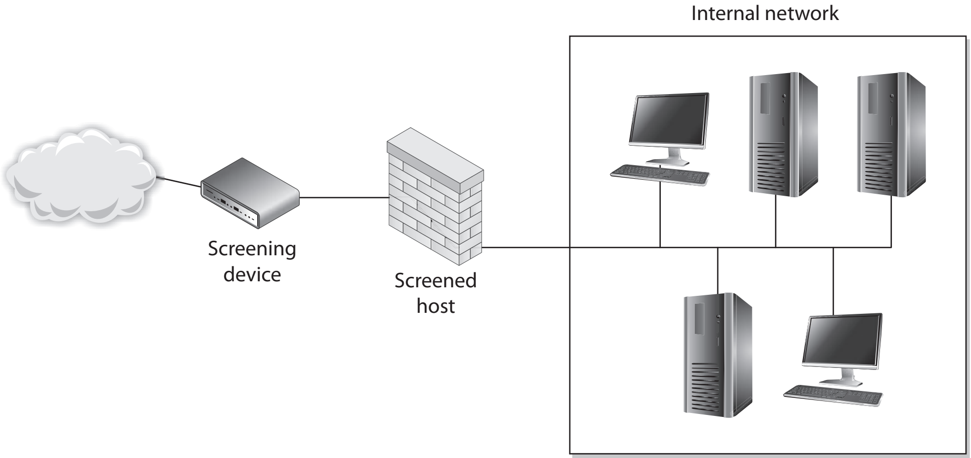


Figure 21-8 A screened host is a firewall that is screened by a router.

more rules to the traffic and drops the denied packets. Then the traffic moves to the internal destination hosts. The screened host (the firewall) is the only device that receives traffic directly from the router. No traffic goes directly from the Internet, through the router, and to the internal network. The screened host is always part of this equation.

If the firewall is an application-based system, protection is provided at the network layer by the router through packet filtering, and at the application layer by the firewall. This arrangement offers a high degree of security, because for an attacker to be successful, she would have to compromise two systems.

What does the word "screening" mean in this context? As shown in Figure 21-8, the router is a screening device and the firewall is the screened host. This just means there is a layer that scans the traffic and gets rid of a lot of the "junk" before the traffic is directed toward the firewall. A screened host is different from a screened subnet, which is described next.

Screened Subnet A *screened-subnet* architecture adds another layer of security to the screened-host architecture. The external firewall screens the traffic entering the DMZ network. However, instead of the firewall then redirecting the traffic to the internal network, an interior firewall also filters the traffic. The use of these two physical firewalls creates a DMZ.

In an environment with only a screened host, if an attacker successfully breaks through the firewall, nothing lies in her way to prevent her from having full access to the internal network. In an environment using a screened subnet, the attacker would have to hack through another firewall to gain access. In this layered approach to security, the more layers provided, the better the protection. Figure 21-9 shows a simple example of a screened subnet.

The examples shown in the figures are simple in nature. Often, more complex networks and DMZs are implemented in real-world systems. Figures 21-10 and 21-11 show some other possible architectures of screened subnets and their configurations.

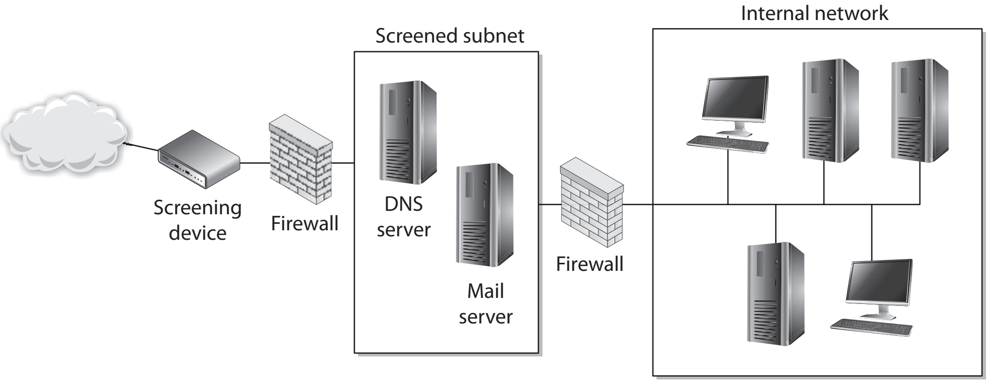


Figure 21-9 With a screened subnet, two firewalls are used to create a DMZ.

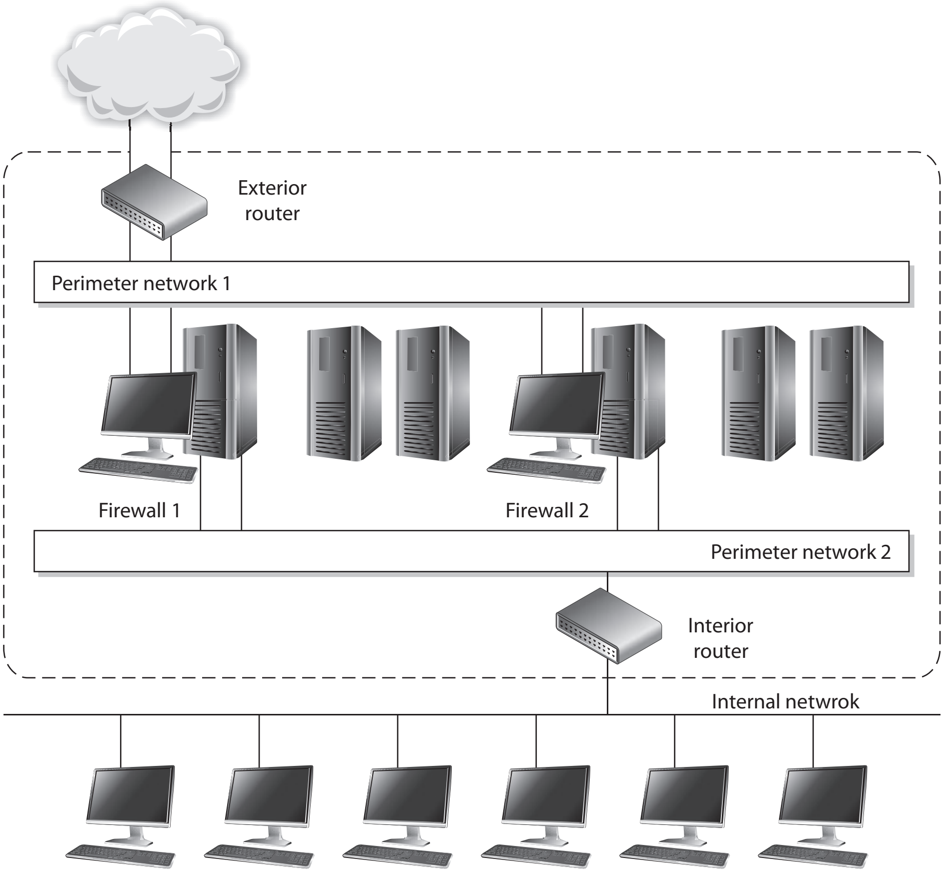


Figure 21-10 A screened subnet can have different networks within it and different firewalls that filter for specific threats.

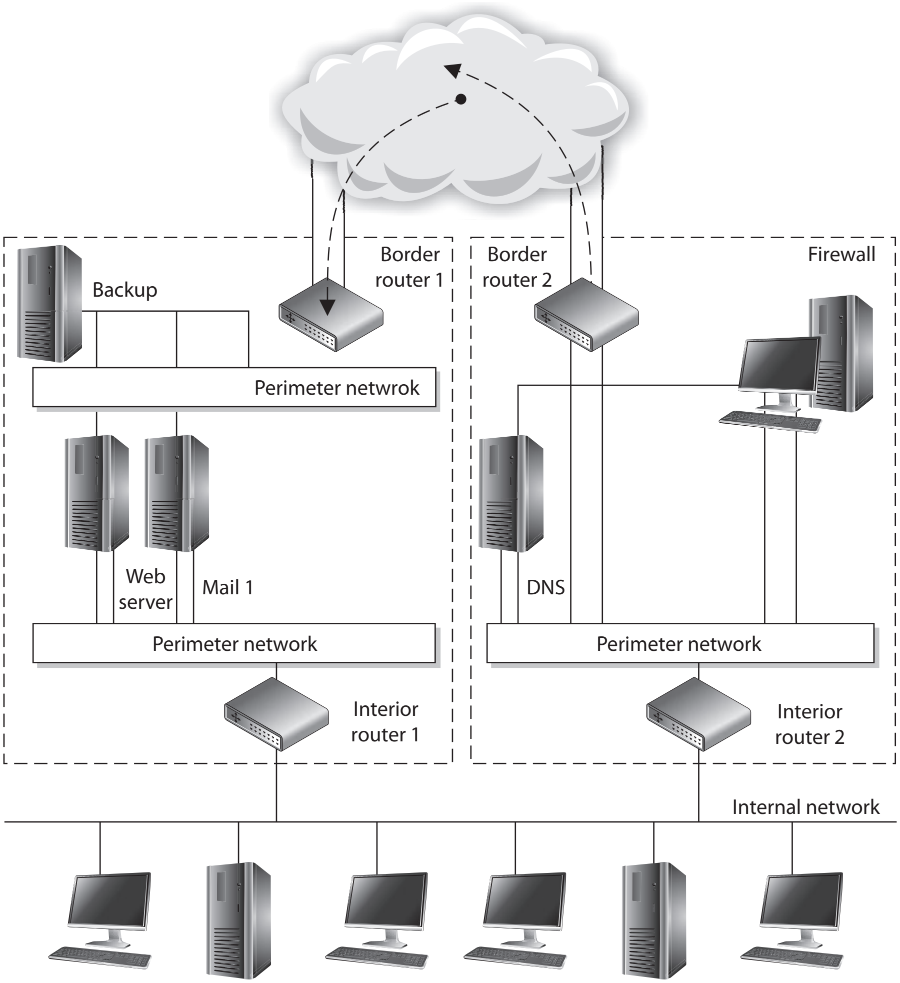


Figure 21-11 Some architectures have separate screened subnets with different server types in each.

The screened-subnet approach provides more protection than a stand-alone firewall or a screened-host firewall because three devices are working together and an attacker must compromise all three devices to gain access to the internal network. This architecture also sets up a DMZ between the two firewalls, which functions as a small network isolated among the trusted internal and untrusted external networks. The internal users usually

Firewall Architecture Characteristics

It is important to understand the following characteristics of these firewall architecture types:

Dual-homed:

- A single computer with separate NICs connected to each network.
- Used to divide an internal trusted network from an external untrusted network.
- Must disable a computer's forwarding and routing functionality so the two networks are truly segregated.

Screened host:

- A router filters (screens) traffic before it is passed to the firewall.

Screened subnet:

- An external router filters (screens) traffic before it enters the subnet. Traffic headed toward the internal network then goes through two firewalls.

have limited access to the servers within this area. Web, e-mail, and other public servers often are placed within the DMZ. Although this solution provides the highest security, it also is the most complex. Configuration and maintenance can prove to be difficult in this setup, and when new services need to be added, three systems may need to be reconfigured instead of just one.

TIP Sometimes a screened-host architecture is referred to as a single-tiered configuration and a screened subnet is referred to as a two-tiered configuration. If three firewalls create two separate DMZs, this may be called a three-tiered configuration.

Organizations used to deploy a piece of hardware for every network function needed (DNS, mail, routers, switches, storage, web), but today many of these items run within virtual machines on a smaller number of hardware machines. This reduces software and hardware costs and allows for more centralized administration, but these components still need to be protected from each other and external malicious entities. As an analogy, let's say that 15 years ago each person lived in their own house and a police officer was placed between each house so that the people in the houses could not attack each other. Then last year, many of these people moved in together so that now at least five people live in the same physical house. These people still need to be protected from each other, so some of the police officers had to be moved inside the houses to enforce the laws and keep the peace. Analogously, virtual firewalls have "moved into" the virtualized environments to provide the necessary protection between virtualized entities.

As illustrated in Figure 21-12, a network can have a traditional physical firewall on the physical network and *virtual firewalls* within the individual virtual environments.

Virtual firewalls can provide bridge-type functionality in which individual traffic links are monitored between virtual machines, or they can be integrated within the hypervisor. The hypervisor is the software component that carries out virtual machine management and oversees guest system software execution. If the firewall is embedded within the hypervisor, then it can "see" and monitor all the activities taking place within the system.

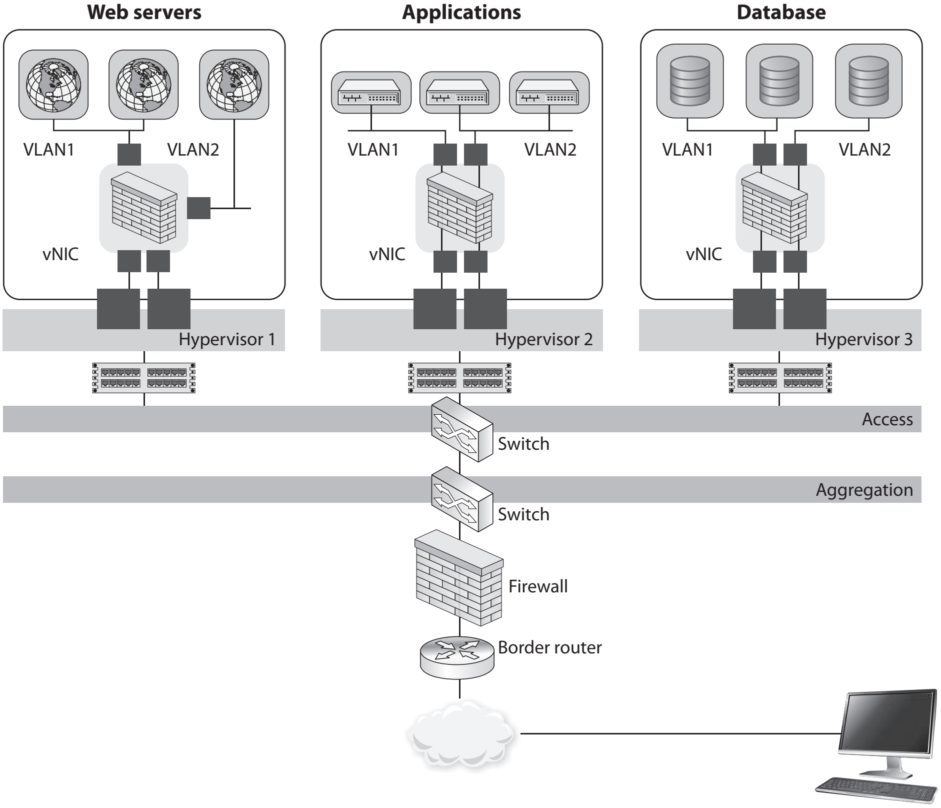


Figure 21-12 Virtual firewalls

Bastion Host

A system is considered a *bastion host* if it is a highly exposed device that is most likely to be targeted by attackers. The closer any system is to an untrusted network, such as the Internet, the more it is considered a target candidate since it has a smaller number of layers of protection guarding it. If a system is on the public side of a DMZ or is directly connected to an untrusted network, it is considered a bastion host; thus, it needs to be extremely locked down.

The system should have all unnecessary services disabled, unnecessary accounts disabled, unneeded ports closed, unused applications removed, unused subsystems and administrative tools removed, and so on. The attack surface of the system needs to be reduced, which means the number of potential vulnerabilities needs to be reduced as much as possible.

A bastion host does not have to be a firewall—the term just relates to the position of the system in relation to an untrusted environment and its threat of attack. Different systems can be considered bastion hosts (mail, web, DNS, etc.) if they are placed on the outer edges of networks.

The "Shoulds" of Firewalls

The default action of any firewall should be to implicitly deny any packets not explicitly allowed. This means that if no rule states that the packet can be accepted, that packet should be denied, no questions asked. Any packet entering the network that has a source address of an internal host should be denied. *Masquerading*, or *spoofing*, is a popular attacking trick in which the attacker modifies a packet header to have the source address of a host inside the network she wants to attack. This packet is spoofed and illegitimate. There is no reason a packet coming from the Internet should have an internal source network address, so the firewall should deny it. The same is true for outbound traffic. No traffic should be allowed to leave a network that does not have an internal source address. If this occurs, it means someone, or some program, on the internal network is spoofing traffic. This is how *zombies* work—the agents used in distributed DoS (DDoS) attacks. If packets are leaving a network with different source addresses, these packets are spoofed and the network is most likely being used as an accomplice in a DDoS attack.

Firewalls should reassemble fragmented packets before sending them on to their destination. In some types of attacks, the hackers alter the packets and make them seem to be something they are not. When a fragmented packet comes to a firewall, the firewall is seeing only part of the picture. It makes its best guess as to whether this piece of a packet is malicious or not. Because these fragments contain only a part of the full packet, the firewall is making a decision without having all the facts. Once all fragments are allowed through to a host computer, they can be reassembled into malicious packages that can cause a lot of damage. A firewall should accept each fragment, assemble the fragments into a complete packet, and then make an access decision based on the whole packet. The drawback to this, however, is that firewalls that do reassemble packet fragments before allowing them to go on to their destination computer cause traffic delay and more overhead. It is up to the organization to decide whether this configuration is necessary and whether the added traffic delay is acceptable.

Many organizations choose to deny network entrance to packets that contain source routing information, which was mentioned earlier. Source routing means that the packet decides how to get to its destination, not the routers in between the source and destination computer. Source routing moves a packet throughout a network on a predetermined path. The sending computer must know about the topology of the network and how to route data properly. This is easier for the routers and connection mechanisms in between, because they do not need to make any decisions on how to route the packet. However, it can also pose a security risk. When a router receives a packet that contains source routing information, the router assumes the packet knows what needs to be done and passes the packet on. In some cases, not all filters may be applied to the packet, and a network administrator may want packets to be routed only through a certain path and not the route a particular packet dictates. To make sure none of this misrouting happens, many firewalls are configured to check for source routing information within the packet and deny it if it is present.

Firewalls are not effective "right out of the box." You really need to understand the type of firewall being implemented and its configuration ramifications. For example, a firewall may have implied rules, which are used before the rules you configure. These implied rules might contradict your rules and override them. In this case, you may think that a certain traffic type is being restricted, but the firewall allows that type of traffic into your network by default.

The following list addresses some of the issues that need you need to understand as they pertain to firewalls:

- Most of the time a distributed approach needs to be used to control all network access points, which cannot happen through the use of just one firewall.
- Firewalls can present a potential bottleneck to the flow of traffic and a single point of failure threat.
- Some firewalls do not provide protection from malware and can be fooled by the more sophisticated attack types.
- Firewalls do not protect against sniffers or rogue wireless access points and provide little protection against insider attacks.

The role of firewalls is becoming more and more complex as they evolve and take on more functionality and responsibility. At times, this complexity works against security professionals because it requires them to understand and properly implement additional functionality. Without an understanding of the different types of firewalls and architectures available, many more security holes can be introduced, which lays out the welcome mat for attackers.

Intrusion Detection and Prevention Systems

The options for intrusion detection and prevention include host-based intrusion detection systems (HIDSs), network-based intrusion detection systems (NIDSs), and wireless intrusion detection systems (WIDSs). Each may operate in detection or prevention mode depending on the specific product and how it is employed. As a refresher, the main difference between an intrusion detection system (IDS) and an intrusion prevention system (IPS) is that an IDS only detects and reports suspected intrusions, while an IPS detects, reports, and stops suspected intrusions. How do they do this? There are two basic approaches: rule-based or anomaly-based.

Rule-Based IDS/IPS

Rule-based intrusion detection and prevention is the simplest and oldest technology. Essentially, we write rules (or subscribe to a service that writes them for us) and load those onto the system. The IDS/IPS monitors the environment in which it is placed, looking for anything that matches a rule. For example, suppose you have a signature for a particular piece of malware. You could create a rule that looks for any data that matches that signature and either raise an alert (IDS) or drop the data and generate the alert (IPS). Rule-based approaches are very effective when we know the telltale signs of an attack. But what if the attacker changes tools or procedures?

The main drawback of rule-based approaches to detecting attacks is that we need to have a rule that accurately captures the attack. This means someone got hacked, investigated the compromise, generated the rule, and shared it with the community. This process takes time and, until the rule is finalized and loaded, the system won't be effective against that specific attack. Of course, there's nothing stopping the adversary from slightly modifying tools or techniques to bypass your new rule either.

Anomaly-Based IDS/IPS

Anomaly-based intrusion detection and prevention uses a variety of approaches to detect things that don't look right. One basic approach is to observe the environment for some time to figure out what "normal" looks like. This is called the *training mode*. Once it has created a baseline of the environment, the IDS/IPS can be switched to *testing mode*, in which it compares observations to the baselines created earlier. Any observation that is significantly different generates an alert. For example, a particular workstation has a pattern of behavior during normal working hours and never sends more than, say, 10MB of data to external hosts during a regular day. One day, however, it sends out 100MB. That is pretty anomalous, so the IDS/IPS raises an alert (or blocks the traffic). But what if that was just the annual report being sent to the regulators?

The main challenge with anomaly-based approaches is that of *false positives*; that is, detecting intrusions when none happened. False positives can lead to fatigue and desensitizing the personnel who need to examine each of these alerts. Conversely, *false negatives* are events that the system incorrectly classifies as benign, delaying the response until the intrusion is detected through some other means. Obviously, both are bad outcomes.

EDR, NDR, and XDR

HIDS and antimalware features are increasingly being bundled into comprehensive *endpoint detection and response (EDR)* platforms. Similarly, NIDSs are evolving into *network detection and response (NDR)* products. These newer solutions do everything that HIDSs and NIDSs do, but also offer a host of other features such as combining rule-based and anomaly detection capabilities. *Extended detection and response (XDR)* platforms take this one step further by correlation of events across multiple sensors, both in the cloud and on premises, to get a more holistic view of what is going on in an environment.

Perhaps the most important step toward reducing errors is to baseline the system. *Baselining* is the process of establishing the normal patterns of behavior for a given network or system. Most of us think of baselining only in terms of anomaly-based IDSs because these typically have to go through a period of learning before they can determine what is anomalous. However, even rule-based IDSs should be configured in accordance with whatever is normal for an organization. There is no such thing as a one-size-fits-all *set* of IDS/IPS rules, though some *individual* rules may very well be applicable to all (e.g., detecting a known specimen of malware).

NOTE The term "perimeter" has lost some of its importance of late. While it remains an important concept in terms of security architecting, it can mislead some into imagining a wall separating us from the bad guys. A best practice is to assume the adversaries are already "inside the wire," which downplays the importance of a perimeter in security operations.

Whitelisting and Blacklisting

One of the most effective ways to tune detection platforms like IDS/IPS is to develop lists of things that are definitely benign and those that are definitely malicious. The platform, then, just has to figure out the stuff that is not on either list. A *whitelist* (more inclusively called an *allow list*) is a set of known-good resources such as IP addresses, domain names, or applications. Conversely, a *blacklist* (also known as a *deny list*) is a set of known-bad resources. In a perfect world, you would only want to use whitelists, because nothing outside of them would ever be allowed in your environment. In reality, we end up using them in specific cases in which we have complete knowledge of the acceptable resources. For example, whitelisting applications that can execute on a computer is an effective control because users shouldn't be installing arbitrary software on their own. Similarly, we can whitelist devices that are allowed to attach to our networks.

Things are different when we can't know ahead of time all the allowable resources. For example, it is a very rare thing for an organization to be able to whitelist websites for every user. Instead, we would rely on blacklists of domain and IP addresses. The problem with blacklists is that the Internet is such a dynamic place that the only thing we can be sure of is that our blacklist will always be incomplete. Still, blacklisting is better than nothing, so we should always try to use whitelists first, and then fall back on blacklists when we have no choice.

Antimalware Software

Traditional antimalware software uses signatures to detect malicious code. Signatures, sometimes referred to as fingerprints, are created by antimalware vendors. A *signature* is a set of code segments that a vendor has extracted from a malware sample. Similar to how our bodies have antibodies that identify and go after specific pathogens by matching segments of their genetic codes, antimalware software has an engine that scans files, e-mail messages, and other data passing through specific protocols and then compares them to its database of signatures. When there is a match, the antimalware software carries out whatever activities it is configured to do, which can be to quarantine the item, attempt to clean it (remove the malware), provide a warning message dialog box to the user, and/or log the event.

Signature-based detection (also called *fingerprint detection*) is a reasonably effective way to detect conventional malware, but it has a delayed response time to new threats. Once malware is detected in the wild, the antimalware vendor must study it, develop and test a new signature, release the signature, and all customers must download it. If the malicious code is just sending out silly pictures to all of your friends, this delay is not so critical. If the malicious software is a new variant of TrickBot (a versatile Trojan behind many ransomware attacks), this amount of delay can be devastating.

Since new malware is released daily, it is hard for the signature-based vendors to keep up. Another technique that almost all antimalware software products use is referred to as *heuristic detection*. This approach analyzes the overall structure of the malicious code, evaluates the coded instructions and logic functions, and looks at the type of data within the virus or worm. So, it collects a bunch of information about this piece of code and assesses the likelihood of it being malicious in nature. It has a type of "suspiciousness counter," which is incremented as the program finds more potentially malicious attributes. Once a predefined threshold is met, the code is officially considered dangerous and the antimalware software jumps into action to protect the system. This allows antimalware software to detect unknown malware, instead of just relying on signatures.

As an analogy, let's say Barney is the town cop who is employed to root out the bad guys and lock them up (quarantine). If Barney uses a signature method, he compares a stack of photographs of bad actors to each person he sees on the street. When he sees a match, he quickly throws the bad guy into his patrol car and drives off. By contrast, if he uses a heuristic method, he watches for suspicious activity. So if someone with a ski mask is standing outside a bank, Barney assesses the likelihood of this being a bank robber against it just being a cold guy in need of some cash.

Some antimalware products create a simulated environment, called a *virtual machine* or *sandbox*, and allow some of the logic within the suspected code to execute in the protected environment. This allows the antimalware software to see the code in question in action, which gives it more information as to whether or not it is malicious.

NOTE The virtual machine or sandbox is also sometimes referred to as an *emulation buffer*. They are all the same thing—a piece of memory that is segmented and protected so that if the code is malicious, the system is protected.

Reviewing information about a piece of code is called *static analysis*, while allowing a portion of the code to run in a virtual machine is called *dynamic analysis*. They are both considered heuristic detection methods.

Now, even though all of these approaches are sophisticated and effective, they are not 100 percent effective because malware writers are crafty. It is a continual cat-and-mouse game that is carried out every day. The antimalware industry comes out with a new way of detecting malware, and the very next week the malware writers have a way to get around this approach. This means that antimalware vendors have to continually increase the intelligence of their products and you have to buy a new version every year.

The next phase in the antimalware software evolution is referred to as behavior blockers. Antimalware software that carries out *behavior blocking* actually allows the suspicious code to execute within the operating system unprotected and watches its interactions with the operating system, looking for suspicious activities. The antimalware software watches for the following types of actions:

- Writing to startup files or the Run keys in the Windows registry
- Opening, deleting, or modifying files
- Scripting e-mail messages to send executable code
- Connecting to network shares or resources
- Modifying an executable logic
- Creating or modifying macros and scripts
- Formatting a hard drive or writing to the boot sector

If the antimalware program detects some of these potentially malicious activities, it can terminate the software and provide a message to the user. The newer-generation behavior blockers actually analyze sequences of these types of operations before determining the system is infected. (The first-generation behavior blockers only looked for individual actions, which resulted in a large number of false positives.) The newer-generation software can intercept a dangerous piece of code and not allow it to interact with other running processes. They can also detect rootkits. In addition, some of these antimalware programs can allow the system to roll back to a state before an infection took place so the damages inflicted can be "erased."

While it sounds like behavior blockers might bring us our well-deserved bliss and utopia, one drawback is that the malicious code must actually execute in real time; otherwise, our systems can be damaged. This type of constant monitoring also requires a high level of system resources. We just can't seem to win.

EXAM TIP Heuristic detection and behavior blocking are considered proactive and can detect new malware, sometimes called "zero-day" attacks. Signature-based detection cannot detect new malware.

Most antimalware vendors use a blend of all of these technologies to provide as much protection as possible. The individual antimalware attack solutions are shown in Figure 21-13.

NOTE Another antimalware technique is referred to as *reputation-based protection*. An antimalware vendor collects data from many (or all) of its customers' systems and mines that data to search for patterns to help identify good and bad files. Each file type is assigned a reputation metric value, indicating the probability of it being "good" or "bad." These values are used by the antimalware software to help it identify "bad" (suspicious) files.

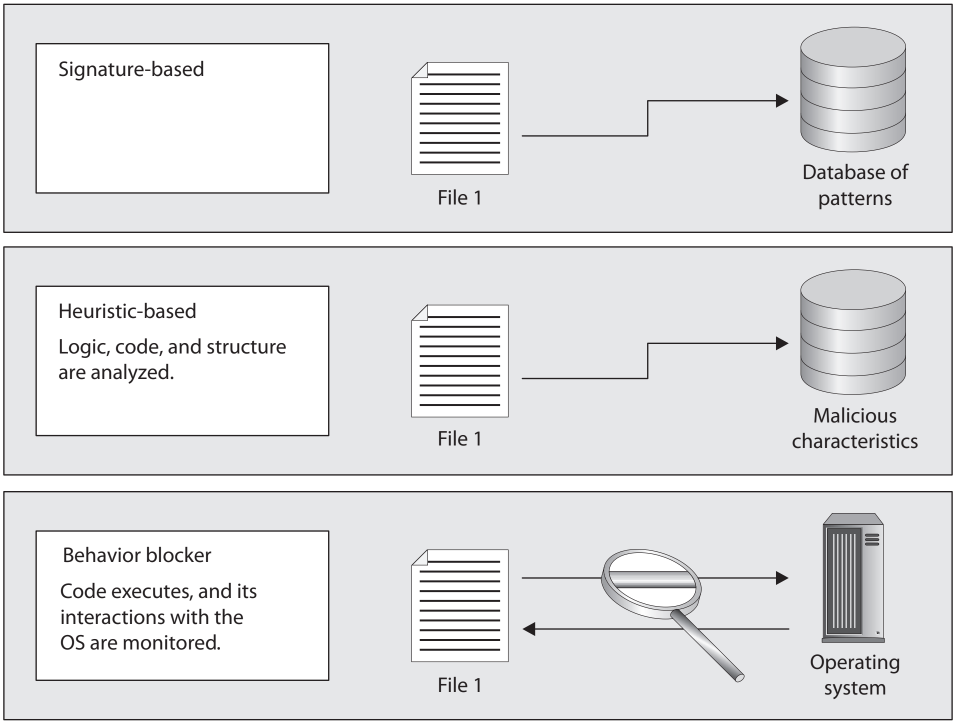

Signature-based
File 1
Database of patterns
Heuristic-based
Logic, code, and structure are analyzed.
File 1
Malicious characteristics
Behavior blocker
Code executes, and its interactions with the OS are monitored.
File 1
Operating system

Figure 21-13 Antimalware vendors use various types of malware detection.

Detecting and protecting an enterprise from the long list of malware requires more than just rolling out antimalware software. Just as with other pieces of a security program, certain administrative, physical, and technical controls must be deployed and maintained.

The organization should either have a stand-alone antimalware policy or have one incorporated into an existing security policy. It should include standards outlining what type of antimalware software and antispyware software should be installed and how they should be configured.

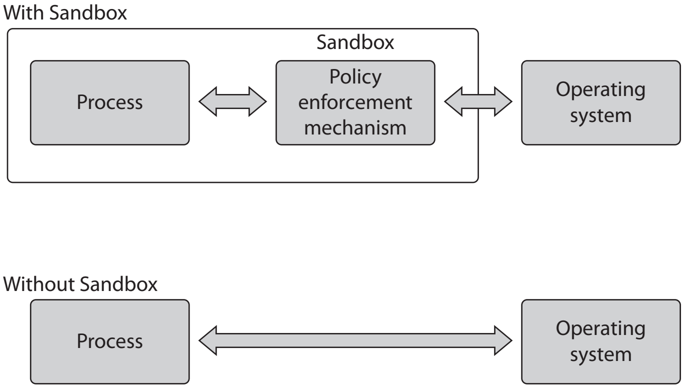
Antimalware information and expected user behaviors should be integrated into the security-awareness program, along with who users should contact if they discover a virus. A standard should cover the do's and don'ts when it comes to malware, which are listed next:

- Every workstation, server, and mobile device should have antimalware software installed.
- An automated way of updating malware signatures should be deployed on each device.
- Users should not be able to disable antimalware software.
- A preplanned malware eradication process should be developed and a contact person designated in case of an infection.
- All external disks (USB drives and so on) should be scanned automatically.
- Backup files should be scanned.
- Antimalware policies and procedures should be reviewed annually.
- Antimalware software should provide boot malware protection.
- Antimalware scanning should happen at a gateway and on each device.
- Virus scans should be automated and scheduled. Do not rely on manual scans.
- Critical systems should be physically protected so malicious software cannot be installed locally.

Since malware has cost organizations millions of dollars in operational costs and productivity hits, many have implemented antimalware solutions at network entry points. The scanning software can be integrated into a mail server, proxy server, or firewall. (The solutions are sometimes referred to as *virus walls*.) This software scans incoming traffic, looking for malware so it can be detected and stopped before entering the network. These products can scan Simple Mail Transport Protocol (SMTP), HTTP, FTP, and possibly other protocol types, but what is important to realize is that the product is only looking at one or two protocols and not *all* of the incoming traffic. This is the reason each server and workstation should also have antimalware software installed.

Sandboxing

A *sandbox* is an application execution environment that isolates the executing code from the operating system to prevent security violations. To the code, the sandbox looks just like the environment in which it would expect to run. For instance, when we sandbox an application, it behaves as if it were communicating directly with the OS. In reality, it is interacting with another piece of software whose purpose is to ensure compliance with security policies. Another instance is that of software (such as helper objects) running in a web browser. The software acts as if it were communicating directly with the browser, but those interactions are mediated by a policy enforcer of some sort. The power of sandboxes is that they offer an additional layer of protection when running code that we are not certain is safe to execute.



Outsourced Security Services

Nearly all of the preventive and detective measures we've discussed in the preceding subsections can be outsourced to an external service provider. Why would we want to do that? Well, for starters, many small and midsized organizations lack the resources to provide a full team of experienced security professionals. We are experiencing workforce shortages that are not likely to be solved in the near term. This means that hiring, training, and retaining qualified personnel is not feasible in many cases. Instead, many organizations have turned to managed security services providers (MSSPs) for third-party provided security services.

EXAM TIP Outsourced security services are what (ISC)^2 refers to as *third-party provided security*.

MSSPs typically offer a variety of services ranging from point solutions to taking over the installation, operation, and maintenance of all technical (and some cases physical) security controls. (Sorry, you still have to provide policies and many administrative controls.) Your costs will vary depending on what you need but, in many cases, you'll get more than you could've afforded if you were to provide these services in-house. Still, there are some issues that you should consider before hiring an MSSP:

- **Requirements** Before you start interviewing potential MSSPs, make sure you know your requirements. You can outsource the day-to-day activities, but you can't outsource your responsibility to understand your own security needs.

- **Understanding** Does the MSSP understand your business processes? Are they asking the right questions to get there? If your MSSP doesn't know what it is that your organization does (and how), they will struggle to provide usable security. Likewise, you need to understand their qualifications and processes. Trust is a two-way street grounded on accurate information.
- **Reputation** It is hard to be a subpar service provider and not have customers complain about you. When choosing an MSSP, you need to devote some time to reading online reviews and asking other security professionals about their experiences with specific companies.
- **Costing** You may not be able to afford the deluxe version of the MSSP's services, so you will likely have to compromise and address only a subset of your requirements. When you have trimmed down your requirements, is it still more cost-effective to go with this provider? Should you go with another? Should you just do it yourself?
- **Liability** Any reasonable MSSP will put limits on their liability if your organization is breached. Read the fine print on the contract and consult your attorneys, particularly if you are in an industry that is regulated by the government.

Honeypots and Honeynets

A *honeypot* is a network device that is intended to be exploited by attackers, with the administrator's goal being to gain information on the attackers' tactics, techniques, and procedures (TTPs). Honeypots can work as early detection mechanisms, meaning that the network staff can be alerted that an intruder is attacking a honeypot system, and they can quickly go into action to make sure no production systems are vulnerable to that specific attack type. A honeypot usually sits in the screened subnet, or DMZ, and attempts to lure attackers to it instead of to actual production computers. Think of honeypots as marketing devices; they are designed to attract a segment of the market, get them to buy something, and keep them coming back. Meanwhile, threat analysts are keeping tabs on their adversaries' TTPs.

To make a honeypot system alluring to attackers, administrators may enable services and ports that are popular to exploit. Some honeypot systems *emulate* services, meaning the actual services are not running but software that acts like those services is available. Honeypot systems can get an attacker's attention by advertising themselves as easy targets to compromise. They are configured to look like the organization's regular systems so that attackers will be drawn to them like bears are to honey.

Another key to honeypot success is to provide the right kind of bait. When someone attacks your organization, what is it that they are after? Is it credit card information, patient files, intellectual property? Your honeypots should look like systems that would allow the attacker to access the assets for which they are searching. Once compromised, the directories and files containing this information must appear to be credible. It should also take a long time to extract the information, so that we maximize the contact time with our "guests."

A *honeynet* is an entire network that is meant to be compromised. While it may be tempting to describe honeynets as networks of honeypots, that description might be a bit misleading. Some honeynets are simply two or more honeypots used together. However, others are designed to ascertain a specific attacker's intent and dynamically spawn honeypots that are designed to be appealing to that particular attacker. As you can see, these very sophisticated honeynets are not networks of preexisting honeypots, but rather adaptive networks that interact with the adversaries to keep them engaged (and thus under observation) for as long as possible.

NOTE *Black holes* are sometimes confused with honeynets, when in reality they are almost the opposite of them. Black holes typically are routers with rules that silently drop specific (typically malicious) packets without notifying the source. They normally are used to render botnet and other known-bad traffic useless. Whereas honeypots and honeynets allow us to more closely observe our adversaries, black holes are meant to make them go away for us.

Wrapping up the honey collection, *honeyclients* are synthetic applications meant to allow an attacker to conduct a client-side attack while also allowing the threat analysts an opportunity to observe the TTPs being used by their adversaries. Honeyclients are particularly important in the honey family, because most of the successful attacks happen on the client side, and honeypots are not particularly well suited to track client-side attacks. Suppose you have a suspected phishing or spear phishing attack that you'd like to investigate. You could use a honeyclient to visit the link in the e-mail and pretend it is a real user. Instead of getting infected, however, the honeyclient safely catches all the attacks thrown at it and reports them to you. Since it is not really the web browser it is claiming to be, it is impervious to the attack and provides you with information about the actual tools the attacker is throwing at you. Honeyclients come in different flavors, with some being highly interactive (meaning a human has to operate them), while others involve low interaction (meaning their behavior is mostly or completely automated).

Organizations use these systems to identify, quantify, and qualify specific traffic types to help determine their danger levels. The systems can gather network traffic statistics and return them to a centralized location for better analysis. So as the systems are being attacked, they gather intelligence information that can help the network staff better understand what is taking place within their environment.

It should be clear from the foregoing that honeypots and honeynets are not defensive controls like firewalls and IDSs, but rather help us collect threat intelligence. To be effective, they must be closely monitored by a competent threat analyst. By themselves, honeypots and honeynets do not improve your security posture. However, they can give your threat intelligence team invaluable insights into your adversaries' methods and capabilities.

It is also important to make sure that the honeypot systems are not connected to production systems and do not provide any "jumping off" points for the attacker. There have been instances where companies improperly implemented honeypots and they were exploited by attackers, who were then able to move from those systems to the company's internal systems. The honeypots need to be properly segmented from any other live systems on the network.

On a smaller scale, organizations may choose to implement *tarpits*, which are similar to honeypots in that they appear to be easy targets for exploitation. A tarpit can be configured to appear as a vulnerable service that attackers commonly attempt to exploit. Once the attackers start to send packets to this "service," the connection to the victim system seems to be live and ongoing, but the response from the victim system is slow and the connection may time out. Most attacks and scanning activities take place through automated tools that require quick responses from their victim systems. If the victim systems do not reply or are very slow to reply, the automated tools may not be successful because the protocol connection times out.

NOTE Deploying honeypots and honeynets has potential liability issues. Be sure to consult your legal counsel before starting down this road.

Artificial Intelligence Tools

Artificial intelligence (AI) is a multidisciplinary field primarily associated with computer science, with influences from mathematics, cognitive psychology, philosophy, and linguistics (among others). At a high level, AI can be divided into two different approaches, as shown in Figure 21-14: symbolic and non-symbolic; the key difference is in how each represents knowledge. Both approaches are concerned with how knowledge is organized, how inference proceeds to support decision-making, and how the system learns.

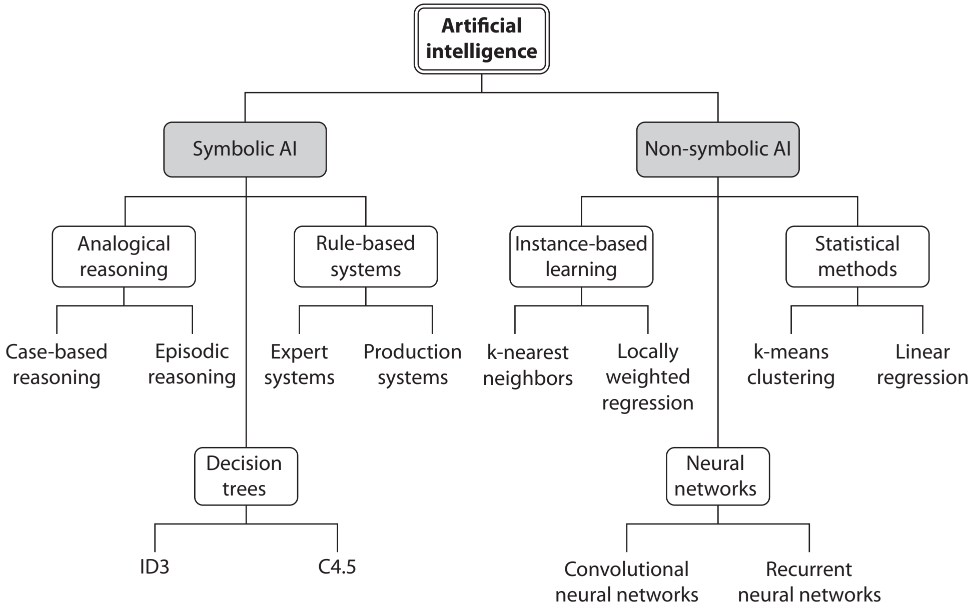


Figure 21-14 A partial taxonomy of artificial intelligence

In symbolic approaches to AI, system developers model real-world concepts, their relationships, and how they interact to solve a set of problems using a set of symbols (e.g., words or tokens). Symbolic AI requires considerable knowledge engineering of both the problem and solution domains, which makes it labor-intensive. However, it yields results that are inherently explainable to humans since the results are derived from human knowledge models in the first place. Symbolic AI systems include the expert systems that became prolific in the 1980s. These relied on extensive interviewing of subject matter experts and time-consuming encoding of their expertise in a series of conditional structures. Unsurprisingly, these early systems were unable to adapt or learn absent human intervention, which is a problem when we consider the number of exceptions that apply to almost all processes.

Another approach to AI departs from the use of symbolic representations of human knowledge and focuses instead on learning patterns in data for classifying objects, predicting future results, or clustering similar sets of data. These non-symbolic AI approaches are where many of the most recent advances have occurred, primarily in classification tasks such as image and voice recognition. In the current vernacular, these non-symbolic approaches are commonly called *machine learning (ML)* even though symbolic systems may also learn. As with symbolic approaches, non-symbolic ML systems also incorporate knowledge representations and reasoning. The knowledge representation is typically quantitative vectors (i.e., non-symbolic) with features from the dataset that describe the input (e.g., pixels from an image, frequencies from an audio file, word vectors, etc.).

Whereas symbolic AI requires considerable knowledge engineering, non-symbolic AI generally requires significant data acquisition and data curating, which can be labor-intensive even for domains where data is readily available. However, rather than having to program the knowledge, as in a symbolic system, the non-symbolic ML system acquires its knowledge in the form of numeric parameters (i.e., weights) through offline training with datasets with millions of examples. As training progresses, the ML model learns the correct parameters that minimize a cost function. That function typically deals with classifying some sample (helpful for finding malware) or making a prediction (allowing us to detect anomalies like spikes in outbound traffic).

Classification determines the class of a new sample based on what is known about previous samples. A common example of this is an algorithm called k-nearest neighbors (KNN), which is a supervised learning technique in which the nearest k neighbors influence the classification of the new point (e.g., if more than half of its k nearest neighbors are in one class, then the new point also belongs in that class). For cybersecurity, this is helpful when trying to determine whether a binary file is malware or detecting whether an e-mail is spam.

Prediction compares previous data samples and determines what the next sample(s) should be. If you have ever taken a statistics class, you may recall a type of analysis called *regression*, in which you try to determine the line (or curve) that most closely approximates a sequence of data points. We use the same approach to prediction in ML by learning from previous observations to determine where the next data point(s) should appear, which is useful for network flow analysis.

On the other hand, there is also unsupervised learning such as clustering, where we do not have a preconception of which classes (or even how many) exist; we determine where the samples naturally clump together. One of the most frequently used clustering algorithms is k-means clustering, in which new data points are added to one of the k clusters based on which one is closest to the new point. Clustering is useful for anomaly detection.

Finally, reinforcement learning tunes decision-making parameters toward choices that lead to positive outcomes in the environment. For example, one might have a security analyst provide feedback to an anomaly detector when it incorrectly classifies a malicious file or event (i.e., a false positive). This feedback adjusts the internal model's weights, so that its anomaly classification improves.

AI has shortcomings that you must consider before employing it. Neither symbolic nor non-symbolic AI approaches cope well with novel situations, and both require a human to re-engineer (symbolic) or retrain (non-symbolic) the algorithms. Symbolic, knowledge-engineered systems may contain underlying biases of the individual(s) who encode the system. Training data sets for non-symbolic approaches may contain biases that are not representative of the operational environment. These biases lead to either false positives or, worse, false negatives when the system is deployed. The best way forward is to combine both approaches, using each other's strengths to offset the other's weaknesses.

Logging and Monitoring

Logging and monitoring are two key activities performed by a SOC using the various tools we just discussed (and probably a few others). These two tasks go hand in hand, since you can't really monitor (at least not very effectively) if you are not logging and, conversely, logging makes little sense if you aren't monitoring. In the sections that follow, we first address how to collect and manage logs, and then discuss the ways in which you should be monitoring those logs (as well as other real-time data feeds).

Log Management

We discussed log reviews and how to prevent log tampering in Chapter 18. To understand how logs support day-to-day security operations, however, we need to take a step back and review why we might be logging system events in the first place. After all, if you don't have clear goals in mind, you will likely collect the wrong events at least some of the time.

Logging Requirements

Earlier in this chapter, we discussed cyberthreat intelligence and, in particular, the collection management framework (CMF). That section on the CMF is a great one to review when you're thinking about what your logging goals should be. After all, logs are data sources that can (and probably should) feed your threat intelligence. Just like intelligence requirements are meant to answer questions from decision-makers, logs should do the same for your SOC analysts. There should be specific questions your security team routinely asks, and those are the questions that should drive what you log and how. For example, you may be concerned about data leaks of your sensitive research projects to overseas threat actors. What events from which system(s) would you need to log in order to monitor data egress? How often will you be checking logs (which determines how long you must retain them)? If you simply go with default logging settings, you may be ill informed when it comes to monitoring.

Log Standards

Another best practice is to standardize the format of your logs. If you are using a security information and event management (SIEM) system (which we'll discuss shortly), then that platform will take care of normalizing any logs you forward to it. Otherwise, you'll have to do it yourself using either the configuration settings on the system that's logging (if it allows multiple formats) or by using a data processing pipeline such as the open-source Logstash.

NOTE It is essential that you standardize the timestamps on all logs across your environment. If your organization is small, you can use local time; otherwise, we recommend you always use Coordinated Universal Time (UTC).

Something else to consider as you standardize your logs is who will be consuming them. Many SOCs leverage tools for automation, such as some of the AI techniques we discussed earlier. These automated systems may have their own set of requirements for formatting, frequency of updates, or log storage. You should ensure that your standards address the needs of all stakeholders (even non-human ones).

Logging Better

Finally, as with anything else you do in cybersecurity, you want to evaluate the effectiveness of your log management efforts and look for ways to sustain what you're doing well and improve the rest. Establishing and periodically evaluating metrics is an excellent approach to objectively determine opportunities for improvement. For example, how often do analysts lack information to classify an event because of incomplete logging? What logs, events, and fields are most commonly used when triaging alerts? Which are never needed? These questions will point to metrics, and the metrics, in turn, will tell you how well your logging supports your goals.

Security Information and Event Management

A *security information and event management (SIEM)* system is a software platform that aggregates security information (like asset inventories) and security events (which could become incidents) and presents them in a single, consistent, and cohesive manner. SIEMs collect data from a variety of sensors, perform pattern matching and correlation of events, generate alerts, and provide dashboards that allow analysts to see the state of the network. One of the best-known commercial solutions is Splunk, while on the open-source side the Elastic Stack (formerly known as the Elasticsearch-Logstash-Kibana, or ELK, stack) is very popular. It is worth noting that, technically, both of these systems are data analytics platforms and not simply SIEMs. Their ability to ingest, index, store, and retrieve large volumes of data applies to a variety of purposes, from network provisioning to marketing to enterprise security.

Among the core characteristics of SIEMs is the ability to amass all relevant security data and present it to the security analyst in a way that makes sense. Before these devices became mainstream, security personnel had to individually monitor a variety of systems and manually piece together what all this information might mean. Most SIEMs now include features that group together information and events that seem to be related to each other (or "correlated" in the language of statistics). This allows the analyst to quickly determine the events that are most important or for which there is the most evidence.

SIEM correlations require a fair amount of fine-tuning. Most platforms, out of the box, come with settings that are probably good enough to get you started. You'll have to let your SIEM tool run for a while (one week or longer) for it to start making sense of your environment and giving you meaningful alerts. Inevitably, you'll find that your analysts are drowning in false positives (sadly, a very common problem with automated platforms) that consume their time and joy. This is where you start tuning your settings using things like whitelists and analyst ratings that will make the platform more accurate. You may also discover blind spots (that is, incidents that your SIEM did not pick up) due to insufficient logging or inadequate sensor placement, so you tune a bit there too.

NOTE SIEM fine-tuning should follow your established configuration management processes.

Security Orchestration, Automation, and Response

A tool that is becoming increasingly popular in SOCs is the security orchestration, automation, and response (SOAR) platform. SOAR is an integrated system that enables more efficient security operations through automation of various workflows. The following are the three key components of a SOAR solution:

- **Orchestration** This refers to the integration and coordination of other security tools such as firewalls, IDS/IPS, and SIEM platforms. Orchestration enables automation.
- **Automation** SOAR platforms excel at automating cybersecurity playbooks and workflows, driving significant efficiency gains where those processes exist (or are created).
- **Response** Incident response workflows can involve dozens (or even hundreds) of distinct tasks. A SOAR platform can automatically handle many of those, freeing up the incident responders to work on what humans do best.

Egress Monitoring

A security practice that is oftentimes overlooked by smaller organizations is *egress monitoring*, which is keeping an eye on (and perhaps restricting) the information that is flowing *out* of our networks. Chapter 6 introduced data loss prevention (DLP), which is a very specific use case of this. Beyond DLP, we should be concerned about ensuring that our platforms are not being used to attack others and that our personnel are not communicating (knowingly or otherwise) with unsavory external parties.

A common approach to egress monitoring is to allow only certain hosts to communicate directly with external destinations. This allows us to focus our attention on a smaller set of computers that presumably would be running some sort of filtering software. A good example of this approach is the use of a web gateway, which effectively implements a man-in-the-middle "attack" on all of our organization's web traffic. It is not uncommon to configure these devices to terminate (and thus decrypt) all HTTPS traffic and to do deep packet inspection (DPI) before allowing information to flow out of the network.

User and Entity Behavior Analytics

While most attacks historically are caused by external threat actors, we must not neglect to monitor the activities of users and entities within our organizations. Even if we never encounter a malicious insider, our users are oftentimes unwitting accomplices when they visit the wrong site, click the wrong link, or open the wrong attachment. *User and entity behavior analytics (UEBA)* is a set of processes that determines normal patterns of behavior so that abnormalities can be detected and investigated. For example, if a user hardly ever sends large amounts of data out to the Internet and then one day starts sending megabytes' worth, that would trigger a UEBA alert. Maybe the transmission was perfectly legitimate, but perhaps it was the early part of a data loss incident.

UEBA can exist as a stand-alone product or as a feature in some other tool, such as an EDR or NDR platform. Either way, UEBA uses machine learning to predict future behaviors based on past observations, and statistical analyses to determine when a deviation from the norm is significant enough to raise an alert. As with any other type of solution that offers behavioral analytics, UEBA solutions are prone to false positives. This means that you would probably need to put some effort into fine-tuning a UEBA solution, even after its training period.

EXAM TIP UEBA is a good choice for detecting both malicious insiders and benign user accounts that have been taken over by a malicious actor.

Continuous Monitoring

NIST Special Publication 800-137, *Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations*, defines *information security continuous monitoring* as "maintaining ongoing awareness of information security, vulnerabilities, and threats to support organizational risk management decisions." Think of ISCM as an ongoing and structured verification of security controls. Are the existing controls still the right ones? Are they still effective? If not, why? These are some of the questions to which continuous monitoring provides answers. It is a critical part of the risk management framework we covered in Chapter 2.

There is a distinction here between logging, monitoring, and continuous monitoring. Your logging policies should be pretty expansive. Data storage is cheap and you want to capture as much data as you can in case you ever need it. Monitoring is more limited because it typically requires a human to personally do it, or at least to deal with the reports (such as SIEM alerts) that come out of it. You would, for example, monitor traffic on a certain port when it looks suspicious and then move on to monitoring something else when you determine that traffic is benign. Continuous monitoring is much more prescriptive. It is a deliberate, risk-based process to determine what gets monitored, how it is monitored, and what to do with the information you gather.

In the end, the whole point of continuous monitoring is to determine if the controls remain effective (in the face of changing threat and organizational environments) at reducing risk to acceptable levels. To do this, you need to carefully consider which metrics would allow you to say "yes" or "no" for each control. For example, suppose you are concerned about the risk of malware infections in your organization, so you implement antimalware controls. As part of continuous monitoring for those controls, you could measure the number of infections in some unit of time (day, week, month).

The metrics and measurements provide data that must be analyzed in order to make it actionable. Continuing our malware example, if your controls are effective, you would expect the number of infections to remain steady over time or (ideally) decrease. You would also want to consider other information in the analysis. For example, your malware infections could go up if your organization goes through a growth spurt and hires a bunch of new people, or the infections could go down during the holidays because many employees are taking vacation. The point is that the analysis is not just about understanding what is happening, but also why.

Finally, continuous monitoring involves deciding how to respond to the findings. If your organization's malware infections have increased and you think this is related to the surge in new hires, should you provide additional security awareness training or replace the antimalware solution? Deciding what to do about controls that are no longer sufficiently effective must take into account risk, cost, and a host of other organizational issues.

Continuous monitoring is a deliberate process. You decide what information you need, then collect and analyze it at a set frequency, and then make business decisions with that information. Properly implemented, this process is a powerful tool in your prevention kit.

Chapter Review

Most of the time spent by the typical organization conducting security operations is devoted to emplacing and maintaining the preventive and detective measures, and then using those to log events and monitor the environment. Entire books have been written on these topics, so in this chapter we just covered the essentials. A key takeaway is that tools alone will never be enough to give you the visibility you need to detect attacks; you need the integration of people, processes, and technology. We may have put a bit more focus on technology in this chapter, but we wanted to close it by highlighting the fact that well-trained people, working as a team and following existing processes, are essential components of security operations. This is particularly true when things go wrong and we need to respond to incidents, which we're about to cover in the next chapter.

Quick Review

- The security operations center (SOC) encompasses the people, processes, and technology that allow logging and monitoring of preventive controls, detection of security events, and incident response.
- Tier 1 security analysts spend most of their time monitoring security tools and other technology platforms for suspicious activity.
- Tier 2 security analysts dig deeper into the alerts, declare security incidents, and coordinate with incident responders and intelligence analysts to further investigate, contain, and eradicate the threats.
- Threat intelligence is evidence-based knowledge about an existing or emerging menace or hazard to assets that can be used to inform decisions regarding responses to that menace or hazard.
- Threat intelligence is commonly derived from three types of sources: threat data feeds, open-source intelligence (OSINT), and internal systems.
- Cyberthreat hunting is the practice of proactively looking for threat actors in your networks.
- Firewalls support and enforce the organization's network security policy by restricting access to one network from another network.
- Packet-filtering firewalls make access decisions based upon network-level protocol header values using access control lists (ACLs).
- Stateful firewalls add to the capabilities of packet-filtering firewalls by keeping track of the state of a connection between two endpoints.
- Proxy firewalls intercept and inspect messages before delivering them to the intended recipients.
- A next-generation firewall (NGFW) combines the attributes of the previously discussed firewalls, but adds a signature-based and/or behavioral analysis IPS engine, as well as cloud-based threat data sharing.
- Intrusion detection and prevention systems (IDS/IPS) can be categorized as either host-based (HIDS) or network-based (NIDS) and rule-based or anomaly-based.
- A whitelist is a set of known-good resources such as IP addresses, domain names, or applications. Conversely, a blacklist is a set of known-bad resources.

- Antimalware software is most effective when it is installed in every entry and end point and covered by a policy that delineates user training as well as software configuration and updating.
- A sandbox is an application execution environment that isolates the executing code from the operating system to prevent security violations.
- A honeypot is a network device that is intended to be exploited by attackers, with the administrator's goal being to gain information on the attackers' tactics, techniques, and procedures.
- A honeynet is an entire network that is meant to be compromised.
- Honeyclients are synthetic applications meant to allow an attacker to conduct a client-side attack while also allowing the security analysts an opportunity to observe the techniques being used by their adversaries.
- Machine learning (ML) systems acquire their knowledge in the form of numeric parameters (i.e., weights), through training with datasets consisting of millions of examples. In supervised learning, ML systems are told whether or not they made the right decision. In unsupervised training, they learn by observing an environment. Finally, in reinforcement learning they get feedback on their decisions from the environment.
- Effective logging requires a standard time zone for all timestamps.
- A security information and event management (SIEM) system is a software platform that aggregates security information (like asset inventories) and security events (which could become incidents) and presents them in a single, consistent, and cohesive manner.
- Security orchestration, automation, and response (SOAR) platforms are integrated systems that enable more efficient security operations through automation of various workflows.
- Egress monitoring is the process of scanning (and perhaps restricting) the information that is flowing out of our networks.
- User and entity behavior analytics (UEBA) is a set of processes that determines normal patterns of behavior so that abnormalities can be detected and investigated.
- Continuous monitoring allows organizations to maintain ongoing awareness of information security, vulnerabilities, and threats to support organizational risk management decisions.

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.

Use the following scenario to answer Questions 1–3. The startup company at which you are the director of security is going through a huge growth spurt and the CEO has decided it's time to let you build out a security operations center (SOC). You already have two cybersecurity analysts (one is quite experienced), a brand-new security information and event management (SIEM) platform, and pretty good security processes in place.

1. The number of alerts on your SIEM is overwhelming your two analysts and many alerts go uninvestigated each day. How can you correct this?
   - **A.** Hire an intelligence analyst to help you focus your collection efforts.
   - **B.** Tune the SIEM platform to reduce false-positive alerts.
   - **C.** Establish a threat hunting program to find attackers before they trigger alerts.
   - **D.** Establish thresholds below which events will not generate alerts.
2. You hire an intelligence analyst and want her to start addressing intelligence requirements. Which of the following should be her first step?
   - **A.** Finding out what questions decision-makers need answered
   - **B.** Establishing a collection management framework
   - **C.** Identifying data sources
   - **D.** Subscribing to a threat data feed
3. Your SOC is maturing rapidly and you are ready to start a cyberthreat hunting program. Which of the following describes the crux of this effort?
   - **A.** Proving or negating hypotheses of threat actions based on threat intelligence
   - **B.** Neutralizing threat actors before they can breach your organization
   - **C.** Digging deeper into the alerts to determine if they constitute security incidents
   - **D.** Allowing hunters an opportunity to observe techniques used by their adversaries
4. A firewall that can only make decisions based on examining a single network layer header is called a
   - **A.** Stateful firewall
   - **B.** Screened host
   - **C.** Packet filter
   - **D.** Next-generation firewall
5. A firewall that understands the three-step handshake of a TCP connection is called a
   - **A.** Packet filter
   - **B.** Proxy firewall
   - **C.** Transport-layer proxy
   - **D.** Stateful firewall

6. What is the main challenge with anomaly-based approaches to intrusion detection and prevention?
   - **A.** False positives
   - **B.** Needing a rule that accurately captures the attack
   - **C.** Cost
   - **D.** Immaturity of the technology

7. Which of the following is an effective technique for tuning automated detection systems like IDS/IPS and SIEMs?
   - **A.** Access control lists
   - **B.** State tables
   - **C.** Whitelists
   - **D.** Supervised machine learning

8. Which of the following terms would describe a system designed to ascertain a specific attacker's intent and dynamically spawn multiple virtual devices that are designed to be appealing to that particular attacker?
   - **A.** Honeypot
   - **B.** Honeyclient
   - **C.** Honeyseeker
   - **D.** Honeynet

9. Which of the following is *not* a typical application of machine learning?
   - **A.** Classification
   - **B.** Prediction
   - **C.** Clustering
   - **D.** Knowledge engineering

10. Which of the following is *not* true about continuous monitoring?
    - **A.** It involves ad hoc processes that provide agility in responding to novel attacks.
    - **B.** Its main goal is to support organizational risk management.
    - **C.** It helps determine whether security controls remain effective.
    - **D.** It relies on carefully chosen metrics and measurements.

Answers

1. **B.** False positives are a very common problem with automated platforms like SIEMs, but they can be alleviated by fine-tuning the platform. An intelligence analyst could help a little bit but would clearly not be the best answer, while threat hunting would be a distractor for such a young SOC that still needs to get alerts

under control. Ignoring low-scoring alerts as a matter of policy would be a very dangerous move when dealing with stealthy attackers.

2. **A.** Threat intelligence is meant to help decision-makers choose what to do about a threat. It answers a question that these leaders may have. The CMF and data sources are all important, of course, but they are driven by the requirements that come out of leaders' questions. After the requirements are known, the intelligence analyst may (or may not) need to subscribe to a threat data feed.

3. **A.** The crux of threat hunting is to develop a hypothesis of adversarial action based on threat intelligence, and then to prove or negate the hypothesis. Inherent in this description are two factors: a) the adversary is already inside the network, and b) no alerts tipped off the defenders to the adversary's presence. These factors negate answers B and C. Answer D describes the purpose of a honeypot, not threat hunting.

4. **C.** Packet filtering is a firewall technology that makes access decisions based upon network-level protocol header values. The device that is carrying out packet-filtering processes is configured with access control lists (ACLs), which dictate the type of traffic that is allowed into and out of specific networks.

5. **D.** Stateful firewalls keep track of the state of a protocol connection, which means they understand the three-step handshake a TCP connection goes through (SYN, SYN/ACK, ACK).

6. **A.** The main challenge with anomaly-based approaches is that of false positives—detecting intrusions when none happened. These can lead to fatigue and desensitizing the personnel who need to examine each of these alerts. Despite this shortcoming, anomaly-based approaches are mature and cost-effective technologies that are differentiated from rule-based systems by not needing rules that accurately capture attacks.

7. **C.** One of the most effective ways to tune detection platforms like IDS/IPS is to develop lists of things that are definitely benign and those that are definitely malicious. The platform, then, just has to figure out the stuff that is not on either list. A whitelist (more inclusively called an allow list) is a set of known-good resources such as IP addresses, domain names, or applications.

8. **D.** Some honeynets are designed to ascertain a specific attacker's intent and dynamically spawn honeypots that are designed to be appealing to that particular attacker. These very sophisticated honeynets are not networks of preexisting honeypots, but rather adaptive networks that interact with the adversaries to keep them engaged (and thus under observation) for as long as possible.

9. **D.** Machine learning (ML), which is a non-symbolic approach to artificial intelligence (AI), is typically used for classification and prediction (using supervised or semi-supervised learning) as well as clustering (using unsupervised learning). Knowledge engineering is a requirement for symbolic forms for AI, such as expert systems, which are not ML in the common sense of the term.

10. **A.** Continuous monitoring is a deliberate, data-driven process supporting organizational risk management. One of the key questions it answers is whether controls are still effective at mitigating risks. Continuous monitoring could potentially lead to a decision to implement specific ad hoc processes, but these would not really be part of continuous monitoring.

CHAPTER 22

Security Incidents

This chapter presents the following:

- Incident management
- Incident response planning
- Investigations

It takes 20 years to build a reputation and few minutes of cyber-incident to ruin it.

—Stephane Nappo

No matter how talented your security staff may be, or how well everyone in your organization complies with your excellent security policies and procedures, or what cutting-edge technology you deploy, the sad truth is that the overwhelming odds are that your organization will experience a major compromise (if it hasn't already). What then? Having the means to manage incidents well can be just as important as anything else you do to secure your organization. In this chapter, we will cover incident management in general and then drill down into the details of incident response planning.

Although ISC^2 differentiates incident management and incident investigations, for many organizations, the latter is part of the former. This differentiation is useful to highlight the fact that some investigations involve suspects who may be our own colleagues. While many of us would enjoy the challenge of figuring out how an external threat actor managed to compromise our defenses, there is nothing fun about substantiating allegations that someone we work with did something wrong that caused losses to the organization. Still, as security professionals, we must be ready for whatever threats emerge and deal with the ensuing incidents well and rapidly.

Overview of Incident Management

There are many incident management models, but all share some basic characteristics. They all require that we identify the event, analyze it to determine the appropriate countermeasures, correct the problem(s), and, finally, take measures to keep the event from happening again. (ISC)^2 has broken out these four basic actions and prescribes seven phases in the incident management process: detection, response, mitigation, reporting, recovery, remediation, and lessons learned. Your own organization will have a unique approach, but it is helpful to baseline it off the industry standard.

Although we commonly use the terms “event” and “incident” interchangeably, there are subtle differences between the two. A *security event* is any occurrence that can be observed, verified, and documented. These events are not necessarily harmful. For example, a remote user login, changes to the Windows Registry on a host, and system reboots are all security events that could be benign or malicious depending on the context. A *security incident* is one or more related events that negatively affect the organization and/or impact its security posture. That remote login from our previous example could be a security incident if it was a malicious user logging in. We call reacting to these issues “incident response” (or “incident handling”) because something is negatively affecting the organization and causing a security breach.

EXAM TIP A security event is not necessarily a security violation, whereas a security incident is.

Many types of security incidents (malware, insider attacks, terrorist attacks, and so on) exist, and sometimes an incident is just human error. Indeed, many incident response individuals have received a frantic call in the middle of the night because a system is acting “weird.” The reasons could be that a deployed patch broke something, someone misconfigured a device, or the administrator just learned a new scripting language and rolled out some code that caused mayhem and confusion.

Many organizations are at a loss as to who to call or what to do right after they have been the victim of a cybercrime. Therefore, all organizations should have an *incident management policy (IMP)*. This document indicates the authorities and responsibilities regarding incident response for everyone in the organization. Though the IMP is frequently drafted by the CISO or someone on that person’s team, it is usually signed by whichever executive “owns” organizational policies. This could be the chief information officer (CIO), chief operations officer (COO), or chief human resources officer (CHRO). It is supported by an incident response plan that is documented and tested before an incident takes place. (More on this plan later.) The IMP should be developed with inputs from all stakeholders, not just the security department. Everyone needs to work together to make sure the policy covers all business, legal, regulatory, and security (and any other relevant) issues.

The IMP should be clear and concise. For example, it should indicate whether systems can be taken offline to try to save evidence or must continue functioning at the risk of destroying evidence. Each system and functionality should have a priority assigned to it. For instance, if a file server is infected, it should be removed from the network, but not shut down. However, if the mail server is infected, it should not be removed from the network or shut down, because of the priority the organization attributes to the mail server over the file server. Tradeoffs and decisions such as these have to be made when formulating the IMP, but it is better to think through these issues before the situation occurs, because better logic is usually possible before a crisis, when there’s less emotion and chaos.

Incident Management

Incident management includes proactive and reactive processes. Proactive measures need to be put into place so that incidents can be prevented or, failing that, detected quickly. Reactive measures need to be put into place so that detected incidents are dealt with properly.

Most organizations have only reactive management processes, which walk through how an incident should be handled. A more holistic approach is an incident management program that includes both proactive and reactive incident management processes, ensuring that triggers are monitored to make sure all incidents are actually uncovered. This commonly involves log aggregation, a security information and event management (SIEM) system, and user education. Having clear ways of dealing with incidents is not necessarily useful if you don't have a way to find out if incidents are indeed taking place.

All organizations should develop an *incident response team*, as mandated by the incident management policy, to respond to the large array of possible security incidents. The purpose of having an incident response (IR) team is to ensure that the organization has a designated group of people who are properly skilled, who follow a standard set of procedures, and who jump into action when a security incident takes place. The team should have proper reporting procedures established, be prompt in their reaction, work in coordination with law enforcement, and be recognized (and funded) by management as an important element of the overall security program. The team should consist of representatives from various business units, such as the legal department, HR, executive management, the communications department, physical/corporate security, IS security, and information technology.

There are three different types of incident response teams that an organization can choose to put into place. A *virtual* team is made up of experts who have other duties and assignments within the organization. It is called "virtual" because its members are not full-time incident responders but instead are called in as needed and may be physically remote. This type of team introduces a slower response time, and members must neglect their regular duties should an incident occur. However, a *permanent* team of folks who are dedicated strictly to incident response can be cost prohibitive to smaller organizations. The third type is a *hybrid* of the virtual and permanent models. Certain core members are permanently assigned to the team, whereas others are called in as needed.

Regardless of the type, the incident response team should have the following basic items available:

- A list of outside agencies and resources to contact or report to.
- An outline of roles and responsibilities.
- A call tree to contact these roles and outside entities.
- A list of computer or forensic experts to contact.
- A list of steps to take to secure and preserve evidence.

- A list of items that should be included in a report for management and potentially the courts.
- A description of how the different systems should be treated in this type of situation. (For example, remove the systems from both the Internet and the network and power them down.)

When a suspected crime is reported, the incident response team should follow a set of predetermined steps to ensure uniformity in their approach and that no steps are skipped. First, the IR team should investigate the report and determine whether an actual crime has been committed. If the team determines that a crime has been committed, they should inform senior management immediately. If the suspect is an employee, the team should contact a human resources representative right away. The sooner the IR team begins documenting events, the better. If someone is able to document the starting time of the crime, along with the employees and resources involved, that provides a good foundation for evidence. At this point, the organization must decide if it wants to conduct its own forensic investigation or call in experts. If experts are going to be called in, the system that was attacked should be left alone in order to try and preserve as much evidence of the attack as possible. If the organization decides to conduct its own forensic investigation, it must deal with many issues and address tricky elements. (Forensics will be discussed later in this chapter.)

Computer networks and business processes face many types of threats, each requiring a specialized type of recovery. However, an incident response team should draft and enforce a basic outline of how *all* incidents are to be handled. This is a much better approach than the way many organizations deal with these threats, which is usually in an ad hoc, reactive, and confusing manner. A clearly defined incident-handling process is more cost-effective, enables recovery to happen more quickly, and provides a uniform approach with certain expectation of its results.

Incident handling should be closely related to disaster recovery planning (covered in Chapter 23) and should be part of the organization's disaster recovery plan, usually as an appendix. Both are intended to react to some type of incident that requires a quick response so that the organization can return to normal operations. Incident handling is a recovery plan that responds to malicious technical threats. The primary goal of incident handling is to contain and mitigate any damage caused by an incident and to prevent any further damage. This is commonly done by detecting a problem, determining its cause, resolving the problem, and documenting the entire process.

Without an effective incident-handling program, individuals who have the best intentions can sometimes make the situation worse by damaging evidence, damaging systems, or spreading malicious code. Many times, the attacker booby-traps the compromised system to erase specific critical files if a user does something as simple as list the files in a directory. A compromised system can no longer be trusted because the internal commands listed in the path could be altered to perform unexpected activities. The system could now have a back door for the attacker to enter when he wants, or could have a logic bomb silently waiting for a user to start snooping around, only to destroy any and all evidence.

Incident handling should also be closely linked to the organization's security training and awareness program to ensure that these types of mishaps do not take place. Past issues that the incident response team encountered can be used in future training sessions to help others learn what the organization is faced with and how to improve response processes.

Employees need to know how to report an incident. Therefore, the incident management policy should detail an escalation process so that employees understand when evidence of a crime should be reported to higher management, outside agencies, or law enforcement. The process must be centralized, easy to accomplish (or the employees won't bother), convenient, and welcomed. Some employees feel reluctant to report incidents because they are afraid they will get pulled into something they do not want to be involved with or accused of something they did not do. There is nothing like trying to do the right thing and getting hit with a big stick. Employees should feel comfortable about the process, and not feel intimidated by reporting suspicious activities.

The incident management policy should also dictate how employees should interact with external entities, such as the media, government, and law enforcement. This, in particular, is a complicated issue influenced by jurisdiction, the status and nature of the crime, and the nature of the evidence. Jurisdiction alone, for example, depends on the country, state, or federal agency that has control. Given the sensitive nature of public disclosure, communications should be handled by communications, human resources, or other appropriately trained individuals who are authorized to publicly discuss incidents. Public disclosure of a security incident can lead to two possible outcomes. If not handled correctly, it can compound the negative impact of an incident. For example, given today's information-driven society, denial and "no comment" may result in a backlash. On the other hand, if public disclosure is handled well, it can provide the organization with an opportunity to win back public trust. Some countries and jurisdictions either already have or are contemplating breach disclosure laws that require organizations to notify the public if a security breach involving personally identifiable information (PII) is even suspected. So, being open and forthright with third parties about security incidents often is beneficial to organizations.

A sound incident-handling program works with outside agencies and counterparts. The members of the team should be on the mailing list of the Computer Emergency Response Team (CERT) so they can keep up-to-date about new issues and can spot malicious events, hopefully before they get out of hand. CERT is a division of the Software Engineering Institute (SEI) that is responsible for monitoring and advising users and organizations about security preparation and security breaches.

NOTE Resources for CERT can be found at https://www.cert.org/incident-management/.

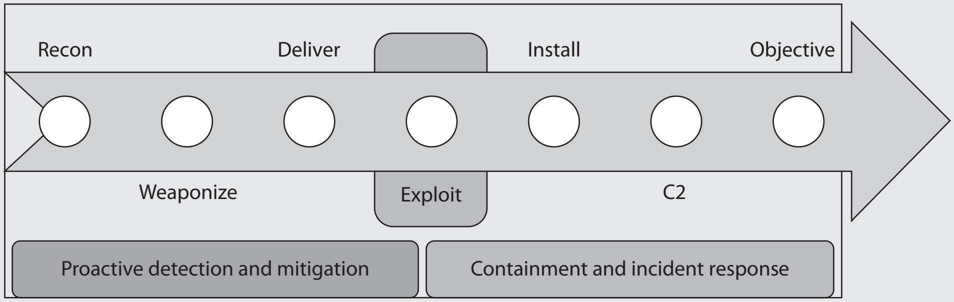
The Cyber Kill Chain

Even as we think about how best to manage incidents, it is helpful to consider a model that describes the stages attackers must complete to achieve their objectives. In their seminal 2011 white paper titled "Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains," Eric Hutchins, Michael Cloppert, and Rohan Amin (employees of Lockheed Martin Corporation, publisher of the white paper) describe a seven-stage intrusion model that has become an industry standard known as the Cyber Kill Chain framework. The seven stages are described here:

1. **Reconnaissance** The adversary has developed an interest in your organization as a target and begins a deliberate information-gathering effort to find vulnerabilities.
2. **Weaponization** Armed with detailed-enough information, the adversary determines the best way into your systems and begins preparing and testing the weapons to be used against you.
3. **Delivery** The cyber weapon is delivered into your system. In over 95 percent of the published cases, this delivery happens via e-mail.
4. **Exploitation** The malicious software is executing on a CPU within your network. This may have launched when the target user clicked a link, opened an attachment, visited a website, or plugged in a USB thumb drive. It could also (in somewhat rare cases) be the result of a remote exploit. One way or another, the attacker's software is now running in your systems.
5. **Installation** Most malicious software is delivered in stages. First, there is the exploit that compromised the system in the prior step. Then, some other software is installed in the target system to ensure persistence, ideally with a good measure of stealth.
6. **Command and Control (C2)** Once the first two stages of the software (exploit and persistence) have been executed, most malware will "phone home" to the attackers to let them know the attack was successful and to request updates and instructions.
7. **Actions on Objectives** Finally, the malware is ready to do whatever it was designed to do. Perhaps the intent is to steal intellectual property and send it to an overseas server. Or perhaps this particular effort is an early phase in a grander attack, so the malware will pivot off the compromised system. Whatever the case, the attacker has won at this point.

As you can probably imagine, the earlier in the kill chain we identify the attack, the greater our odds are of preventing the adversaries from achieving their objectives.

This is a critical concept in this model: if you can thwart the attack before stage four (exploitation), you stand a better chance of winning. Early detection, then, is the key to success.


Recon
Deliver
Install
Objective
Weaponize
Exploit
C2
Proactive detection and mitigation
Containment and incident response

Incident response is the component of incident management that is executed when a security incident takes place. It starts with detecting the incident and eventually leads to the application of lessons learned during the response. Let's take a closer look at each of the steps in the incident response process.

Detection

The first and most important step in responding to an incident is to realize that you have a problem in the first place. The organization's incident response plan should have specific criteria and a process by which the security staff declares that an incident has occurred. The challenge, of course, is to separate the wheat from the chaff and zero in on the alerts or other indicators that truly represent an immediate danger to the organization.

Detection boils down to having a good sensor network implemented throughout your environment. There are three types of sensors: technical, human, and third-party. *Technical sensors* are, perhaps, the type most of us are used to dealing with. They are provided by the previously mentioned SIEM systems and the other types of systems introduced in Chapter 21: detection and response (EDR), network detection and response (NDR), and security orchestration, automation, and response (SOAR). *Human sensors* can be just as valuable if everyone in your organization has the security awareness to notice odd events and promptly report them to the right place. Many organizations use a special e-mail address to which anyone can send an e-mail report. *Third-party sensors* (technical or human) exist in other organizations. For example, maybe you have a really good relationship with your supply chain partners, and they will alert you to incidents in their environments that appear related to you. That third party could also be a government agency letting you know you've been hacked, which is never a good way to start your day, but is better than not knowing.

Despite this abundance of sensors, detecting incidents can be harder than it sounds, for a variety of reasons. First, sophisticated adversaries may use tools and techniques that you are unable to detect (at least at first). Even if the tools or techniques are known to you, they may very well be hiding under a mound of false positives in your SIEM system. In some (improperly tuned) systems, the ratio of false positives to true positives can be ten to one (or higher). This underscores the importance of tuning your sensors and analysis platforms to reduce the rate of false positives as much as possible.

Response

Having detected the incident, the next step is to respond by containing the damage that has been or is about to be done to your most critical assets. The goal of containment during the response phase is to prevent or reduce any further damage from this incident so that you can begin to mitigate and recover. Done properly, mitigation buys the IR team time for a proper investigation and determination of the incident's root cause. The response strategy should be based on the category of the attack (e.g., internal or external), the assets affected by the incident, and the criticality of those assets. So, what kind of mitigation strategy is best? Well, it depends.

When complete isolation or containment is not a viable solution, you may opt to use boundary devices to stop one system from infecting another. This involves temporarily changing firewall/filtering router rule configuration. Access control lists can be applied to minimize exposure. These response strategies indicate to the attacker that his attack has been noticed and countermeasures are being implemented. But what if, in order to perform a root cause analysis, you need to keep the affected system online and not let on that you've noticed the attack? In this situation, you might consider installing a honeynet or honeypot to provide an area that will contain the attacker but pose minimal risk to the organization. This decision should involve legal counsel and upper management because honeynets and honeypots can introduce liability issues, as discussed in Chapter 21. Once the incident has been contained, you need to figure out what just happened by putting the available pieces together.

This is the substage of analysis, where more data is gathered (audit logs, video captures, human accounts of activities, system activities) to try and figure out the root cause of the incident. The goals are to figure out who did this, how they did it, when they did it, and why. Management must be continually kept abreast of these activities because they will be making the big decisions on how this situation is to be handled.

EXAM TIP Watch out for the context in which the term "response" is used. It can refer to the entire seven-phase incident management process or to the second phase of it. In the second usage, you can think of it as *initial* response aimed at containment.

Mitigation

Having "stopped the bleeding" with the initial containment response, the next step is to determine how to properly mitigate the threat. Though the instinctive reaction may be to clean up the infected workstation or add rules to your firewalls and IDS/IPS, this well-intentioned response could lead you on an endless game of whack-a-mole or, worse yet, blind you to the adversary's real objective. What do you know about the adversary? Who is it? What are they after? Is this tool and its use consistent with what you have already seen? Part of the mitigation stage is to figure out what information you need in order to restore security.

Once you have a hypothesis about the adversary's goals and plans, you can test it. If this particular actor is usually interested in PII on your high-net-worth clients but the incident you detected was on a (seemingly unrelated) host in the warehouse, was that an initial entry or pivot point? If so, then you may have caught the attacker before they worked their way further along the kill chain. But what if you got your attribution wrong? How could you test for that? This chain of questions, combined with quantifiable answers from your systems, forms the basis for an effective response. To quote the famous hockey player Wayne Gretzky, we should all "skate to where the puck is going to be, not where it has been."

NOTE It really takes a fairly mature threat intelligence capability to determine who is behind an attack (attribution), what are their typical tactics, techniques, and procedures (TTPs), and what might be their ultimate objective. If you do not have this capability, you may have no choice but to respond only to what you're detecting, without regard for what the adversary may actually be trying to do.

Once you are comfortable with your understanding of the facts of the incident, you move to eradicate the adversary from the affected systems. It is important to gather evidence before you recover systems and information. The reason is that, in many cases, you won't know that you will need legally admissible evidence until days, weeks, or even months after an incident. It pays, then, to treat each incident as if it will eventually end up in a court of justice.

Once all relevant evidence is captured, you can begin to fix all that was broken. The mitigation phase ends when you have affected systems that, while still isolated from the production networks, are free from adversarial control. For hosts that were compromised, the best practice is to simply reinstall the system from a gold master image and then restore data from the most recent backup that occurred prior to the attack. You may also have to roll back transactions and restore databases from backup systems. Once you are done, it is as if the incident never happened. Well, almost.

CAUTION An attacked or infected system should never be trusted, because you do not necessarily know all the changes that have taken place and the true extent of the damage. Some malicious code could still be hiding somewhere. Systems should be rebuilt to ensure that all of the potential bad mojo has been released by carrying out a proper exorcism.

Reporting

Though we discuss reporting at this point in order to remain consistent with the incident response process that (ISC)^2 identifies, incident reporting and documentation occurs at various stages in the response process. In many cases involving sophisticated attackers, the IR team first learns of the incident because someone else reports it. Whether it is an internal user, an external client or partner, or even a government entity, this initial report becomes the starting point of the entire process. In more mundane cases, we become aware that something is amiss thanks to a vigilant member of the security staff or one of the sensors deployed to detect attacks. However we learn of the incident, this first report starts what should be a continuous process of documentation.

According to NIST Special Publication 800-61, Revision 2, *Computer Security Incident Handling Guide*, the following information should be reported for each incident:

- Summary of the incident
- Indicators
- Related incidents
- Actions taken
- Chain of custody for all evidence (if applicable)
- Impact assessment
- Identity and comments of incident handlers
- Next steps to be taken

Recovery

Once the incident is mitigated, you must turn your attention to the recovery phase, in which the aim is to restore full, trustworthy functionality to the organization. It is one thing to restore an individual affected device, which is what we do in mitigation, and another to restore the functionality of business processes, which is the goal of recovery. For example, suppose you have a web service that provides business-to-business (B2B) logistic processes for your organization and your partner organizations. The incident to which you're responding affected the database and, after several hours of work, you mitigated that system and are ready to put it back online. In this recovery stage, you would certify the system as trustworthy and then integrate it back into the web service, thus restoring the business capability.

It is important to note that the recovery phase is characterized by significant testing to ensure the following:

- The affected system is really trustworthy
- The affected system is properly configured to support whatever business processes it did previously
- No compromises exist in those processes

The third characteristic of this phase is assured by close monitoring of all related systems to ensure that the compromise did not persist. Doing this during off-peak hours helps ensure that, should we discover anything else malicious, the impact to the organization is reduced.

Remediation

It is not enough to put the pieces of Humpty Dumpty back together again. You also need to ensure that the attack is never again successful. In the remediation phase, which can (and should) run concurrently with the other phases, you decide which security controls (e.g., updates, configuration changes, firewall/IDS/IPS rules) need to be put in place or modified. There are two steps to this. First, you may have controls that are hastily put into effect because, even if they cause some other issues, their immediate benefit outweighs the risks. Later on, you should revisit those controls and decide which should be made permanent (i.e., through your change management process) and what others you may want to put in place.

NOTE For best results, the remediation phase should start right after detection and be conducted in parallel with the other phases.

Another aspect of remediation is the identification of indicators of attack (IOAs) that can be used in the future to detect this attack in real time (i.e., as it is happening) as well as indicators of compromise (IOCs), which tell you when an attack has been successful and your security has been compromised. Typical indicators of both attack and compromise include the following:

- Outbound traffic to a particular IP address or domain name
- Abnormal DNS query patterns
- Unusually large HTTP requests and/or responses
- DDoS traffic
- New registry entries (in Windows systems)

At the conclusion of the remediation phase, you have a high degree of confidence that this particular attack will never again be successful against your organization. Ideally, you should incorporate your IOAs and IOCs into the following lessons learned stage and share them with the community so that no other organization can be exploited in this manner. This kind of collaboration with partners (and even competitors) makes the adversary have to work harder.

EXAM TIP Mitigation, recovery, and remediation are conveniently arranged in alphabetical order. First you stop the threat, then you get back to business as usual, and then you ensure the threat is never again able to cause this incident.

Lessons Learned

Closure of an incident is determined by the nature or category of the incident, the desired incident response outcome (for example, business resumption or system restoration), and the team's success in determining the incident's source and root cause. Once you have determined that the incident is closed, it is a good idea to have a team briefing that includes all groups affected by the incident to answer the following questions:

- What happened?
- What did we learn?
- How can we do it better next time?

The team should review the incident and how it was handled and carry out a postmortem analysis. The information that comes out of this meeting should indicate what needs to go into the incident response process and documentation, with the goal of continuous improvement. Instituting a formal process for the briefing provides the team with the ability to start collecting data that can be used to track its performance metrics.

Incident Response Planning

Incident management is implemented through two documents: the incident management policy (IMP) and the incident response plan (IRP). As discussed in the previous section, the IMP establishes authorities and responsibilities across the entire organization. The IMP identifies the IR lead for the organization and describes what every staff member is required to do with regard to incidents. For example, the IMP describes how employees are to report suspected incidents, to whom the report should be directed, and how quickly it should be done.

The IRP gets into the details of what should be done when responding to suspected incidents. The key sections of the IRP cover roles and responsibilities, incident classification, notifications, and operational tasks, all of which are described in the sections that follow. Normally, the IRP does not include detailed procedures for responding to specific incidents (e.g., phishing, data leak, ransomware), but establishes the framework within which all incidents will be addressed. Specific procedures are usually documented in *runbooks*, which are step-by-step scripts developed to deal with incidents that are either common enough or damaging enough to require this level of detailed documentation. Runbooks are described after the IRP sections.

Roles and Responsibilities

The group of individuals who make up the incident response team must have a variety of skills. They must also have a solid understanding of the systems affected by the incident, the system and application vulnerabilities, and the network and system configurations. Although formal education is important, real-world applied experience combined with proper training is key for these folks.

Many organizations divide their IR teams into two sub-teams. The first is the core team of incident responders, who come from the IT and security departments. These individuals are technologists who handle the routine incidents like restoring a workstation whose user inadvertently clicked the wrong link and caused self-infected damage. The second, or extended, team consists of individuals in other departments who are activated for more complex incidents. The extended team includes attorneys, public relations specialists, and human resources staff (to name a few). The exact makeup of this extended team will vary based on the specifics of the incident, but the point is that these are individuals whose day-to-day duties don't involve IT or security, and yet they are essential to a good response. Table 22-1 shows some examples of the roles and responsibilities in these two teams.

Role	Responsibilities
Core IR Team	
Chief information security officer (CISO)	• Develops and maintains the IR plan • Communicates with senior organizational leadership • Directs security controls before and after incidents
Director of security operations	• Directs execution of the IR plan • Communicates with applicable law enforcement agencies • Declares security incidents
IR team lead	• Overall responsibility for the IR plan • Communicates with senior organizational leadership • Maintains repository of incident response lessons learned
Cybersecurity analyst	• Monitors and analyzes security events • Nominates events for escalation to security incidents • Performs additional analyses for IR team lead as required
IT support specialist	• Manages security platforms • Implements mitigation, recovery, and remediation measures as directed by the IR team lead
Threat intelligence analyst	• Provides intelligence products related to incidents • Maintains repository of incident facts to support future intelligence products
Extended IR Team	
Human resources manager	• Provides oversight for incident-related human resource requirements (e.g., employee relations, labor agreements)
Legal counsel	• Provides oversight for incident-related legal requirements (e.g., liability issues, requirement for law enforcement reporting/coordination) • Ensures evidence collected maintains its forensic value in the event the organization chooses to take legal action
Public relations	• Ensures communications during an incident protect the confidentiality of sensitive information • Prepares communications to stockholders and the press
Business unit lead	• Balances IR actions and business requirements • Ensures business unit support to the IR team

Table 22-1 IR Team Roles and Responsibilities

In addition to these two teams, most organizations rely on third parties when the requirements of the incident response exceed the organic capabilities of the organization. Unless you have an exceptionally well-resourced internal IR team, odds are that you'll need help at some point. The best course of action is to enter into an IR services agreement with a reputable provider *before* any incidents happen. By taking care of the contract and nondisclosure agreement (NDA) beforehand, the IR service provider will be able to jump right into action when time is of the essence. Another time-saving measure is to coordinate a familiarization visit with your IR provider. This will allow the folks who may one day come to your aid to become familiar with your organization, infrastructure, policies, and procedures. They will also get a chance to meet your staff, so everyone learns everyone else's capabilities and limitations.

Incident Classification

The IR team should have a way to quickly determine whether the response to an incident requires that everyone be activated 24/7 or the response can take place during regular business hours over the next couple of days. There is, obviously, a lot of middle ground between these two approaches, but the point is that incident classification criteria should be established, understood by the whole team, and periodically reviewed to ensure that it remains relevant and effective.

There is no one-size-fits-all approach to developing an incident classification framework, but regardless of how you go about it, you should consider three incident dimensions:

- **Impact** If you have a risk management program in place, classifying an incident according to impact should be pretty simple since you've already determined the losses as part of your risk calculations. All you have to do is establish the thresholds that differentiate a bad day from a terrible one.
- **Urgency** The urgency dimension speaks to how quickly the incident needs to be mitigated. For example, an ongoing exfiltration of sensitive data needs to be dealt with immediately, whereas a scenario where a user caused self-infected damage with a bitcoin mining browser extension shouldn't require IR team members to get out of bed in the middle of the night.
- **Type** This dimension helps the team identify the resources that need to be notified and mobilized to deal with the incident. The team that handles the data exfiltration incident mentioned earlier is probably going to be different than the one that handles the infected browser.

Not all organizations explicitly call out each of these dimensions (and some organizations have more dimensions), but it is important to at least consider them. The simplest approach to incident classification simply uses severity and assigns various levels to this parameter depending on whether certain conditions are met. Table 22-2 shows a simple classification matrix for a small to medium-sized organization.

Severity	Criteria	Initial Response Time
Severity 1 (critical)	• Confirmed incident compromising mission-critical systems • Active exfiltration, alteration, or destruction of sensitive data • Incident requiring notification to government regulators • Life-threatening ongoing physical situation (e.g., suspicious package on site, unauthorized/hostile person, credible threat)	1 hour
Severity 2 (high)	• Confirmed incident compromising systems that are not mission-critical • Active exfiltration of non-sensitive data • Time-sensitive investigation of employees • Non-life-threatening but serious, ongoing physical situation (e.g., unauthorized person, theft of property)	4 hours
Severity 3 (moderate)	• Possible incident affecting any systems • Security policy violations • Long-term employee investigations requiring extensive collection and analysis • Non-life-threatening past physical situation (e.g., sensitive area left unsecured overnight)	48 hours

Table 22-2 Sample Incident Classification Matrix

The main advantage of formally classifying incidents is that it allows the preauthorized commitment of resources within specific timeframes. For example, if one of your SOC tier 2 analysts declares a severity 1 (critical) incident, she could be authorized to call the external IR service provider, committing the organization to pay the corresponding fees. There would be no need to get a hold of the CISO and get permission.

Notifications

Another benefit of classifying incidents is that it lets the IR team know who they need to inform and how frequently. Obviously, we don't want to call the CISO at home whenever an employee violates a security policy. On the other hand, we really don't want the CEO to find out the organization had an incident from reading the morning news. Keeping the right decision-makers informed at the right cadence enables everyone to do their jobs well, engenders trust, and leads to unified external messaging.

Table 22-3 shows an example notification matrix that builds on the classification shown previously in Table 22-2.

Notifications to external parties such as customers, partners, government regulators, and the press should be handled by communications professionals and not by the cybersecurity staff. The technical members of the IR team provide the facts to these communicators, who then craft messages (in coordination with the legal and marketing teams) that do not make things worse for the organization either legally or reputationally. Properly handled, IR communications can help improve trust and loyalty to the

Stakeholder	Severity Level	Notification
Executive leaders	S1	Immediate via e-mail and phone
	S2	On the next daily operational report
	S3	None
CISO	S1	Immediate via e-mail and phone
	S2	Within 4 hours via e-mail and phone
	S3	On the next daily operational report
Affected business units	S1	Immediate via e-mail and phone
	S2	Within 4 hours via e-mail
	S3	On the next daily operational report
Affected customers/partners	S1	Within 8 hours via e-mail
	S2	Within 72 hours via e-mail
	S3	None

Table 22-3 Sample Incident Notification Matrix

organization. Improperly handled, however, these notifications (or the lack thereof) can ruin (and have ruined) organizations.

Operational Tasks

Keeping stakeholders informed is just one of the many tasks involved in incident response. Just like any other complex endeavor, we should leverage structured approaches to ensure that all required tasks are performed, and that they are done consistently and in the right order. Now, of course, different types of incidents require different procedures. Responding to a ransomware attack requires different procedures than the procedures for responding to a malicious insider trying to steal company secrets. Still, all incidents follow a very similar pattern at a high level. We already saw this in the discussion of the seven phases in the incident management process that you need to know for the CISSP exam, which apply to all incidents.

Many organizations deal with the need for completeness and consistency in IR by spelling out operational tasks in the IRP, sometimes with a field next to each task to indicate when the task was completed. The IR team lead can then just walk down this list to ensure the right things are being done in the right order. Table 22-4 shows a sample operational tasks checklist.

Table 22-4 is not meant to be all-inclusive but it does capture the most common tasks that apply to every IR in most organizations. As mentioned earlier, different types of incidents require different approaches. While the task list should be general enough to accommodate these specialized procedures, we also want to keep it specific enough to serve as an overall execution plan.

Operational Task	Date/Time Completed
Pre-Execution	
Identify assets affected	
Obtain access (physical and logical) to all affected assets	
Determine forensic evidence requirements	
Review compliance requirements (e.g., GDPR, HIPAA, PCI DSS)	
Initiate communications plan	
Response	
Perform immediate actions to mitigate the impact of the incident	
Validate detection mechanisms	
Request relevant intelligence from threat intelligence team	
Gather and preserve incident-related data (e.g., PCAP, log files)	
Develop an initial timeline of incident-related activity	
Develop mitigation plan based on initial assessment	
Mitigation	
Verify availability of backup/redundant system (if mission-critical system was compromised)	
Activate backup/redundant systems for continuity of operations (if mission-critical system was compromised)	
Isolate affected assets	
Collect forensic evidence from compromised systems (if applicable)	
Remove active threat mechanisms to limit further activity	
Initiate focused monitoring of the environment for additional activity	
Recovery	
Restore affected systems' known-good backups or gold masters	
Validate additional controls on restored systems prevent reoccurrence	
Reconnect restored systems to production networks	
Verify no additional threat activity exists on restored systems	
Remediation	
Finalize root cause, threat mechanisms, and incident timeline	
Identify IOCs and IOAs	
Initiate change management processes to prevent reoccurrence	
Implement preventive and detective controls to prevent reoccurrence	

Table 22-4 Sample Operational Tasks List

Runbooks

When we need specialized procedures, particularly when we expect a certain type of incident to happen more than once, we want to document those procedures to ensure we don't keep reinventing the wheel every time a threat actor gets into our systems. A *runbook* is a collection of procedures that the IR team will follow for specific types of incidents. Think of a runbook as a cookbook. If you feel like having a bean casserole for dinner, you open your cookbook and look up that recipe. It'll tell you what ingredients you need and what the step-by-step procedure is to make it. Similarly, a runbook has tabs for the most likely and/or most dangerous incidents you may encounter. Once the incident is declared by the SOC (or whoever is authorized to declare an incident has occurred), the IR team lead opens the runbook and looks up the type of incident that was declared. The runbook specifies what resources are needed (e.g., specific roles and tools) and how to apply them.

When developing runbooks, you have to be careful that the documentation doesn't take more time and resources to develop than you would end up investing in responding to that incident type. As with any other control, the cost of a runbook cannot exceed the cost of doing nothing (and figuring things out on the fly). For that reason, most organizations focus their runbooks on incidents that require complex responses and those that are particularly sensitive. Other incidents can be (and usually are) added to the runbook, but those additions are deliberate decisions of the SOC manager based on the needs of the organization. For example, if an organization experiences high turnover rates, it might be helpful for new staff to have a more comprehensive runbook to which they can turn.

Another aspect to consider is that runbooks are only good if they are correct, complete, and up to date. Even if you do a great job when you first write runbooks, you'll have to invest time periodically in keeping them updated. For best results, incorporate runbooks into your change management program so that, whenever an organizational change is made, the change advisory board (CAB) asks the question: does this require an update to the IR runbooks?

Investigations

Whatever type of security incident we're facing, we should treat the systems and facilities that it affects as potential crime scenes. The reason is that what may at first appear to have been a hardware failure, a software defect, or an accidental fire may have in fact been caused by a malicious actor targeting the organization. Even acts of nature like storms or earthquakes may provide opportunities for adversaries to victimize us. Because we are never (initially) quite sure whether an incident may have a criminal element, we should treat all incidents as if they do (until proven otherwise).

Since computer crimes are only increasing and will never really go away, it is important that all security professionals understand how computer investigations should be carried out. This includes understanding legal requirements for specific situations, the chain of custody for evidence, what type of evidence is admissible in court, incident response procedures, and escalation processes.

Cops or No Cops?

Management needs to make the decision as to whether law enforcement should be called during an incident response. The following are some of the issues to understand if law enforcement is brought in:

- You may not have a choice in certain cases (e.g., cases involving national security, child pornography, etc.).
- Law enforcement agencies bring significant investigative capability.
- The organization may lose control over where the investigation leads once law enforcement is involved.
- Secrecy of compromise is not promised; it could become part of public record.
- Evidence will be collected and may not be available for a long period of time.

Successfully prosecuting a crime requires solid evidence. Computer forensics is the art of retrieving this evidence and preserving it in the proper ways to make it admissible in court. Without proper computer forensics, few computer crimes could ever be properly and successfully presented in court. The most common reasons evidence is deemed inadmissible in court are lack of qualified staff handling it, lack of established procedures, poorly written policy, or a broken chain of custody.

When a potential computer crime takes place, it is critical that the investigation steps are carried out properly to ensure that the evidence will be admissible to the court (if the matter goes that far) and can stand up under the cross-examination and scrutiny that will take place. As a security professional, you should understand that an investigation is not just about potential evidence on a disk drive. The context matters during an investigation, including the people, network, connected internal and external systems, applicable laws and regulations, management's stance on how the investigation is to be carried out, and the skill set of whoever is carrying out the investigation. Messing up just one of these components could make your case inadmissible or at least damage it if it is brought to court.

Motive, Opportunity, and Means

Today's computer criminals are similar to their traditional counterparts. To understand the "why" in crime, it is necessary to understand the motive, opportunity, and means—or MOM. This is the same strategy used to determine the suspects in a traditional, noncomputer crime.

Motive is the "who" and "why" of a crime. The motive may be induced by either internal or external conditions. A person may be driven by the excitement, challenge, and adrenaline rush of committing a crime, which would be an internal condition. Examples of external conditions might include financial trouble, a sick family member, or other dire straits. Understanding the motive for a crime is an important piece in figuring out who would engage in such an activity. For example, financially motivated attackers such as those behind ransomware want to get your money. In the case of ransomware purveyors, they realize that if they don't decrypt a victim's data after payment of the ransom, the word will get out and no other victims will pay the ransom. For this reason, most modern ransomware actors reliably turn over decryption keys upon payment. Some ransomware gangs even go the extra mile and set up customer service operations to help victims with payment and decryption issues.

Opportunity is the "where" and "when" of a crime. Opportunities usually arise when certain vulnerabilities or weaknesses are present. If an organization does not regularly patch systems (particularly public-facing ones), attackers have all types of opportunities within that network. If an organization does not perform access control, auditing, and supervision, employees may have many opportunities to embezzle funds and defraud the organization. Once a crime fighter finds out why a person would want to commit a crime (motive), she will look at what could allow the criminal to be successful (opportunity).

Means pertains to the abilities a criminal would need to be successful. Suppose a crime fighter was asked to investigate a case of fraud facilitated by a subtle but complex modification made to a software system within a financial institution. If the suspects were three people and two of them just had general computer knowledge, but the third one was a programmer and system analyst, the crime fighter would realize that this person is much likelier to have the means to commit this crime than the other two individuals.

Computer Criminal Behavior

Like traditional criminals, computer criminals have a specific *modus operandi* (*MO*, pronounced "em-oh"). In other words, each criminal typically uses a distinct method of operation to carry out their crime, and that method can be used to help identify them. The difference with computer crimes is that the investigator, obviously, must have knowledge of technology. For example, the MO of a particular computer criminal may include the use of specific tools or targeting specific systems or networks. The method usually involves repetitive signature behaviors, such as sending e-mail messages or programming syntax. Knowledge of the criminal's MO and signature behaviors can be useful throughout the investigative process. Law enforcement can use the information to identify other offenses by the same criminal, for example. The MO and signature behaviors can also provide information that is useful during interviews (conducted by authorized staff members or law enforcement agencies) and potentially a trial.

Psychological crime scene analysis (profiling) can also be conducted using the criminal's MO and signature behaviors. Profiling provides insight into the thought processes of the attacker and can be used to identify the attacker or, at the very least, the tool he used to conduct the crime.

Evidence Collection and Handling

Good evidence is the bedrock on which any sound investigation is built. When dealing with any incident that might end up in court, digital evidence must be handled in a careful fashion so that it can be admissible no matter what jurisdiction is prosecuting a defendant. Within the United States, the *Scientific Working Group on Digital Evidence (SWGDE)* aims to ensure consistency across the forensic community. The principles developed by SWGDE for the standardized recovery of computer-based evidence are governed by the following attributes:

- Consistency with all legal systems
- Allowance for the use of a common language
- Durability
- Ability to cross international and state boundaries
- Ability to instill confidence in the integrity of evidence
- Applicability to all forensic evidence
- Applicability at every level, including that of individual, agency, and country

The international standard on digital evidence handling is ISO/IEC 27037: *Guidelines for Identification, Collection, Acquisition, and Preservation of Digital Evidence*. This document identifies four phases of digital evidence handling, which are identification, collection, acquisition, and preservation. Let's take a closer look at each.

NOTE You must ensure that you have the legal authority to search for and seize digital evidence before you do so. If in doubt, consult your legal counsel.

Identification

The first phase of digital evidence handling is to identify the digital crime scene. Rarely does only one device comprise the scene of the crime. More often than not, digital evidence exists on a multitude of other devices such as routers, network appliances, cloud services infrastructure, smartphones, and even IoT devices. Whether or not you have to secure a court order to seize evidence, you want to be very deliberate about determining what you think you need to collect and where it might exist.

When you arrive at the crime scene (whether it be physical or virtual), you want to carefully document everything you see and do. If you're dealing with a physical crime scene, photograph it from every possible angle before you touch anything. Label wires and cables and then snap a photo of the labeled system before it is disassembled. Remember that you want to instill confidence in the integrity of evidence and how it was handled from the very onset.

Identifying evidence items at a crime scene may not be straightforward. You could discover wireless networks that would allow someone to remotely tamper with the evidence. This would require you to consider ways to isolate the evidence from radio frequency (RF) signals in order to control the crime scene. There may also be evidence in devices (e.g., thumb drives) that are hidden either deliberately or unintentionally. Law enforcement agents sometimes resort to using specially trained dogs that can sniff out