to ensure it works as expected, and then provision it out consistently across the organization.

In a perfect world, all systems that provide the same functionality are configured identically. This makes it easier to manage them throughout their life cycles. As we all know, however, there are plenty of exceptions in the real world. System configuration exceptions often have perfectly legitimate business reasons, so we can't just say "no" to exception requests and keep our lives simple. The system baseline allows us to narrow down what makes these exceptional systems different. Rather than document every single configuration parameter again (which could introduce errors and omissions), all we have to do is document what is different from a given baseline.

Baselines do more than simply tell us what systems (should) look like at a given point in time; they also document earlier configuration states for those systems. We want to keep old baselines around because they tell the story of how a system evolved. Properly annotated, baselines tell us not only the "what" but also the "why" of configurations over time.

A related concept to baselining is the golden image, which is a preconfigured, standard template from which all user workstations are provisioned. A golden image is known by many other names including gold master, clone image, master image, and base image. Whatever name you use, it saves time when provisioning systems because all you have to do is clone the image onto a device, enter a handful of parameters unique to the system (such as the hostname), and it's ready for use. Golden images also improve security by consistently applying security controls to every cloned system. Another advantage is a reduction in configuration errors, which also means a lower risk of inadvertently introduced vulnerabilities.

Provisioning

We already addressed secure provisioning in Chapter 5 but the topic bears revisiting in the context of configuration management. Recall that provisioning is the set of all activities required to provide one or more new information services to a user or group of users ("new" meaning previously not available to that user or group). Technically, provisioning and

Configuration Management vs. Change Management

Change management is a *business* process aimed at deliberately regulating the changing nature of business activities such as projects or IT services. It is concerned with issues such as changing the features in a system being developed or changing the manner in which remote workers connect to the internal network. While IT and security personnel are involved in change management, they are usually not in charge of it.

Configuration management is an *operational* process aimed at ensuring that controls are configured correctly and are responsive to the current threat and operational environments. As an information security professional, you would likely lead in configuration management but simply participate in change management processes.

configuration are two different but related activities. Provisioning generally entails acquiring, installing, and launching a new service. Depending on how this is done, that service may still need to be configured (and possibly even baselined).

Automation

As you can imagine, configuration management requires tracking and updating a lot of information on many different systems. This is why mature organizations leverage automation for many of the required tasks, including maintaining individual configuration items in a *configuration management database (CMDB)*. The CMDB can store information about all organizational assets, their baselines, and their relationships to one another. Importantly, a CMDB provides versioning so that, if a configuration error is made, reverting to a previous baseline is easy.

More elaborate automation tools are capable of not only tracking configurations but also provisioning systems that implement them. Perhaps the best-known tool in this regard, particularly for virtualized or cloud infrastructures, is Ansible, which is an open-source configuration management, deployment, and orchestration tool. Through the use of playbooks written in YAML (which, recursively, stands for “YAML Ain’t Markup Language”), Ansible allows automated asset provisioning and configuration.

Resource Protection

In Chapter 5, we defined assets as anything of worth to the organization. A related concept is a *resource*, which is anything that is required to perform an activity or accomplish a goal. So, a resource can also be an asset if you own it and it has inherent value to you. In the context of security operations, a resource is anything the organization needs to accomplish any of its tasks. This includes hardware, software, data, and the media on which the last two are stored.



EXAM TIP Though assets and resources are, technically, slightly different things, you should treat them as synonymous in the exam.

We will discuss how to protect hardware resources later in this chapter when we cover physical security. Though we already covered software, data, and media protections in Chapter 6, the topic is worth revisiting as it applies to managing security operations. There are three types of digital resources that are of particular interest in this regard: system images, source files, and backups.

System Images

Because system images are essential to efficiently provisioning systems, they are a key resource both during normal operations and when we are responding to a security incident. Presumably, the images we use to clone new (or replacement) systems are secure because (as a best practice) we put a lot of work into hardening them and ensuring they contain no known vulnerabilities. However, if adversaries were able to modify the images so as to introduce vulnerabilities, they would have free access to any system provisioned using the tainted images. Similarly, if the images were destroyed (deliberately, accidentally, or through an act of nature), recovering from a large-scale incident would be much more difficult and time-consuming.

Source Files

If the images were unavailable or otherwise compromised, we would have to rebuild everything from scratch. There are also cases in which we just need to install specific software. Either way, we need reliable source files. Source files contain the code that executes on a computer to provide applications or services. This code can exist in either executable form or as a sequence of statements in a high-level language such as C/C++, Java, or Python. Either way, it is possible for adversaries to insert malicious code into source files so that any system provisioned using them will be vulnerable. Worse yet, if you work for a software company with clients around the world, your company may be a much more interesting target for advanced persistent threats (APTs) who may want to compromise your software to breach your customers. This kind of software supply-chain attack is best exemplified by the SolarWinds attack of 2020.

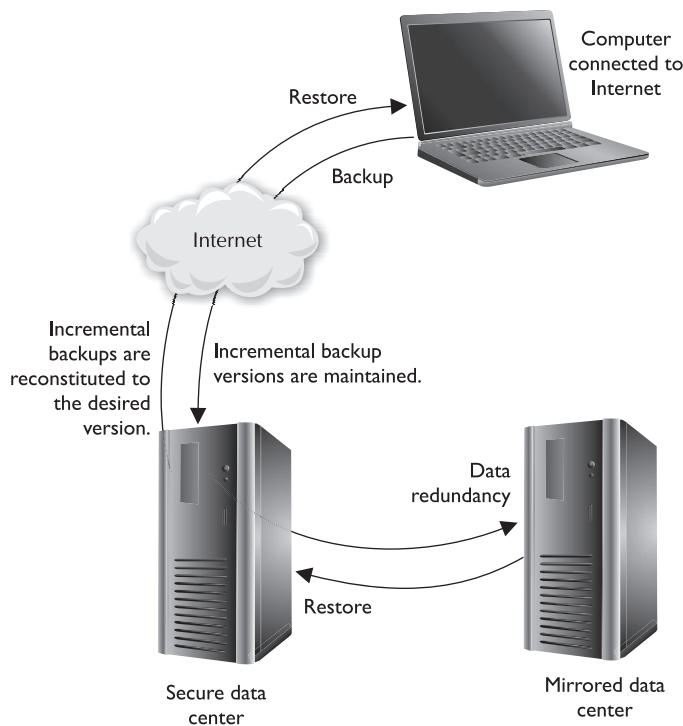
Even if your organization is not likely to be targeted by APTs, you are probably concerned about ransomware attacks. Having good backups is the key to quickly recovering from ransomware (without having to pay the ransom), but it hinges on the integrity and availability of the backup data. Many cybercriminals deliberately look for backups and encrypt them also to force their victims to pay the ransom.

Backups

Backing up software and having backup hardware devices are two large parts of network availability. You need to be able to restore data if a hard drive fails, a disaster takes place, or some type of software corruption occurs.

Every organization should develop a policy that indicates what gets backed up, how often it gets backed up, and how these processes should occur. If users have important information on their workstations, the operations department needs to develop a method that indicates that backups include certain directories on users' workstations or that users move their critical data to a server share at the end of each day to ensure it gets backed up. Backups may occur once or twice a week, every day, or every three hours. It is up to the organization to determine this interval. The more frequent the backups, the more resources will be dedicated to it, so there needs to be a balance between backup costs and the actual risk of potentially losing data.

An organization may find that conducting automatic backups through specialized software is more economical and effective than spending IT work-hours on the task. The integrity of these backups needs to be checked to ensure they are happening as expected—rather than finding out right after two major servers blow up that the automatic backups were saving only temporary files.



Protecting Backups from Ransomware

The best way to minimize your risks due to ransomware is to have effective backups that are beyond the reach of the cybercriminals and can quickly restore affected systems. This means putting the greatest distance (and security controls) possible between a system

and its backups. Obviously, you should never store backups on the system itself or on a directly connected external drive. The following are some tips on how to keep your backups away from threat actors:

- *Use a different OS for your backup server.* Most ransomware today targets a single type of OS (mostly Windows). Even if the attack is not automated, threat actors are likelier to be proficient in whatever OS they are attacking, so having your backups managed by a system running a different OS automatically gives you a leg up.
- *Get your backups out of town.* Whatever you do, make sure your backups are not on a drive that is directly attached to the asset you are protecting, or even on the same LAN segment (like in the same data center). The more distance, the better, especially if you can layer controls like ACLs or even use data diodes. We know of data so sensitive that its backups are physically transported to other states or countries periodically.
- *Go old school.* Consider using older technologies like optical discs and magnetic tapes. You may get some weird looks from your early-adopter colleagues, but you may save the day when things go sideways on you.
- *Protect your backups like your career depends on it.* (It may!) Stay up to date on the latest techniques cybercriminals are using to attack backups and ensure you have adequate controls in place to prevent them from being effective.

Hierarchical Storage Management

Hierarchical storage management (HSM) provides continuous online backup functionality. It combines hard disk technology with the cheaper and slower optical or tape jukeboxes. The HSM system dynamically manages the storage and recovery of files, which are copied to storage media devices that vary in speed and cost. The faster media holds the files that are accessed more often, and the seldom-used files are stored on the slower devices, or *near-line* devices, as shown in Figure 20-1. The storage media could include optical discs, magnetic disks, and tapes. This functionality happens in the background without the knowledge of the user or any need for user intervention.

HSM works, according to tuning based on the trade-off between the cost of storage and the availability of information, by migrating the actual content of less used files to lower-speed, lower-cost storage, while leaving behind a “stub,” which looks to the user like it contains the full data of the migrated file. When the user or an application accesses the stub, the HSM uses the information in the stub to find the real location of the information and then retrieve it transparently for the user.

This type of technology was created to save money and time. If all data was stored on hard drives, that would be expensive. If a lot of the data was stored on tapes, it would take too long to retrieve the data when needed. So HSM provides a terrific approach by providing you with the data you need, when you need it, without having to bother the administrator to track down some tape or optical disc.

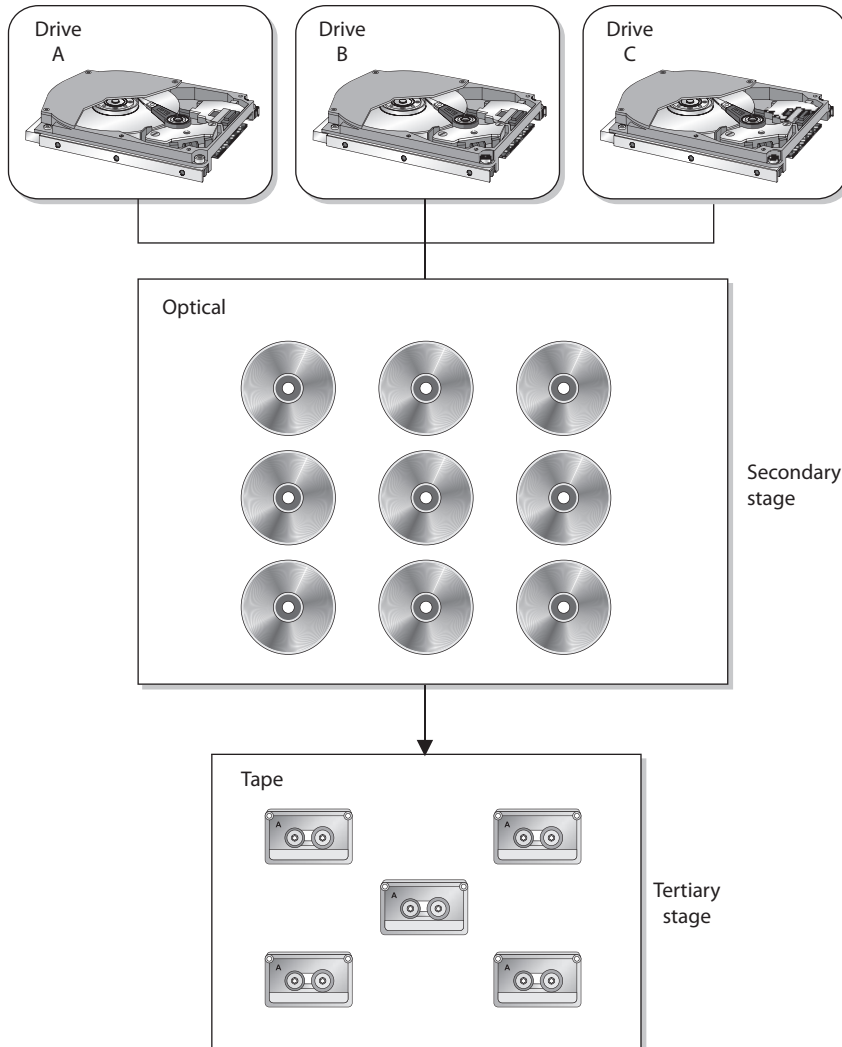


Figure 20-1 HSM provides an economical and efficient way of storing data.

Backups should include the underlying operating system and applications, as well as the configuration files for both. Systems are attached to networks, and network devices can experience failures and data losses as well. Data loss of a network device usually means the configuration of the network device is lost completely (and the device will not even boot up), or that the configuration of the network device reverts to defaults (which, though it will boot up, does your network little good). Therefore, the configurations of network and other nonsystem devices (for example, the phone system) in the environment are also necessary.

Vulnerability and Patch Management

Dealing with new vulnerabilities and their corresponding patches is an inevitability in cybersecurity. The trick is to deal with these in an informed and deliberate manner. While the following sections treat vulnerability management and patch management separately, it is important to consider them as two pieces of the same puzzle in real life. We may learn of a new vulnerability for which a patch does not yet exist. Equally bad would be applying a patch that brings down a critical business system. For these reasons (among many others), we should manage vulnerabilities and patches in a synchronized and coordinated manner across our organizations.

Vulnerability Management

No sufficiently complex information system can ever be completely free of vulnerabilities. *Vulnerability management* is the cyclical process of identifying vulnerabilities, determining the risks they pose to the organization, and applying security controls that bring those risks to acceptable levels. Many people equate vulnerability management with periodically running a vulnerability scanner against their systems, but the process must include more than just that. Vulnerabilities exist not only in software, which is what the scanners assess, but also in business processes and in people. Flawed business processes, such as sharing proprietary information with parties who have not signed a nondisclosure agreement (NDA), cannot be detected by vulnerability scanners. Nor can they detect users who click malicious links in e-mails. What matters most is not the tool or how often it is run, but having a formal process that looks at the organization holistically and is closely tied to the risk management process.

Vulnerability management is part of our risk management process. We identify the things that we have that are of value to us and the threat actors that might take those away from us or somehow interfere with our ability to benefit from them. Then we figure out how these actors might go about causing us losses (in other words, exploiting our vulnerabilities) and how likely these events might be. As we discussed in Chapter 2, this gives us a good idea of our risk exposure. The next step is to decide which of those risks we will address and how. The “how” is typically through the application of a security control. Recall that we can never bring our risk to zero, which means we will always have vulnerabilities for which we have no effective controls. These unmitigated risks exist because we think the chance of them being realized or their impact on the organization (or both) is low enough for the risk to be tolerable. In other words, the cost of mitigating the risk is not worth the return on our investment. For those risks, the best we can do is continually monitor for changes in their likelihood or potential impact.

As you can see, vulnerability management is all about finding vulnerabilities, understanding their impact on the organization, and determining what to do about them. Since information system vulnerabilities can exist in software, processes, or people, it is worthwhile to discuss how we implement and support vulnerability management in each of these areas.

Software Vulnerabilities

Vulnerabilities are usually discovered by security researchers who notify vendors and give them some time (at least two weeks) to work on a patch before the researchers make their findings public. This is known as responsible or ethical disclosure. The Computer Emergency Response Team Coordination Center (CERT/CC) is the main clearinghouse for vulnerability disclosures. Once a vulnerability is discovered, vulnerability scanner vendors release plug-ins for their tools. These plug-ins are essentially simple programs that look for the presence of one specific flaw.



NOTE Some organizations have their own in-house vulnerability research capability or can write their own plug-ins. In our discussion, we assume the more general case in which vulnerability scanning is done using third-party commercial tools whose licenses include subscriptions to vulnerability feeds and related plug-ins.

As previously mentioned, software vulnerability scanning is what most people think of when they hear the term vulnerability management. Scanning is simply a common type of vulnerability assessment that can be divided into four phases:

- 1. Prepare** First, you have to determine the scope of the vulnerability assessment. What are you testing and how? Having defined the scope, you schedule the event and coordinate it with affected asset and process owners to ensure it won't interfere with critical business processes. You also want to ensure you have the latest vulnerability signatures or plug-ins for the systems you will be testing.
- 2. Scan** For best results, the scan is automated, follows a script, and happens outside of the regular hours of operation for the organization. This reduces the chance that something goes unexpectedly wrong or that you overlook a system. During the scan, it is helpful to monitor resource utilization (like CPU and bandwidth) to ensure you are not unduly interfering with business operations.
- 3. Remediate** In a perfect world, you don't find any of the vulnerabilities for which you were testing. Typically, however, you find a system that somehow slipped through the cracks, so you patch it and rescan just to be sure. Sometimes, however, there are legitimate business reasons why a system can't be patched (at least right away), so remediation may require deploying a compensating control or (in the worst case) accepting the risk as is.
- 4. Document** This important phase is often overlooked because some organizations rely on the reports that are automatically generated by the scanning tools. These reports, however, don't normally include important details like why a vulnerability may intentionally be left unpatched, the presence of compensating controls elsewhere, or the need for more/less frequent scanning of specific systems. Proper documentation ensures that assumptions, facts, and decisions are preserved to inform future decisions.

Process Vulnerabilities

A process vulnerability exists whenever there is a flaw or weakness in a business process, independent of the use of automation. For example, suppose a user account provisioning process requires only an e-mail from a supervisor asking for an account for the new hire. Since e-mail messages can be spoofed, a threat actor could send a fake e-mail impersonating a real supervisor. If the system administrator creates the account and responds with the new credentials, the adversary would now have a legitimate account with whatever authorizations were requested.

Process vulnerabilities frequently are overlooked, particularly when they exist at the intersection of multiple departments within the organization. In the example, the account provisioning process vulnerability exists at the intersection of a business area (where the fictitious user will supposedly work), IT, and human resources.

A good way to find process vulnerabilities is to periodically review existing processes using a red team. As introduced in Chapter 18, a red team is a group of trusted individuals whose job is to look at something from an adversary's perspective. Red teaming is useful in many contexts, including identifying process vulnerabilities. The red team's task in this context would be to study the processes, understand the organization's environment, and then look for ways to violate its security policies. Ideally, red team exercises should be conducted whenever any new process is put in place. Realistically, however, these events take place much less frequently (if at all).



NOTE The term *red team exercise* is often used synonymously with *penetration test*. In reality, a red team exercise can apply to any aspect of an organization (people, processes, facilities, products, ideas, information systems) and aims to emulate the actions of threat actors seeking specific objectives. A penetration test, on the other hand, is focused on testing the effectiveness of security controls in facilities and/or information systems.

Human Vulnerabilities

By many accounts, over 90 percent of security incidents can be traced back to a member of an organization doing something they shouldn't have, maliciously or otherwise. This implies that if your vulnerability management is focused exclusively on hardware and software systems, you may not be reducing your attack surface by much. A common approach to managing human vulnerabilities is social engineering assessments. We briefly introduced social engineering in Chapter 18 as a type of attack but return to it now as a tool in your vulnerability management toolkit.

Chris Hadnagy, one of the world's leading experts on the subject, defines social engineering as "the act of manipulating a person to take an action that *may* or *may not* be in the 'target's' best interest." A social engineering assessment involves a team of trained personnel attempting to exploit vulnerabilities in an organization's staff. This could result in targets revealing sensitive information, allowing the social engineers into restricted areas, clicking malicious links, or plugging into their computer a thumb drive laden with malware.

A social engineering assessment, much like its nefarious counterpart, consists of three phases:

- 1. Open-source intelligence (OSINT) collection** Before manipulating a target, the social engineer needs to learn as much as possible about that person. This phase is characterized by searches for personal information in social media sites; web searches; and observation, eavesdropping, and casual conversations. Some OSINT tools allow quick searches of a large number of sources for information on specific individuals or organizations.
- 2. Assessment planning** The social engineer could go on gathering OSINT forever but at some point (typically very quickly) will have enough information to formulate a plot to exploit one or more targets. Some people respond emotionally to certain topics, while others may best be targeted by impersonating someone in a position of authority. The social engineer identifies the kinds of engagements, topics, and pretexts that are likeliest to work against one or more targets.
- 3. Assessment execution** Regardless of how well planned an assessment may be, we know that no plan survives first contact. Social engineers have to think quickly on their feet and be very perceptive of their targets' states of mind and emotions. In this phase, they engage targets through some combination of personal face-to-face, telephonic, text, or e-mail exchange and persuade them to take some action that compromises the security of the organization.

Rarely is a social engineering assessment not effective. At the end of the event, the assessors report their findings and use them to educate the organization on how to avoid falling for these tricks. Perhaps the most common type of assessment is in the form of phishing, but a real human vulnerability assessment should be much more comprehensive.

Patch Management

According to NIST Special Publication 800-40, Revision 3, *Guide to Enterprise Patch Management Technologies*, patch management is “the process for identifying, acquiring, installing, and verifying patches for products and systems.” *Patches* are software updates intended to remove a vulnerability or defect in the software, or to provide new features or functionality for it. Patch management is, at least in a basic way, an established part of organizations' IT or security operations already.

Unmanaged Patching

One approach to patch management is to use a decentralized or unmanaged model in which each software package on each device periodically checks for updates and, if any are available, automatically applies them. While this approach may seem like a simple

solution to the problem, it does have significant issues that could render it unacceptably risky for an organization. Among these risks are the following:

- **Credentials** Installing patches typically requires users to have admin credentials, which violates the principle of least privilege.
- **Configuration management** It may be difficult (or impossible) to attest to the status of every application in the organization, which makes configuration management much more difficult.
- **Bandwidth utilization** Having each application or service independently download the patches will lead to network congestion, particularly if there is no way to control when this will happen.
- **Service availability** Servers are almost never configured to automatically update themselves because this could lead to unscheduled outages that have a negative effect on the organization.

There is almost no advantage to decentralized patch management, except that it is better than doing nothing. The effort saved by not having management overhead is more than balanced by the additional effort you'll have to put into responding to incidents and solving configuration and interoperability problems. Still, there may be situations in which it is not possible to actively manage some devices. For instance, if your users are allowed to work from home using personal devices, then it would be difficult to implement the centralized approach we discuss next. In such situations, the decentralized model may be the best to take, provided you also have a way to periodically (say, each time users connect back to the mother ship) check the status of their updates.

Centralized Patch Management

Centralized patch management is considered a best practice for security operations. There are multiple approaches to implementing it, however, so you must carefully consider the pluses and minuses of each. The most common approaches are

- **Agent based** An update agent is installed on each device. This agent communicates with one or more update servers and compares available patches with software and versions on the local host, updating as needed.
- **Agentless** One or more hosts remotely connect to each device on the network using admin credentials and check the remote device for needed updates. A spin on this is the use of Active Directory objects in a domain controller to manage patch levels.
- **Passive** Depending on the fidelity that an organization requires, it may be possible to passively monitor network traffic to infer the patch levels on each networked application or service. While minimally intrusive to the end devices, this approach is also the least effective since it may not always be possible to uniquely identify software versions through their network traffic artifacts.

Regardless of the approach you take, you want to apply the patches as quickly as possible. After all, every day you delay is an extra day that your adversaries have to exploit

your vulnerabilities. The truth is that you can't (or at least shouldn't) always roll out the patch as soon as it comes out. There is no shortage of reports of major outages caused by rolling out patches without first testing their effects. Sometimes the fault lies with the vendor, who, perhaps in its haste to remove a vulnerability, failed to properly test that the patch wouldn't break any other functionality of the product. Other times the patch may be rock solid and yet have a detrimental second- or third-order effect on other systems on your hosts or networks. This is why testing the patch before rolling it out is a good idea.

Virtualization technologies make it easier to set up a patch test lab. At a minimum, you want to replicate your critical infrastructure (e.g., domain controller and production servers) in this virtual test environment. Most organizations also create at least one virtual machine (VM) that mimics each deployed operating system, with representative services and applications.



NOTE It is often possible to mitigate the risk created by a software vulnerability using other controls, such as rules for your firewalls, intrusion detection system (IDS), or intrusion protection system (IPS). This can buy time for you to test the patches. It also acts as a compensatory control.

Whether or not you are able to test the patches before pushing them out (and you really should), it is also a good idea to patch your subnets incrementally. It may take longer to get to all systems, but if something goes wrong, it will only affect a subset of

Reverse Engineering Patches

Zero-day exploits are able to successfully attack vulnerabilities that are not known to the software vendor or users of its software. For that reason, zero-day exploits are able to bypass the vast majority of controls such as firewalls, antimalware, and IDS/IPS. Though zero-day exploits are very powerful, they are also exceptionally hard to develop and very expensive to buy in the underground markets.

There is an easier and cheaper way for attackers to exploit recent vulnerabilities, and that is by reverse engineering the software patches that vendors push out. This approach takes advantage of the delay between a patch being available and it getting pushed to all the vulnerable computers in the organization. If the attacker can reverse engineer the patch faster than the defenders use it to update all computers, then the attacker wins. Some vendors are mitigating this threat by using *code obfuscation*, which, in an ironic turn of events, is a technique developed by attackers almost 30 years ago in an effort to thwart the then simple pattern-matching approach of antimalware solutions.

Even with code obfuscation, it is just a matter of time before the bad guys figure out what the vulnerability is. This puts pressure on the defenders to roll out the patches across the entire organization as quickly as possible. In this haste, organizations sometimes overlook problem indicators. Add to this a healthy application of Murphy's law and you see why it is imperative to have a way to deal with these unknowns. A *rollback plan* (previously discussed in the "Change Management" section of this chapter) describes the steps by which a change is reversed in order to restore functionality or integrity.

your users and services. This gradual approach to patching also serves to reduce network congestion that could result from all systems attempting to download patches at the same time. Obviously, the benefits of gradual patching need to be weighed against the additional exposure that the inherent delays will cause.

Physical Security

We already discussed physical security in Chapter 10, but our focus then was on the design of sites and facilities. The CISSP CBK breaks physical security into design, which falls under Domain 3 (Security Architecture and Engineering), and operations, which falls in the current Domain 7 (Security Operations). We follow the same approach here.

As with any other defensive technique, physical security should be implemented using the defense-in-depth secure design principle. For example, before an intruder can get to the written recipe for your company's secret barbeque sauce, she will need to climb or cut a fence, slip by a security guard, pick a door lock, circumvent a biometric access control reader that protects access to an internal room, and then break into the safe that holds the recipe. The idea is that if an attacker breaks through one control layer, there will be others in her way before she can obtain the company's crown jewels.



NOTE It is also important to have a diversity of controls. For example, if one key works on four different door locks, the intruder has to obtain only one key. Each entry should have its own individual key or authentication combination.

This defense model should work in two main modes: one mode during normal facility operations and another mode during the time the facility is closed. When the facility is closed, all doors should be locked with monitoring mechanisms in strategic positions to alert security personnel of suspicious activity. When the facility is in operation, security gets more complicated because authorized individuals need to be distinguished from unauthorized individuals. Perimeter security controls deal with facility and personnel access controls and with external boundary protection mechanisms. Internal security controls deal with work area separation and personnel badging. Both perimeter and internal security also address intrusion detection and corrective actions. The following sections describe the elements that make up these categories.

External Perimeter Security Controls

Your first layer of defense is your external perimeter. This could be broken down into distinct, concentric areas of increasing security. Let's consider an example taken from the *Site Security Design Guide*, published by the U.S. General Services Administration (GSA) Public Buildings Service, which is shown in Figure 20-2. In it, we see the entire site is fenced off, which actually creates two security zones: the (external) neighborhood (zone 1) and the standoff perimeter (zone 2). Depending on risk levels, the organization may want to restrict site access and parking by creating a third zone. Even if the risk is fairly low, it may be desirable to ensure that vehicles are unable to get too close to the building.

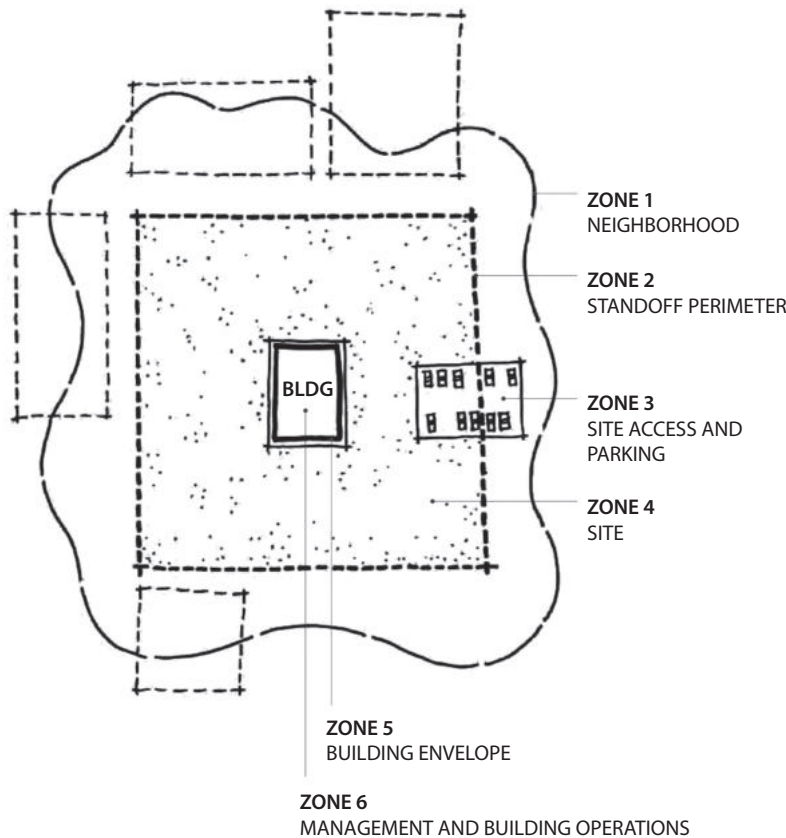


Figure 20-2 Security zones around a facility (Source: https://www.wbdg.org/FFC/GSA/site_security_dg.pdf)

This protects the facility against accidents, but also against explosions. (A good rule of thumb is to ensure there is a 200-foot standoff distance between any vehicles and buildings.) Then there is the rest of the enclosed site (zone 4), which could include break areas for employees, backup power plants, and anything else around the building exterior. Finally, there's the inside of the building, which we'll discuss later in this chapter. Each of these zones has its own set of requirements, which should be increasingly restrictive the closer someone gets to the building.

External perimeter security controls are usually put into place to provide one or more of the following services:

- Control pedestrian and vehicle traffic flows
- Provide various levels of protection for different security zones
- Establish buffers and delaying mechanisms to protect against forced entry attempts
- Limit and control entry points

These services can be provided by using the following control types (which are not all-inclusive):

- **Access control mechanisms** Locks and keys, an electronic card access system, personnel awareness
- **Physical barriers** Fences, gates, walls, doors, windows, protected vents, vehicular barriers
- **Intrusion detection** Perimeter sensors, interior sensors, annunciation mechanisms
- **Assessment** Guards, surveillance cameras
- **Response** Guards, local law enforcement agencies
- **Deterrents** Signs, lighting, environmental design

Several types of perimeter protection mechanisms and controls can be put into place to protect an organization's facility, assets, and personnel. They can deter would-be intruders, detect intruders and unusual activities, and provide ways of dealing with these issues when they arise. Perimeter security controls can be natural (hills, rivers) or manmade (fencing, lighting, gates). Landscaping is a mix of the two. In Chapter 10, we explored Crime Prevention Through Environmental Design (CPTED) and how this approach is used to reduce the likelihood of crime. Landscaping is a tool employed in the CPTED method. Sidewalks, bushes, and created paths can point people to the correct entry points, and trees and spiky bushes can be used as natural barriers. These bushes and trees should be placed such that they cannot be used as ladders or accessories to gain unauthorized access to unapproved entry points. Also, there should not be an overwhelming number of trees and bushes, which could provide intruders with places to hide. In the following sections, we look at the manmade components that can work within the landscaping design.

Fencing

Fencing can be quite an effective physical barrier. Although the presence of a fence may only delay dedicated intruders in their access attempts, it can work as a psychological deterrent by telling the world that your organization is serious about protecting itself.

Fencing can provide crowd control and helps control access to entrances and facilities. However, fencing can be costly and unsightly. Many organizations plant bushes or trees in front of the fencing that surrounds their buildings for aesthetics and to make the building less noticeable. But this type of vegetation can damage the fencing over time or negatively affect its integrity. The fencing needs to be properly maintained, because if a company has a sagging, rusted, pathetic fence, it is equivalent to telling the world that the company is not truly serious and disciplined about protection. But a nice, shiny, intimidating fence can send a different message—especially if the fencing is topped with three rungs of barbed wire.

When deciding upon the type of fencing, several factors should be considered. For example, when using metal fencing, the gauge of the metal should correlate to the types of physical threats the organization most likely faces. After carrying out the risk analysis (covered in Chapter 2), the physical security team should understand the probability of

enemies attempting to cut the fencing, drive through it, or climb over or crawl under it. Understanding these threats will help the team determine the requirements for security fencing.

The risk analysis results will also help indicate what height of fencing the organization should implement. Fences come in varying heights, and each height provides a different level of security:

- Fences *three to four feet high* only deter casual trespassers.
- Fences *six to seven feet high* are considered too high to climb easily.
- Fences *eight feet high* (possibly with strands of barbed or razor wire at the top) deter the more determined intruder and clearly demonstrate your organization is serious about protecting its property.

The barbed wire on top of fences can be tilted in or out, which also provides extra protection. A prison would have the barbed wire on top of the fencing pointed in, which makes it harder for prisoners to climb and escape. Most organizations would want the barbed wire tilted out, making it harder for someone to climb over the fence and gain access to the premises.

Critical areas should have fences at least eight feet high to provide the proper level of protection. The fencing must be taut (not sagging in any areas) and securely connected to the posts. The fencing should not be easily circumvented by pulling up its posts.

Fencing: Gauges, Mesh Sizes, and Security

The gauge of fence wiring is the thickness of the wires used within the fence mesh. The lower the gauge number, the larger the wire diameter:

- **11 gauge** = 0.0907-inch diameter
- **9 gauge** = 0.1144-inch diameter
- **6 gauge** = 0.162-inch diameter

The mesh sizing is the minimum clear distance between the wires. Common mesh sizes are 2 inches, 1 inch, and 3/8 inch. It is more difficult to climb or cut fencing with smaller mesh sizes, and the heavier-gauged wiring is harder to cut. The following list indicates the strength levels of the most common gauge and mesh sizes used in chain-link fencing today:

- **Extremely high security** 3/8-inch mesh, 11 gauge
- **Very high security** 1-inch mesh, 9 gauge
- **High security** 1-inch mesh, 11 gauge
- **Greater security** 2-inch mesh, 6 gauge
- **Normal industrial security** 2-inch mesh, 9 gauge