Chapter 17
Preventing and Responding to Incidents

THE CISSP TOPICS COVERED IN THIS CHAPTER INCLUDE:

✓ **Domain 7.0: Security Operations**

- 7.2 Conduct logging and monitoring activities
    - 7.2.1 Intrusion detection and prevention system (IDPS)
    - 7.2.2 Security information and event management (SIEM)
    - 7.2.3 Continuous monitoring and tuning
    - 7.2.4 Egress monitoring
    - 7.2.5 Log management
    - 7.2.6 Threat intelligence (e.g., threat feeds, threat hunting)
- 7.6 Conduct incident management
    - 7.6.1 Detection
    - 7.6.2 Response
    - 7.6.3 Mitigation
    - 7.6.4 Reporting
    - 7.6.5 Recovery
    - 7.6.6 Remediation
    - 7.6.7 Lessons learned
- 7.7 Operate and maintain detection and preventative measures
    - 7.7.2 Intrusion detection systems (IDS) and intrusion prevention systems (IPS)
    - 7.7.3 Whitelisting/blacklisting
    - 7.7.4 Third-party provided security services
    - 7.7.5 Sandboxing

- 7.7.6 Honeypots/honeynets
- 7.7.7 Anti-malware
- 7.7.8 Machine learning and artificial intelligence (AI) based tools

The Security Operations domain for the CISSP certification exam includes several objectives directly related to incident management. Effective incident management helps an organization respond when attacks occur to limit the scope of an attack. Organizations implement preventive measures to protect against and detect attacks, and this chapter covers many of these controls and countermeasures. Logging and monitoring provide assurances that security controls are in place and provide the desired protection.

Conducting Incident Management

One of the primary goals of any security program is to prevent security incidents. However, despite the best efforts of IT and security professionals, incidents occur. When they do, an organization must be able to respond to limit or contain the incident. The primary goal of incident management is to minimize the impact on the organization.

Defining an Incident

Before digging into incident management, it's important to understand the definition of an incident. Although that may seem simple, you'll find that different sources have slightly different definitions.

In general, an *incident* is any event that has a negative effect on the confidentiality, integrity, or availability of an organization's assets. Notice that this definition encompasses events as diverse as direct attacks, natural occurrences such as a hurricane or earthquake, and even accidents, such as someone unintentionally cutting cables for a live network.

In contrast, a *computer security incident* (sometimes called just a *security incident*) commonly refers to an incident that is the result of an attack or the result of malicious, intentional, or accidental actions on the part of users. For example, request for comments (RFC) 2350, Expectations for Computer Security Incident Response, defines both a security incident and a computer security incident as “any adverse event which compromises some aspect of computer or network security.”

NIST SP 800-61—Computer Security Incident Handling Guide, defines a computer security incident as “a violation or imminent threat of violation of computer security policies, acceptable use policies, or standard security practices.”

NIST documents, including SP 800-61, can be accessed from the NIST publications page: `http://csrc.nist.gov/Publications`.

In the context of incident management, an incident is referring to a computer security incident. However, you'll often see it listed as just an incident. For example, within the CISSP Security Operations domain, the “Conduct incident management” objective is clearly referring to computer security incidents.

> NOTE In this chapter, any reference to an incident is to a computer security incident. Organizations handle some incidents, such as weather events or natural disasters, using other methods, such as a business continuity plan (covered in Chapter 3, “Business Continuity Planning”) or a disaster recovery plan (covered in Chapter 18, “Disaster Recovery Planning”).

Organizations commonly define the meaning of a computer security incident within their security policy or incident management plans. The definition is usually one or two sentences long and includes examples of common events that the organization classifies as security incidents, such as the following:

- Any attempted network intrusion
- Any attempted denial-of-service attack
- Any detection of malicious software
- Any unauthorized access of data
- Any violation of security policies

Incident Management Steps

Effective incident management is handled in several steps or phases. Figure 17.1 shows the seven steps involved in incident management as outlined in the CISSP objectives. It's important to realize that incident management is an ongoing activity, and the results of the lessons learned stage are used to improve detection methods or help prevent a repeated incident. The following sections describe these steps in more depth.

NOTE You may run across documentation that lists these steps differently. For example, NIST SP 800-61, an excellent resource for learning more about incident handling, identifies the following four phases in the incident response life cycle: 1) preparation, 2) detection and analysis, 3) containment, eradication, and recovery, and 4) post-incident recovery. Still, no matter how documentation lists the steps, it contains many of the same elements and has the same goal of effectively managing incident response. The key point is that you can expect to see the steps shown in Figure 17.1 on the live CISSP exam.

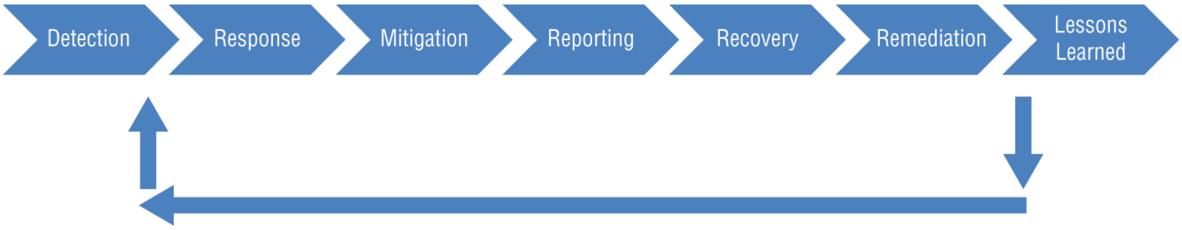


FIGURE 17.1 Incident management

It's important to stress that incident management does not include a counterattack against the attacker. Launching attacks on others is counterproductive and often illegal. If an employee can identify the attacker and launch an attack, it will likely result in an escalation of the attacker's actions. In other words, the attacker may now consider it personal and regularly launch grudge attacks. In addition, it's likely that the attacker is hiding behind one or more innocent victims. Attackers often use spoofing methods to hide their identity or launch attacks by bots in a botnet. Counterattacks may be against an innocent victim rather than an attacker.

Detection

IT environments include multiple methods of detecting potential incidents. The following list identifies many of the common methods used to detect potential incidents. It also includes notes on how these methods report the incidents:

- Intrusion detection and prevention systems (described later in this chapter) send alerts to administrators when they detect a potential incident.
- Anti-malware software will often display a pop-up window to indicate when it detects malware.
- Many automated tools regularly scan audit logs looking for predefined events, such as the use of special privileges. When they detect specific events, they typically send an alert to administrators.
- End users sometimes detect irregular activity and contact technicians or administrators for help. When users report events, such as the inability to access a network resource or

update a system, it alerts IT personnel about a potential incident.

Notice that just because an IT professional receives an alert from an automated tool or a user complaint, this doesn't always mean an incident has occurred. Intrusion detection and prevention systems often give false alarms, and end users are prone to simple user errors. IT personnel investigate these events to determine whether they are incidents.

Many IT professionals are classified as first responders for incidents. They are the first ones on the scene and know how to differentiate typical IT problems from security incidents. They are similar to medical first responders, who have outstanding skills and abilities to provide medical assistance at accident scenes and help get the patients to medical facilities when necessary. The medical first responders have specific training to help them determine the difference between minor and major injuries. Further, they know what to do when they come across a major injury. Similarly, IT professionals need specific training to determine the difference between a typical problem that needs troubleshooting and a security incident that they need to escalate.

After investigating an event and determining it is a security incident, IT personnel move to the next step: response. In many cases, the individual doing the initial investigation will escalate the incident to bring in other IT professionals to respond.

Response

After detecting and verifying an incident, the next step is response. The response varies depending on the severity of the incident. Many organizations have a designated incident response team—sometimes called a cyber incident response team (CIRT) or computer security incident response team (CSIRT). The organization activates the team during a major security incident but does not typically activate the team for minor incidents. A formal incident response plan documents who would activate the team and under what conditions.

Team members are trained on incident response and the organization's incident response plan. Typically, team members investigate the incident, contain and assess the damage, collect evidence, report the incident, and perform recovery procedures. They also participate in the remediation and lessons learned stages, and help with root cause analysis.

The more quickly an organization can respond to an incident, the better chance they have at limiting the damage. If an incident continues for hours or days, the damage is likely to be greater. For example, an attacker may be trying to access a customer database. A quick response can prevent the attacker from obtaining any meaningful data. However, if given continued unobstructed access to the database for several hours or days, the attacker may be able to get a copy of the entire database.

After an investigation is over, management may decide to prosecute responsible individuals. Because of this, it's important to protect all data as evidence during the investigation. Chapter 19, “Investigations and Ethics,” covers incident handling and response in the context of supporting investigations. If any possibility of prosecution exists, team members take extra steps to protect the evidence. This ensures that the evidence can be used in legal procedures.

> NOTE Computers should not be turned off when containing an incident. Temporary files and data in volatile random access memory (RAM) will be lost if the computer is powered down. Forensics experts have tools they can use to retrieve data in temporary files and volatile RAM as long as the system is kept powered on. However, this evidence is lost if someone turns the computer off or unplugs it.

Mitigation

Mitigation steps attempt to contain an incident. One of the primary goals of effective incident management is to limit the effect or scope of an incident. For example, if an infected computer is sending data out its network adapter, a technician can disable the network adapter or disconnect the cable to the computer. Sometimes containment involves disconnecting a network from other networks to contain the problem within a single network. When the problem is isolated, security personnel can address it without worrying about it spreading to the rest of the network.

In some cases, responders take steps to mitigate the incident, but without letting the attacker know that the attack has been detected. This allows security personnel to monitor the attacker's activities and determine the scope of the attack.

Reporting

Reporting refers to reporting an incident within the organization and to organizations and individuals outside the organization. Although there's no need to report a minor malware infection to a company's CEO, upper-level management does need to know about serious security breaches.

As an example, the medical debt collections firm R1 RCM was hit by a ransomware attack in August 2020. R1 RCM has partnered with over 750 healthcare companies, and they held personal data on millions of patients. This included Social Security numbers, medical diagnostic data, and financial data. The attack reportedly occurred about a week before the company was planning to release its quarterly financial reports. Although R1 RCM didn't provide internal communications details, you can bet someone notified the CEO soon after the attack was detected.

Organizations often have a legal requirement to report some incidents outside of the organization. Most countries (and many smaller jurisdictions, including states and cities) have enacted regulatory compliance laws to govern security breaches, particularly as they apply to sensitive data retained within information systems. These laws typically include a requirement to report the incident, especially if the security breach exposed customer data. Laws differ from locale to locale, but all seek to protect the privacy of individual records and information, to protect consumer identities, and to establish standards for financial practice and corporate governance. Every organization has a responsibility to know what laws apply to it and to abide by those laws.

Many jurisdictions have specific laws governing the protection of personally identifiable information (PII). If a data breach exposes PII, the organization must report it. Different laws have different reporting requirements, but most include a requirement to notify individuals affected by the incident. In other words, if an attack on a system resulted in an attacker gaining PII about you, the owners of the system have a responsibility to inform you of the attack and what data the attackers accessed.

In response to serious security incidents, the organization should consider reporting the incident to official agencies. In the United States, this may mean notifying the Federal Bureau of Investigation (FBI), district attorney offices, and state and local law enforcement agencies. In Europe, organizations may report the incident to the International Criminal Police Organization (INTERPOL) or some other entity based on the incident and their location. These agencies may assist in investigations, and the data they collect may help them prevent future attacks against other organizations.

> NOTE Organizations sometimes choose not to involve law enforcement to avoid negative publicity or an intrusive investigation. However, this is not an option if personal information is exposed. Additionally, some third-party standards, such as the Payment Card Industry Data Security Standard (PCI DSS), require organizations to report certain security incidents to law enforcement. Many incidents are not reported because they aren't recognized as incidents. This is often the result of inadequate training. The obvious solution is to ensure that personnel have relevant training. Training should teach individuals how to recognize incidents, what to do in the initial response, and how to report an incident.

Recovery

The next step is to recover the system or return it to a fully functioning state. This step can be very simple for minor incidents and may only require a reboot. However, a major incident may require completely rebuilding a system. Rebuilding the system includes restoring all data from the most recent backup.

When a compromised system is rebuilt from scratch, it's important to ensure it is configured properly and is at least as secure as it was before the incident. If an organization has effective configuration management and change management programs, these programs will provide the necessary documentation to ensure the rebuilt systems are configured properly. Things to double-check include the following:

- Access control lists (ACLs), which include firewall or router rules
- Services and protocols, ensuring that unneeded services and protocols are disabled or removed
- Patches, ensuring that all up-to-date patches are installed
- User accounts, ensuring that they have changed from their default configurations
- Compromises, ensuring that any known compromises have been reversed

> In some cases, an attacker may have installed malicious code on a system during an attack. This attack may not be apparent without a detailed inspection of the system. The most secure method of restoring a system after an incident is completely rebuilding the system from scratch. If investigators suspect that an attacker may have modified code on the system, rebuilding a system may be a good option.

Remediation

In the remediation stage, personnel look at the incident, identify what allowed it to occur, and then implement methods to prevent it from happening again. This step includes performing a root cause analysis.

A root cause analysis examines the incident to determine what allowed it to happen. For example, if attackers successfully accessed a database through a website, personnel would examine all the system elements to determine what allowed the attackers to succeed. If the root cause analysis identifies a vulnerability that can be mitigated, this stage will recommend a change.

It could be that the web server didn't have up-to-date patches, allowing the attackers to gain remote control of the server. Remediation steps might include implementing a patch management program. Perhaps the website application wasn't using adequate input validation techniques, allowing a successful SQL injection attack. Remediation would involve updating the application to include input validation. Maybe the database is located on the web server instead of in a backend database server. Remediation might include moving the database to a server behind an additional firewall.

Lessons Learned

During the lessons learned stage, personnel examine the incident and the response to see if there are any lessons to be learned. The incident response team will be involved in this stage, but other employees who are knowledgeable about the incident will also participate.

While examining the response to the incident, personnel look for any areas where they can improve their response. For example, if the response team took a long time to contain the incident, the examination tries to determine why. It might be that personnel don't have adequate training or the knowledge and expertise to respond effectively. They may not have recognized the incident when they received the first notification, allowing an attack to continue longer than necessary. First responders may not have recognized the need to protect evidence and inadvertently corrupted it during the response.

Remember, the output of this stage can be fed back to the detection stage of incident management. For example, administrators may realize that attacks are getting through undetected and increase their detection capabilities and recommend changes to their intrusion detection systems.

It is common for the incident response team to create a report when they complete a lessons learned review. Based on the findings, the team may recommend changes to procedures, the addition of security controls, or even changes to policies. Management will decide what recommendations to implement and is responsible for the remaining risk for any recommendations they reject.

Real World Scenario

Delegating Incident Management to Users

In one organization where one of the authors worked, the responsibility to respond to computer infections was extended to users. Close to each computer was a checklist that identified common symptoms of malware infection. If users suspected their computers were infected, the checklist instructed them to disable or disconnect the network adapter and contact the help desk to report the issue. By disabling or disconnecting the network adapter, they helped contain the malware to their system and stopped it from spreading any further.

This isn't possible in all organizations, but in this case, users were part of a very large network operations center, and they were all involved in some form of computer support. In other words, they weren't typical end users but instead had a substantial amount of technical expertise.

Implementing Detection and Preventive Measures

Ideally, an organization can avoid incidents completely by implementing preventive countermeasures. However, no matter how effective preventive countermeasures are, incidents will still happen. Other controls help detect incidents and respond to them.

Chapter 2, "Personnel Security and Risk Management Concepts," discusses controls in more depth. This section covers many of the specific controls designed to prevent and detect security incidents. As a reminder, the following list describes preventive and detection controls:

Preventive Control A *preventive control* attempts to thwart or stop unwanted or unauthorized activity from occurring. Examples of preventive controls are fences, locks, biometrics, separation of duties policies, job rotation policies, data classification, access control methods, encryption, smartcards, callback procedures, security policies, security awareness training, antivirus software, firewalls, and intrusion prevention systems.

Detection Control A *detection control* attempts to discover or detect unwanted or unauthorized activity. Detection controls operate after the fact and can discover the activity only after it has occurred. Examples of detection controls are security guards, motion detectors, recording and reviewing of events captured by security cameras or closed-circuit television (CCTV), job rotation policies, mandatory vacation policies, audit trails, honeypots or honeynets, intrusion detection systems, violation reports, supervision and reviews of users, and incident investigations.

NOTE You may notice the use of both *preventative* and *preventive*. Although most documentation currently uses only *preventive*, the CISSP objectives include both usages. For example, Domain 1 includes references to preventive controls. This chapter covers objectives from Domain 7, and Domain 7 refers to preventative measures. For simplicity, we are using preventive in this chapter, except when quoting the CISSP objectives.

Basic Preventive Measures

Although there is no single step you can take to protect against all attacks, you can take some basic steps that go a long way to protect against many types of attacks. Many of these steps are described in more depth in other areas of the book but are listed here as an introduction to this section.

Keep systems and applications up-to-date. Vendors regularly release patches to correct bugs and security flaws, but these only help when they're applied. Patch management (covered in Chapter 16, “Managing Security Operations”) ensures that systems and applications are kept up-to-date with relevant patches.

Remove or disable unneeded services and protocols. If a system doesn't need a service or protocol, it should not be running. Attackers cannot exploit a vulnerability in a service or protocol that isn't running on a system. As an extreme contrast, imagine a web server is running every available service and protocol. It is vulnerable to potential attacks on any of these services and protocols.

Use intrusion detection and prevention systems. Intrusion detection and prevention systems observe activity, attempt to detect attacks, and provide alerts. Intrusion prevention systems can often block or stop attacks. These systems are described in more depth later in this chapter.

Use up-to-date anti-malware software. Chapter 21, “Malicious Code and Application Attacks,” covers various types of malicious code such as viruses and worms. A primary countermeasure is anti-malware software, covered later in this chapter.

Use firewalls. Firewalls can prevent many different types of attacks. Network-based firewalls protect entire networks, and host-based firewalls protect individual systems. Chapter 11, “Secure Network Architecture and Components,” included information on using firewalls within a network, and this chapter includes a section describing how firewalls can prevent attacks.

Implement configuration and system management processes. Configuration and system management processes help ensure that systems are deployed in a secure manner and remain in a secure state throughout their lifetimes. Chapter 16 covers configuration and change management processes.

> NOTE Thwarting an attacker's attempts to breach your security requires vigilant efforts to keep systems patched and properly configured. Firewalls and intrusion detection and prevention systems often provide the means to detect and gather evidence to prosecute attackers that have breached your security.

Understanding Attacks

Security professionals need to be aware of common attack methods so that they can take proactive steps to prevent them, recognize them when they occur, and respond appropriately in response to an attack. This section provides an overview of many common attacks. The following sections discuss many of the preventive measures used to thwart these and other attacks.

> NOTE We've attempted to avoid duplication of specific attacks but also provide a comprehensive coverage of different types of attacks throughout this book. In addition to this chapter, you'll see different types of attacks in other chapters. For example, Chapter 7, “PKI and Cryptographic Applications,” covered some cryptographic attacks; Chapter 12, “Secure Communications and Network Attacks,” covered different types of network-based attacks; Chapter 14, “Controlling and Monitoring Access,” covered various access control attacks; and Chapter 21 covers various attacks related to malicious code and applications.

Botnets

Botnets are quite common today. The computers in a botnet are like robots (referred to as *bots* and sometimes *zombies*). Multiple bots in a network form a botnet and will do whatever attackers instruct them to do. A bot herder is typically a criminal who controls all the computers in the botnet via one or more command-and-control (C&C or C2) servers.

The bot herder enters commands on the server, and the bots check in with the command-and-control server to receive instructions. Bots can be programmed to contact the server periodically or remain dormant until a specific programmed date and time or in response to an event, such as when specific traffic is detected. Bot herders commonly instruct the bots within a botnet to launch a wide range of DDoS attacks, send spam and phishing emails, or rent the botnets out to other criminals.

Computers are typically joined to a botnet after being infected with some type of malicious code or malicious software. Once the computer is infected, it often gives the bot herder remote access to the system and additional malware is installed. In some cases, the bots install malware on the infected systems. These may search for files that include passwords or other information of interest to the attacker. The malware sometimes installs keyloggers to capture user keystrokes and send them back to the attacker. Bot herders often issue commands to the bots, causing them to launch attacks.

Botnets of more than 40,000 computers are relatively common, and botnets controlling millions of systems have been active in the past. Some bot herders control more than one botnet.

There are many methods of protecting systems from being joined to a botnet, so it's best to use a defense-in-depth strategy, implementing multiple layers of security. Because systems are typically joined to a botnet after becoming infected with malware, it's important to ensure that systems and networks are protected with up-to-date anti-malware software. Some malware takes advantage of unpatched flaws in operating systems and applications, so keeping a system up-to-date with patches helps keep it protected. However, attackers are increasingly creating new malware that bypasses the anti-malware software, at least temporarily. They are also discovering vulnerabilities that don't have patches available yet.

Educating users is extremely important as a countermeasure against botnet infections. Worldwide, attackers are almost constantly sending out malicious phishing emails. Some include malicious attachments that join systems to a botnet if the user opens them. Others include links to malicious sites that attempt to download malicious software or try to trick the user into downloading the malicious software. Others try to trick users into giving up their passwords, and attackers then use these harvested passwords to infiltrate systems and networks. Training users about these attacks and maintaining a high level of security awareness can often help prevent many attacks.

Many malware infections are browser-based, allowing user systems to become infected when the user is surfing the web. Keeping browsers and their plug-ins up-to-date is an important security practice. Additionally, most browsers have strong security built-in, and these features shouldn't be disabled. For example, most browsers support sandboxing (covered later in the “Sandboxing” section of this chapter) to isolate web applications, but some browsers include the ability to disable sandboxing.

Disabling sandboxing might improve the performance of the browser slightly, but the risk is significant.

Real World Scenario

Botnets, IoT, and Embedded Systems

Attackers have traditionally infected desktop and laptop computers with malware and joined them to botnets. Although this still occurs, attackers have been expanding their reach to the Internet of Things (IoT).

For example, attackers used the Mirai malware to launch a DDoS attack on DNS servers hosted by Dyn. Most of the devices involved in this attack were IoT devices such as internet-connected cameras, digital video recorders, and home-based routers that were infected and added to the Mirai botnet. The attack effectively prevented users from accessing many popular websites such as Twitter, Netflix, Amazon, Reddit, Spotify, and more. The research company Gartner estimates there are as many as 20 billion IoT devices in use in 2020, giving attackers many more targets.

Embedded systems include any device with a processor, an operating system, and one or more dedicated apps. Some examples include devices that control traffic lights, medical equipment, automatic teller machines (ATMs), printers, thermostats, digital watches, and digital cameras. Many automobiles include multiple embedded systems such as those used for cruise control, backup sensors, rain/wiper sensors, dashboard displays, engine controls and monitors, suspension controls, and more. When any of these devices have connectivity to the Internet, they become part of the IoT.

This explosion of embedded systems is certainly improving many products. However, if they have internet access, it's just a matter of time before attackers figure out how to exploit them. Ideally, manufacturers will design and build them with security in mind and include methods to easily update them.

Denial-of-Service Attacks

Denial-of-service (DoS) attacks prevent a system from processing or responding to legitimate traffic or requests for resources and objects. A common form of a DoS attack will transmit so many data packets to a server that it cannot process them all.

Other forms of DoS attacks focus on the exploitation of a known fault or vulnerability in an operating system, service, or application. Exploiting the fault often results in a system crash or 100 percent CPU utilization. No matter what the actual attack consists of, any attack that renders its victim unable to perform normal activities is a DoS attack. DoS attacks can result in decreased performance, system crashes, system reboots, data corruption, blockage of services, and more.

A DoS attack comes from a single system and targets a single system. Of course, this can easily telegraph the attack source. Attackers try to remain anonymous by spoofing the source address. Other times they use a compromised system to launch attacks. The key is that the source address in a DoS attack is rarely the attacker's IP address.

Another form of DoS attack is a *distributed denial-of-service (DDoS)* attack. A DDoS attack occurs when multiple systems attack a single system at the same time. As an example, a group of attackers would launch coordinated attacks against a single system. More often today, though, an attacker will compromise several systems and use them as launching platforms against other victims. Attackers commonly use botnets to launch DDoS attacks as discussed in the previous section.

NOTE DoS attacks are typically aimed at internet-facing systems. In other words, if attackers can access a system via the Internet, it is highly susceptible to a DoS attack. In contrast, DoS attacks are not common for internal systems that are not directly accessible via the Internet. Similarly, many DDoS attacks target internet-facing systems.

There isn't a single DoS or DDoS attack, but these represent types of attacks. Malicious actors are continually creating or discovering new ways to attack systems and have used different protocols doing so. The following sections discuss several specific attacks, and some of these are DoS or DDoS attacks.

The basic preventive measures discussed previously can prevent or mitigate many DoS and DDoS attacks. Additionally, many security companies provide dedicated DDoS mitigation services. These services can sometimes divert or filter enough malicious traffic that the attack doesn't impact users at all.

A *distributed reflective denial-of-service (DRDoS)* attack is a variant of a DoS. It uses a reflected approach to an attack. In other words, it doesn't attack the victim directly but instead manipulates traffic or a network service so that the attacks are reflected back to the victim from other sources. DNS poisoning attacks (covered in Chapter 12), Smurf attacks, and Fraggle attacks (both covered later in this chapter) are examples.

SYN Flood Attack

The *SYN flood attack* is a common DoS attack. It disrupts the standard three-way handshake used by Transmission Control Protocol (TCP) to initiate communication sessions. Normally, a client sends a SYN (synchronize) packet to a server, the server responds with a SYN/ACK (synchronize/acknowledge) packet to the client, and the client then responds with an ACK (acknowledge) packet back to the server. This three-way handshake establishes a communication session that the two systems use for data transfer until the session is terminated with the FIN (finish) or the RST (reset) packet.

> NOTE Chapter 11 discussed the TCP three-way handshake and the TCP communications session in more depth.

However, in a SYN flood attack, the attackers send multiple SYN packets but never complete the connection with an ACK. This is similar to a jokester sticking their hand out to shake hands, but when the other person sticks their hand out in response, the jokester pulls back, leaving the other person hanging.

Figure 17.2 shows an example. Here, a single attacker has sent three SYN packets and the server has responded to each. For each of these requests, the server has reserved system resources to wait for the ACK packet. Servers often wait for the ACK packet for as long as 3 minutes before aborting the attempted session, though administrators can adjust this time.

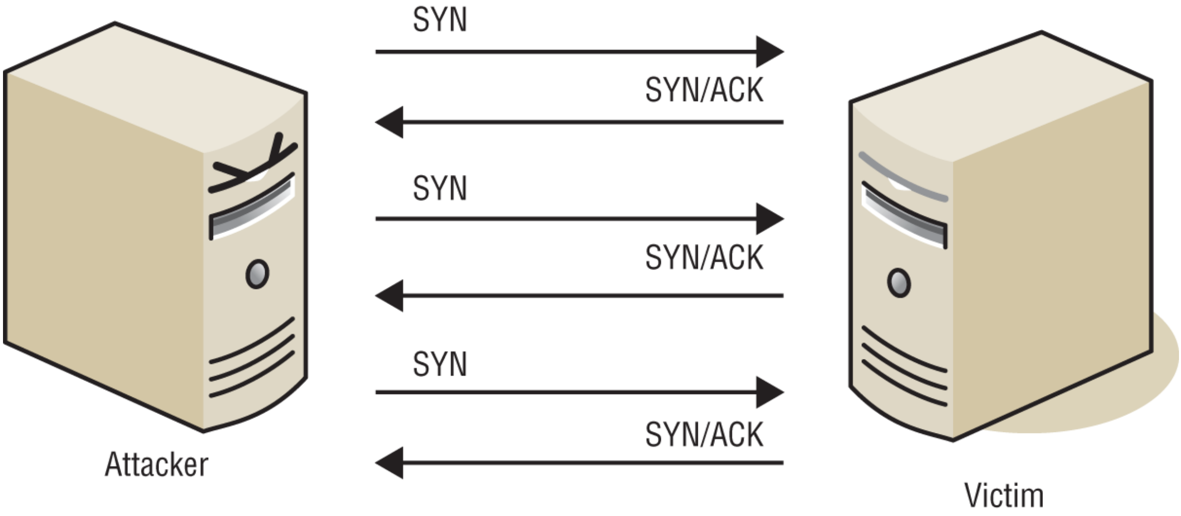


FIGURE 17.2 SYN flood attack

Three incomplete sessions won't cause a problem. However, an attacker will send hundreds or thousands of SYN packets to the victim. Each incomplete session consumes resources, and at some point, the victim becomes overwhelmed and is not able to respond to legitimate requests. The attack can consume available memory and processing power, resulting in the victim slowing to a crawl or actually crashing.

It's common for the attacker to spoof the source address, with each SYN packet having a different source address. This makes it difficult to block the attacker using the source Internet Protocol (IP) address. Attackers have also coordinated attacks launching simultaneous attacks against a single victim as a DDoS attack from a botnet. Limiting the number of allowable open sessions isn't effective as a defense because once the system reaches the limit, it blocks session requests from legitimate users. Increasing the number of allowable sessions on a server results in the attack consuming more system resources, and a server has a finite amount of RAM and processing power.

Using SYN cookies is one method of blocking this attack. These small records consume very few system resources versus the typical resources set aside by a server upon the receipt of a SYN packet from a client. When the server receives an ACK, it checks the SYN cookies and establishes a session. Firewalls often include mechanisms to check for SYN attacks, as do intrusion detection and prevention systems.

Another method of blocking this attack is to reduce the amount of time a server will wait for an ACK. It is typically 3 minutes by default, but in normal operation it rarely takes a legitimate system three minutes to send the ACK packet. By reducing the time, half-open sessions are flushed from the system's memory more quickly.

TCP Reset Attack

Another type of attack that manipulates the TCP session is the TCP reset attack. Sessions are normally terminated with either the FIN (finish) or the RST (reset) packet. Attackers can spoof the source IP address in a RST packet and disconnect active sessions. The two systems then need to reestablish the session. This is primarily a threat for systems that need persistent sessions to maintain data with other systems. When the session is reestablished, they need to re-create the data, so it's much more involved than just sending three packets back and forth to establish the session.

Smurf and Fraggle Attacks

Smurf and Fraggle attacks are both DoS attacks. A *Smurf attack* is another type of flood attack, but it floods the victim with Internet Control Message Protocol (ICMP) echo reply packets instead of with TCP SYN packets. More specifically, it is a spoofed broadcast ping request using the IP address of the victim as the source IP address. That's a mouthful, so it's worthwhile to break it down.

Ping uses ICMP to check connectivity with remote systems. Normally, ping sends an echo request to a single system, and the system responds with an echo reply. However, in a Smurf attack the attacker sends the echo request out as a broadcast to all systems on the network and spoofs the source IP address. All these systems respond with echo replies to the spoofed IP address, flooding the victim with traffic.

Smurf attacks take advantage of an amplifying network (also called a Smurf amplifier) by sending a directed broadcast through a router. All systems on the amplifying network then attack the victim. However, RFC 2644, released in 1999, changed the standard default for routers so that they do not forward directed broadcast traffic. When administrators correctly configure routers in compliance with RFC 2644, a network cannot be an amplifying network. This limits Smurf attacks to a single network. Additionally, it is common to disable ICMP on firewalls, routers, and even many servers to prevent this type of attack using ICMP. When standard security practices are used, Smurf attacks are rarely a problem today.

Fraggle attacks are similar to Smurf attacks. However, instead of using ICMP, a Fraggle attack uses UDP packets over UDP port 7 (echo protocol) and port 19 (character generator protocol). The Fraggle attack will broadcast a UDP packet using the spoofed IP address of the victim. All systems on the network will then send traffic to the victim, just as with a Smurf attack. A variant of a Fraggle attack is a UDP flooding attack using random UDP ports.

Ping Flood

A *ping flood attack* floods a victim with ping requests. This can be very effective when launched by bots within a botnet as a DDoS attack. If tens of thousands of systems simultaneously send ping requests to a system, the system can be overwhelmed trying to answer the ping requests. The victim will not have time to respond to legitimate requests.

A common way that systems handle this today is by blocking ICMP echo request packets. This blocks the ping traffic but not all ICMP traffic. Active intrusion detection systems can detect a ping flood and modify the environment to block ICMP echo requests during the attack.

Legacy Attacks

Many attacks that were successful in the past aren't successful today. However, attackers have a long history of creating attack variants that do succeed. We can't predict what those variants will be next year, but understanding some of the legacy attacks makes it easier to recognize the new variants when they appear. We've listed a few here:

- **Ping of Death:** A Ping-of-Death attack used oversized ping packets. Some operating systems couldn't handle them. In some cases, the systems crashed, and in other cases, the attack caused a buffer overflow error.
- **Teardrop:** A Teardrop attack fragments IP data packets, making them difficult or impossible to be put back together by the receiving system. This often caused systems to crash.
- **LAND:** In a LAND (local area network denial) attack, the attack sends spoofed SYN packets to a victim using the victim's IP address as both the source and destination IP address. A variant is a Banana attack, which redirects outgoing messages from a system back to the system, shutting down all external communication.

NOTE Many other types of attacks cause buffer overflow errors (discussed in Chapter 21). When vendors discover bugs that can cause a buffer overflow, they release patches to fix them. One of the best protections against any buffer overflow attack is to keep a system up-to-date with current patches. Additionally, production systems should not include untested code or allow the use of system or root-level privileges from applications.

Zero-Day Exploit

A *zero-day exploit* refers to an attack on a system exploiting a vulnerability that is unknown to others. However, security professionals use the term in different contexts and it has some minor differences based on the context. Here are some examples:

Attacker discovers a vulnerability first. When an attacker discovers a vulnerability, the attacker can easily exploit it because the attacker is the only one aware of the vulnerability. At this point, the vendor is unaware of the vulnerability and has not developed or released a patch. This is the common definition of a zero-day exploit.

Vendor learns of vulnerability but hasn't released a patch. When vendors learn of a vulnerability, they evaluate the seriousness of the threat and prioritize the development of a patch. Software patches can be complex and require extensive testing to ensure that the patch does not cause other problems. Vendors may develop and release patches within days for serious threats, or they may take months to develop and release a patch for a problem they do not consider serious. Attacks exploiting the vulnerability during this time are often called zero-day exploits because the public does not know about the vulnerability.

Vendor releases patch and systems are attacked within 24 hours. Once a patch is developed, released, and applied, systems should no longer be vulnerable to the exploit. However, organizations often take time to evaluate and test a patch before applying it, resulting in a gap between when the vendor releases the patch and when administrators apply it. Microsoft typically releases patches on the second Tuesday of every month, commonly called "Patch Tuesday." Attackers often try to reverse-engineer the patches to understand them and then exploit them the next day, commonly called "Exploit Wednesday." Some people refer to an attack the day after the vendor releases a patch as a zero-day attack.

> NOTE If an organization doesn't have an effective patch management system, they can have systems that are vulnerable to known exploits. If an attack occurs weeks or months after a vendor releases a patch, this is not a zero-day exploit. Instead, it is an attack on an unpatched system.

Methods used to protect systems against zero-day exploits include many of the basic preventive measures. Ensure that systems are not running unneeded services and protocols to reduce a system's attack surface, enable both network-based and host-based firewalls to limit potentially malicious traffic, and use intrusion detection and prevention systems to help detect and block potential attacks. Additionally, honeypots give administrators an opportunity to observe attacks and may reveal an attack using a zero-day exploit. Honeypots are explained later in this chapter.

Man-in-the-MiddleOn-path Attacks

A *man-in-the-middle (MiTM) attack* (sometimes called an *on-path attack)* occurs when a malicious user establishes a position between two endpoints of an ongoing communication. In this context, the two endpoints are two computers in a network. Note that the MiTM attacker doesn't need to be physically between the two systems for all MiTM attacks. In attacks, the attacker is simply able to monitor all of the traffic between the two systems.

There are two types of man-in-the-middle attacks. One involves copying or sniffing the traffic between two parties, which is basically a sniffer attack as described in Chapter 14. The other type involves attackers positioning themselves in the line of communication, where they act as a store-and-forward or proxy mechanism, as shown in Figure 17.3. The client and server think they are connected directly to each other. However, the malicious actor captures and forwards all data between the two systems. An attacker can collect logon credentials and other sensitive data as well as change the content of messages exchanged between the two systems.

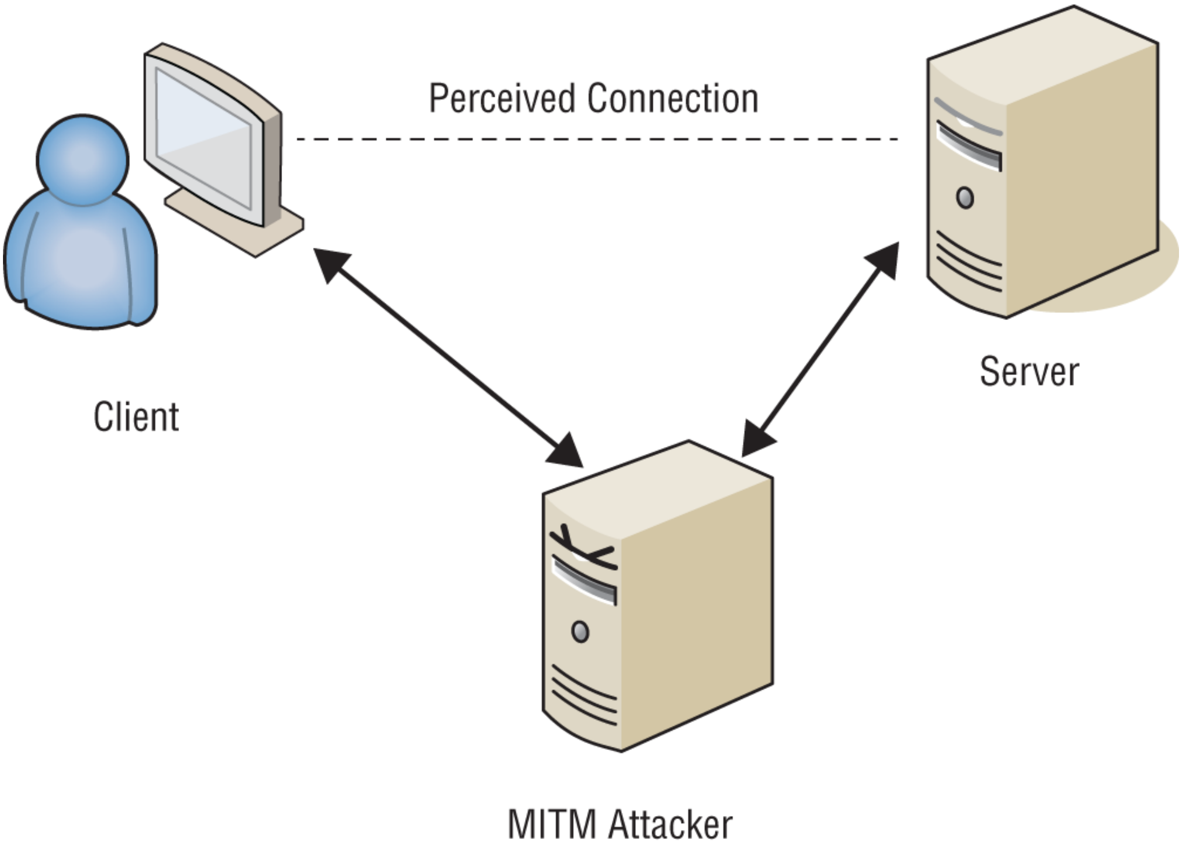


FIGURE 17.3 A man-in-the-middle attack

Man-in-the-middle attacks require more technical sophistication than many other attacks because the attacker needs to successfully impersonate a server from the perspective of the client and impersonate the client from the perspective of the server. A man-in-the-middle attack will often require a combination of multiple attacks. For example, the attacker may alter routing information and DNS values, acquire and install encryption certificates to break into an encrypted tunnel, or falsify Address Resolution Protocol (ARP) lookups as a part of the attack.

Some man-in-the-middle attacks are thwarted by keeping systems up-to-date with patches. An intrusion detection system cannot usually detect man-in-the-middle or hijack attacks, but it can detect abnormal activities occurring over communication links and raise alerts on suspicious activity. Many users often use VPNs to avoid these attacks. Some VPNs are hosted by an employee's organization, but there are also several commercially available VPNs that anyone can use, typically at a cost.

Sabotage

Employee *sabotage* is a criminal act of destruction or disruption committed against an organization by an employee. It can become a risk if an employee is knowledgeable enough about the assets of an organization, has sufficient access to manipulate critical aspects of the environment, and becomes a disgruntled employee. Employee sabotage occurs most often when employees suspect they will be terminated without just cause or if employees retain access after being terminated.

This is another important reason employee terminations should be handled swiftly, and account access should be disabled as soon as possible after the termination. Swift action limits the risk of a disgruntled employee becoming an insider threat. Other safeguards against employee sabotage are intensive auditing, monitoring for abnormal or unauthorized activity, keeping lines of communication open between employees and managers, and properly compensating and recognizing employees for their contributions.

Intrusion Detection and Prevention Systems

The previous section described many common attacks. Attackers are constantly modifying their attack methods, so attacks typically morph over time. Similarly, detection and prevention methods change to adapt to new attacks. Intrusion detection systems (IDSs) and intrusion prevention systems (IPSs) are two methods organizations typically implement to detect and prevent attacks, and they have improved over the years. Together, they use the term intrusion detection and prevention system (IDPS).

An *intrusion* occurs when an attacker can bypass or thwart security mechanisms and access an organization's resources. *Intrusion detection* is a specific form of monitoring that monitors events (often in real time) to detect abnormal activity indicating a potential incident or intrusion. An *intrusion detection system (IDS)* automates the inspection of logs and real-time system events to detect intrusion attempts and system failures. Because an IPS includes detection capabilities, you'll often see them referred to as intrusion detection and prevention systems (IDPSs).

IDSs are an effective method of detecting many DoS and DDoS attacks. They can recognize attacks that come from external connections, such as an attack from the Internet, and attacks that spread internally, such as a malicious worm. Once they detect a suspicious event, they respond by sending alerts or raising alarms. In some cases, they can modify the environment to stop an attack. A primary goal of an IDS is to provide a means for a timely and accurate response to intrusions.

NOTE An IDS is intended as part of a defense-in-depth security plan. It will work with and complement other security mechanisms such as firewalls, but it does not replace other security mechanisms.

An intrusion prevention system (IPS) includes all the capabilities of an IDS but can also take additional steps to stop or prevent intrusions. If desired, administrators can disable an IPS's extra features, essentially causing it to function as an IDS.

NIST SP 800-94—Guide to Intrusion Detection and Prevention Systems (IDPS), provides comprehensive coverage of both intrusion detection and intrusion prevention systems, but for brevity uses IDPS throughout the document to refer to both. In this chapter, we are describing methods used by IDSs to detect attacks, how they can respond to attacks, and the types of IDSs available. We are then adding information on IPSs where appropriate.

Knowledge- and Behavior-Based Detection

An IDS actively watches for suspicious activity by monitoring network traffic and inspecting logs. For example, an IDS can have sensors or agents monitoring key devices such as routers and firewalls on a network. These devices have logs that can record activity, and the sensors can forward these log entries to the IDS for analysis. Some sensors send all the data to the IDS, whereas other sensors inspect the entries and only send specific log entries based on how administrators configure the sensors.

The IDS evaluates the data and can detect malicious behavior using two common methods: knowledge-based detection and behavior-based detection. In short, knowledge-based detection uses signatures similar to the signature definitions used by anti-malware software. Behavior-based detection doesn't use signatures but instead compares activity against a baseline of normal performance to detect anomalies and abnormal behavior. Many IDSs use a combination of both methods.

Knowledge-Based Detection The most common method of detection is *knowledge-based detection* (also called *signature-based detection* or pattern-matching detection). It uses a database of known attacks developed by the IDS vendor. For example, some automated attack tools are available to launch SYN flood attacks, and these tools have known patterns and characteristics defined in a signature database. Real-time traffic is matched against the database, and if the IDS finds a match, it raises an alert. A primary benefit of this method is that it has a low false-positive rate. The primary drawback of a knowledge-based IDS is that it is effective only against known attack methods. New attacks, or slightly modified versions of known attacks, often go unrecognized by the knowledge-based IDS.

Knowledge-based detection on an IDS is similar to signature-based detection used by anti-malware applications. The anti-malware application has a database of known malware and checks files against the database looking for a match. Just as anti-malware software must be regularly updated with new signatures from the anti-malware vendor, IDS databases must be regularly updated with new attack signatures. IDS vendors commonly provide automated methods to update the signatures.

Behavior-Based Detection The second detection type is *behavior-based detection* (also called statistical intrusion detection, anomaly-based detection, and heuristics-based detection). Behavior-based detection starts by creating a baseline of normal activities and events on the system. Once it has accumulated enough baseline data to determine normal activity, it can detect abnormal activity that may indicate a malicious intrusion or event.

This baseline is often created over a finite period such as a week. If the network is modified, the baseline needs to be updated. Otherwise, the IDS may alert you to normal behavior that it identifies as abnormal. Some products continue to monitor the network to learn more about normal activity and will update the baseline based on the observations.

NOTE Chapter 21 covers user and entity behavior analytics (UEBA) functions. UEBA tools create user profiles (similar to a baseline for a network) based on individual behavior. They then watch for deviations in normal behavior that may indicate malicious activity.

Behavior-based IDSs use the baseline, activity statistics, and heuristic evaluation techniques to compare current activity against previous activity to detect potentially malicious events. Many can perform stateful packet analysis similar to how stateful inspection firewalls (covered in Chapter 11) examine traffic based on the state or context of network traffic.

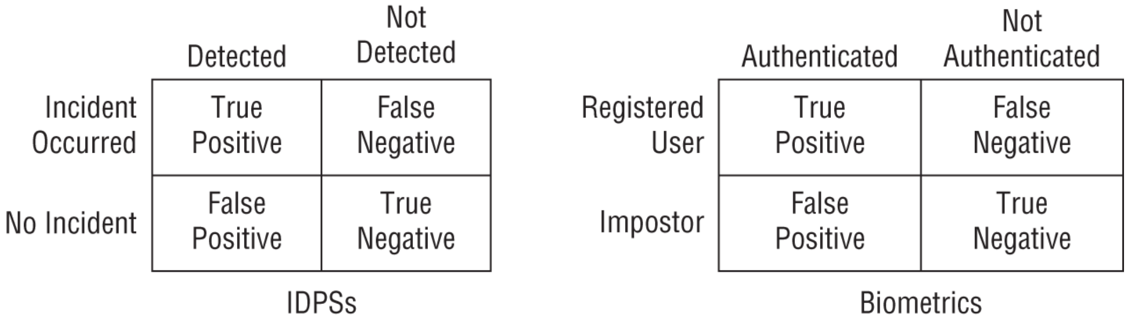
Anomaly analysis adds to an IDS's capabilities by allowing it to recognize and react to sudden increases in traffic volume or activity, multiple failed login attempts, logons or program activity outside normal working hours, or sudden increases in error or failure messages. All of these could indicate an attack that a knowledge-based detection system may not recognize.

A behavior-based IDS can be labeled an expert system or a pseudo-artificial intelligence system because it can learn and make assumptions about events. In other words, the IDS can act like a human expert by evaluating current events against known events. The more information provided to a behavior-based IDS about normal activities and events, the more accurately it can detect anomalies. A significant benefit of a behavior-based IDS is that it can detect newer attacks that have no signatures and are not detectable with the signature-based method.

False Positive or True Negative?

The concept of false positives, false negatives, true positives, and true negatives often causes confusion. However, there are only four possibilities, and with IDPSs they are related to an incident and detection. Either an incident occurred or it didn't, and the IDPS either detected it or it didn't.

The following graphic shows the four possibilities and the following bullets explain them.



	Detected	Not Detected
Incident Occurred	True Positive	False Negative
No Incident	False Positive	True Negative

IDPSs

	Authenticated	Not Authenticated
Registered User	True Positive	False Negative
Impostor	False Positive	True Negative

Biometrics

- True positive: An incident occurs and is detected.
- False negative: An incident occurs but is not detected.
- False positive: An incident is detected but did not occur.
- True negative: An incident does not occur and is not detected.

You'll see the same concepts used in different areas. As an example, biometrics have four possibilities, too. After a user registers with a biometric system, the system should be able to authenticate the user. In contrast, the biometric system shouldn't authenticate impostors (or users who haven't registered with the biometric system).

- True positive: A registered user tries to authenticate and is authenticated.
- False negative: A registered user tries to authenticate but is not authenticated (or is rejected).

- False positive: An impostor tries to authenticate and is authenticated.
- True negative: An impostor tries to authenticate but is not authenticated.

The primary drawback of a behavior-based IDS is that it often raises many false alarms, also called false alerts or false positives. In other words, it incorrectly indicates an attack when an attack isn't present. Patterns of user and system activity can vary widely during normal operations, making it difficult to define normal and abnormal activity boundaries accurately.

In contrast, signature-based systems have a low false positive alarm rate. Either the traffic matches the known signature and is a positive, causing an alarm, or it doesn't. However, signature-based systems can have a high false-negative rate, especially against new attacks. In other words, they do not recognize new attacks because they don't have a signature to detect them, and they don't raise an alarm.

Real World Scenario

False Alarms

Many IDS administrators have a challenge finding a balance between the number of false alarms or alerts that an IDS sends and ensuring that the IDS reports actual attacks. In one organization we know about, an IDS sent a series of alerts over a couple of days that were aggressively investigated but turned out to be false alarms. Administrators began losing faith in the system and regretted wasting time chasing these false alarms.

Later, the IDS began sending alerts on an actual attack. However, administrators were actively troubleshooting another issue that they knew was real, and they didn't have time to chase what they perceived as more false alarms. They simply dismissed the alarms on the IDS and didn't discover the attack until a few days later.

IDS Response

Although knowledge-based and behavior-based IDSs detect incidents differently, they both use an alert system. When the IDS detects an event, it triggers an alarm or alert. It can then respond using a passive or active method. A passive response logs the event and sends a notification. An active response changes the environment to block the activity in addition to logging and sending a notification.

NOTE In some cases, you can measure a firewall's effectiveness by placing a passive IDS before the firewall and another passive IDS after the firewall. By examining the alerts in the two IDSs, you can determine what attacks the firewall is blocking in addition to determining what attacks are getting through.

Passive Response Notifications can be sent to administrators in different ways, such as via email or text messages. In some cases, the alert can generate a report detailing the activity leading up to the event, and logs are available for administrators to get more information if needed. Many 24-hour network operations centers (NOCs) have central monitoring screens viewable by everyone in the main support center. For example, a single wall can have multiple large-screen monitors providing data on different elements of the NOC. The IDS alerts can be displayed on one of these screens to ensure that personnel are aware of the event. These instant notifications help administrators respond quickly and effectively to unwanted behavior.

Active Response Active responses can modify the environment using several different methods. Typical responses include modifying firewall ACLs to block traffic based on ports, protocols, and source addresses, and even disabling all communications over specific cable segments. For example, if an IDS detects a SYN flood attack from a single IP address, the IDS can change the ACL to block all traffic from this IP address. Similarly, if the IDS detects a ping flood attack from multiple IP addresses, it can change the ACL to block all ICMP traffic. The “Firewalls” section, later in this chapter, discusses firewall ACLs in greater depth. An IDS can also block access to resources for suspicious or ill-behaved users. Security administrators configure these active responses in advance and tweak them based on changing needs with the environment.

NOTE An IDS that uses an active response is sometimes referred to as an IPS. This is accurate in some situations. However, an IPS (described later in this section) is placed inline with the traffic. If an active IDS is placed inline with the traffic, it is an IPS. If it is not placed inline with the traffic, it isn't a true IPS because it can only respond to the attack after it has detected an attack in progress. NIST SP 800-94 recommends placing all active IDSs in line with the traffic so that they function as IPSs.

Host- and Network-Based IDSs

IDS types are commonly classified as host-based and network-based. A *host-based IDS (HIDS)* monitors a single computer or host. A *network-based IDS (NIDS)* monitors a network by observing network traffic patterns.

A less-used classification is an application-based IDS, which is a specific type of network-based IDS. It monitors specific application traffic between two or more servers. For example, an application-based IDS can monitor traffic between a web server and a database server looking for suspicious activity.

Host-Based IDS An HIDS monitors activity on a single computer, including process calls and information recorded in system, application, security, and host-based firewall logs. It can often examine events in more detail than an NIDS can, and it can pinpoint specific files compromised in an attack. It can also track processes employed by the attacker.

A benefit of HIDSs over NIDSs is that HIDSs can detect anomalies on the host system that NIDSs cannot detect. For example, an HIDS can detect infections where an intruder has infiltrated a system and is controlling it remotely. You may notice that this sounds similar to what anti-malware software will do on a computer. It is. Many HIDSs include anti-malware capabilities.

Although many vendors recommend installing host-based IDSs on all systems, this isn't common due to some of the disadvantages of HIDSs. Instead, many organizations choose to install HIDSs only on key servers as an added level of protection. Some of the disadvantages of HIDSs are related to the cost and usability. HIDSs are more costly to manage than NIDSs because they require administrative attention on each system, whereas NIDSs usually support centralized administration. An HIDS cannot detect network attacks on other systems. Additionally, it will often consume a significant amount of system resources, degrading the host system's performance. Although it's often possible to restrict the system resources used by the HIDS, this can result in it missing an active attack. Additionally, HIDSs are easier for an intruder to discover and disable, and their logs are maintained on the system, making the logs susceptible to modification during a successful attack.

Network-Based IDS An NIDS monitors and evaluates network activity to detect attacks or event anomalies. A single NIDS can monitor a large network by using remote sensors to collect data at key network locations that send the data to a central management console such as a security information and event management (SIEM) system, described later in this chapter. These sensors can monitor traffic at routers, firewalls, network switches that support port mirroring, and other types of network taps.

Monitoring Encrypted Traffic

Most internet traffic is encrypted using Transport Layer Security (TLS) with HTTPS. Although encryption helps ensure data privacy in transit as it travels over the internet, it also presents challenges for IDPSs.

As an example, imagine a user unwittingly establishes a secure HTTPS session with a malicious site. The malicious site then attempts to download malicious code to the user's system through this channel. Because the malicious code is encrypted, the IDPS cannot examine it, and the code gets through to the client.

Similarly, many botnets have used encryption to bypass inspection by an IDPS. When a bot contacts a command-and-control server, it often establishes an HTTPS session first. It can use this encrypted session to send harvested passwords and other collected data, and receive commands from the server for future activity.

One solution that many organizations have begun implementing is the use of TLS decryptors, sometimes called SSL decryptors. A TLS decryptor detects TLS traffic, takes steps to decrypt it, and sends the decrypted traffic to an IDPS for inspection. This can be very expensive in terms of processing power, so a TLS decryptor is often a stand-alone hardware appliance dedicated to this function, but it can be within an IDPS solution, a next-generation firewall, or some other appliance. Additionally, it is typically placed inline with the traffic, ensuring that all traffic to and from the Internet passes through it.

The TLS decryptor detects and intercepts a TLS handshake between an internal client and an internet server. It then establishes two HTTPS sessions. One is between the internal client and the TLS decryptor; the second is between the TLS decryptor and the Internet server. Although the traffic is transmitted using HTTPS, it is decrypted on the TLS decryptor.

There is a weakness with TLS decryptors, though. Advanced persistent threats (APTs) often encrypt traffic before exfiltrating it out of a network. The encryption is typically performed on a host before establishing a connection with a remote system and sending it. Because the traffic is encrypted on the client and not within a TLS session, the TLS decryptor cannot decrypt it. Similarly, an IDPS may be able to detect that this traffic is encrypted, but it won't be able to decrypt the traffic so that it can inspect it.

Switches are often used as a preventive measure against rogue sniffers. If the IDS is connected to a normal port on the switch, it will capture only a small portion of the network traffic, which isn't very useful. Instead, the switch can be configured to mirror all traffic to a specific port (commonly called port mirroring) used by the IDS. On Cisco switches, the port used for port mirroring is referred to as a Switched Port Analyzer (SPAN) port.

The NIDS central console is often installed on a hardened single-purpose computer. This reduces vulnerabilities in the NIDS and can allow it to operate almost invisibly, making it much harder for attackers to discover and disable it. An NIDS has very little negative effect on the overall network performance. When it is deployed on a single-purpose system, it doesn't adversely affect any other computer's performance. On networks with large volumes of traffic, a single NIDS may be unable to keep up with the flow of data, but adding additional systems to balance the load is possible.

An NIDS can often discover the source of an attack by performing Reverse Address Resolution Protocol (RARP) or reverse DNS lookups. However, because attackers often spoof IP addresses or launch attacks by bots via a botnet, additional investigation is required to determine the actual source. This can be a laborious process and is beyond the scope of the IDS. However, it is possible to discover the source of spoofed IPs with some investigation.

> WARNING It is unethical, risky, and often illegal to launch counterstrikes against an intruder or to attempt to reverse-hack an intruder's computer system. Instead, rely on your logging capabilities and sniffing collections to provide sufficient data to prosecute criminals or improve your environment's security.

An NIDS can usually detect the initiation of an attack or ongoing attacks, but it can't always provide information about an attack's success. It won't know if an attack affected specific systems, user accounts, files, or applications. For example, an NIDS may discover that an attacker sent a buffer overflow exploit through the network, but it won't necessarily know whether the exploit successfully infiltrated a system. However, after administrators receive the alert, they can check relevant systems. Additionally, investigators can use the NIDS logs as part of an audit trail to learn what happened.

Intrusion Prevention Systems

An *intrusion prevention system (IPS)* is a special type of active IDS that attempts to detect and block attacks before they reach target systems. A distinguishing difference between an NIDS and a network-based IPS (NIPS) is that the NIPS is placed inline with the traffic, as shown in Figure 17.4. In other words, all traffic must pass through the NIPS and the NIPS can choose what traffic to forward and what traffic to block after analyzing it. This allows the NIPS to prevent an attack from reaching a target.

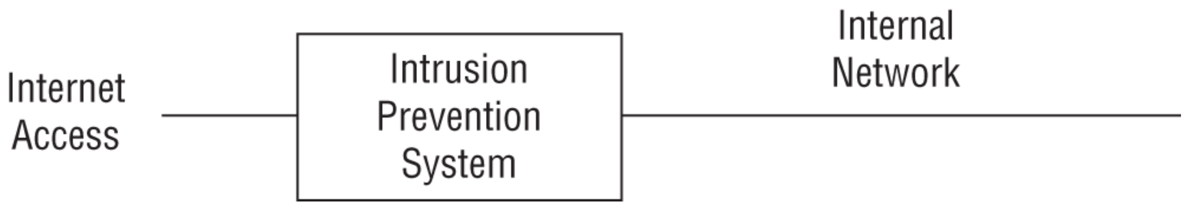


FIGURE 17.4 Intrusion prevention system

In contrast, an active NIDS that is not placed inline can check the activity only after it has reached the target. The active NIDS can take steps to block an attack after it starts but cannot prevent it.

An NIPS can use knowledge-based detection and/or behavior-based detection, just like any other IDS. Additionally, it can log activity and provide notification to administrators just as an IDS would.

NOTE A current trend is the replacement of NIDSs with NIPSs. This can often be done by placing the NIDS inline with the traffic, as shown in Figure 17.4. This allows the device to analyze all the traffic because all the traffic goes through the device, and the device chooses what traffic to forward and what traffic to block. Similarly, many appliances that include detection and prevention capabilities focus their use on an NIPS. Because an NIPS is placed inline with the traffic, it can inspect all traffic as it occurs.

Specific Preventive Measures

Although intrusion detection and prevention systems go a long way toward protecting networks, administrators typically implement additional security controls to protect their networks. The following sections describe several of these as additional preventive measures.

Honeypots and Honeynets

Honeypots are individual computers created as a trap or a decoy for intruders or insider threats. A *honeynet* is two or more networked honeypots used together to simulate a network. They look and act like legitimate systems, but they do not host data of any real value for an attacker. Administrators often configure honeypots with vulnerabilities to tempt intruders into attacking them. They may be unpatched or have security vulnerabilities that administrators purposely leave open. The goal is to grab intruders' attention and keep them away from the legitimate network that is hosting valuable resources. Legitimate users wouldn't access the honeypot, so any access to a honeypot is most likely an unauthorized intruder.

In addition to keeping the attacker away from a production environment, the honeypot allows administrators to observe an attacker's activity without compromising the live environment. In some cases, the honeypot is designed to delay an intruder long enough for the automated IDS to detect the intrusion and gather as much information about the intruder as possible. The longer the attacker spends with the honeypot, the more time an administrator has to investigate the attack and potentially identify the intruder. Some security professionals, such as those engaged in security research, consider honeypots to be effective countermeasures against zero-day exploits because they can observe the attacker's actions.

Honeypots and honeynets can be placed anywhere on a network, but administrators often host them on virtual systems. These are much simpler to re-create after an attack. For example, administrators can configure the honeypot and then take a snapshot of a honeypot virtual machine. If an attacker modifies the environment, administrators can revert the machine to the state it was in when they took the snapshot. When using VMs, administrators should monitor the honeypot or honeynet closely. Attackers can often detect when they are within a VM and may attempt a VM escape attack to break out of the VM.

Administrators often include pseudo-flaws on honeypots to emulate well-known operating system vulnerabilities. *Pseudo-flaws* are false vulnerabilities or apparent loopholes intentionally implanted in a system in an attempt to tempt attackers. Attackers seeking to exploit a known flaw might stumble across a pseudo-flaw and think that they have successfully penetrated a system. More sophisticated pseudo-flaw mechanisms actually simulate the penetration and convince the attacker that they have gained additional access privileges to a system. However, while the attacker is exploring the system, monitoring and alerting mechanisms trigger and alert administrators to the threat.

The use of honeypots raises the issue of enticement versus entrapment. An organization can legally use a honeypot as an enticement device if the intruder discovers it through no outward efforts of the honeypot owner. Placing a system on the Internet with open security vulnerabilities and active services with known exploits is enticement. Enticed attackers make their own decisions to perform illegal or unauthorized actions. Entrapment, which is illegal, occurs when the honeypot owner actively solicits visitors to access the site and then charges them with unauthorized intrusion. In other words, it is entrapment when you trick or encourage someone into performing an illegal or unauthorized action. Laws vary in different countries, so it's important to understand local laws related to enticement and entrapment.

Warning Banners

Warning banners inform users and intruders about basic security policy guidelines. They typically mention that online activities are audited and monitored, and they often provide reminders of restricted activities. In most situations, the wording in banners is important from a legal standpoint because these banners can legally bind users to a permissible set of actions, behaviors, and processes.

Unauthorized personnel who are somehow able to log on to a system also see the warning banner. In this case, you can think of a warning banner as an electronic equivalent of a “no trespassing” sign. Most intrusions and attacks can be prosecuted when warnings clearly state that unauthorized access is prohibited and that any activity will be monitored and recorded.

TIP Warning banners inform both authorized and unauthorized users. These banners typically remind authorized users of the content in acceptable use agreements.

Anti-malware

The most important protection against malicious code is the use of anti-malware software with up-to-date signature files and heuristic capabilities. Attackers regularly release new malware and often modify existing malware to prevent detection by anti-malware software. Anti-malware software vendors look for these changes and develop new signature files to detect new and modified malware. Years ago, anti-malware vendors recommended updating signature files once a week. However, most anti-malware software today includes the ability to check for updates several times a day without user intervention.

NOTE Originally, anti-malware software focused on viruses, and it was called antivirus software. However, as malware expanded to include other malicious code such as Trojans, worms, spyware, and rootkits, vendors expanded their anti-malware software abilities. Today, most anti-malware software will detect and block most malware, so technically, it is anti-malware software. However, most vendors still market their products as antivirus software. The CISSP objectives use the term anti-malware.

Many organizations use a multipronged approach to block malware and detect any malware that gets in. Firewalls with content-filtering capabilities (or specialized content-filter appliances) are commonly used at the boundary between the Internet and the internal network to filter out any type of malicious code. Specialized anti-malware software is installed on email servers to detect and filter out any type of malware passed via email. Additionally, anti-malware software is installed on each system to detect and block malware. Organizations often use a central server to deploy anti-malware software, download updated definitions, and push these definitions out to the clients.

A multipronged approach with anti-malware software on each system in addition to filtering internet content helps protect systems from infections from any source. As an example, using up-to-date anti-malware software on each system will detect and block a virus on an employee's USB flash drive.

Anti-malware vendors commonly recommend installing only one anti-malware application on any system. When a system has more than one anti-malware application installed, the applications can interfere with each other and sometimes cause system problems. Additionally, having more than one scanner can consume excessive system resources.

Following the principle of least privilege also helps. Users will not have administrative permissions on systems and will not be able to install applications that may be malicious. If a virus does infect a system, it can often impersonate the logged-in user. When this user has limited privileges, the virus is limited in its capabilities. Additionally, vulnerabilities related to malware increase as more applications are added. Each additional application provides another potential attack point for malicious code.

Educating users about the dangers of malicious code, how attackers try to trick users into installing it, and what they can do to limit their risks is another protection method. A user can often avoid an infection simply by not clicking a link or opening an attachment sent via email.

Chapter 2 covers social engineering tactics, including phishing, spear phishing, and whaling. When users are educated about these types of attacks, they are less likely to fall for them. Although many users know about these risks, phishing emails continue to flood the Internet and land in users’ inboxes. The only reason attackers keep sending them is that they continue to fool some users.

Education, Policy, and Tools

Malicious software is a constant challenge within any organization using IT resources. Consider Kim, who forwarded a seemingly harmless interoffice joke through email to Larry's account. Larry opened the document, which actually contained active code segments that performed harmful actions on his system. Larry then reported a host of “performance issues” and “stability problems” with his workstation, which he'd never complained about before.

In this scenario, Kim and Larry don't recognize the harm caused by their apparently innocuous activities. After all, sharing anecdotes and jokes through company email is a common way to bond and socialize. What's the harm in that, right? The real question is how can you educate Kim, Larry, and all your other users to be more discreet and discerning in handling shared documents and executables?

The key is a combination of education, policy, and tools. Education should inform Kim that forwarding nonwork materials on the company network is counter to policy and good behavior. Likewise, Larry should learn that opening attachments unrelated to specific work tasks can lead to all kinds of problems (including those he fell prey to here). Policies should clearly identify the acceptable use of IT resources and the dangers of circulating unauthorized materials. Tools such as anti-malware software should be employed to prevent and detect any type of malware within the environment.

Whitelisting and Blacklisting

One of the methods used to control which applications can run and which applications can't run is whitelists and blacklists, though these terms are falling into disuse. Today, it's more common to use the more intuitive phrases allow list (for whitelisting) and deny list or block list (for blacklisting). Using these lists is an effective preventive measure that blocks users from running unauthorized applications.

Using allow lists and deny lists for applications can also help prevent malware infections. The allow list identifies a list of applications authorized to run on a system and blocks all other applications. A deny list identifies a list of applications that are not authorized to run on a system. It's important to understand that a system would only use one list, either an allow list or a deny list.

Some allow lists identify applications using a hashing algorithm to create a hash. However, if an application is infected with a virus, the virus effectively changes the hash, so this type of allow list blocks infected applications from running too. (Chapter 6, “Cryptography and Symmetric Key Algorithms,” covers hashing algorithms in more depth.)

The Apple iOS and iPadOS running on iPhones and iPads, respectively, are examples of extreme versions of allow lists. Users are only able to install apps available from Apple's App Store. Personnel at Apple review and approve all apps on the App Store and quickly remove misbehaving apps. Although it is possible for users to bypass security and jailbreak their iOS devices, most users don't do so, partly because it is a violation of the end-user license agreement (EULA) and voids the warranty.

> NOTE Jailbreaking removes restrictions on iOS devices and permits root-level access to the underlying operating system. It is similar to rooting a device running the Android operating system.

Using a deny list is a good option if administrators know which applications they want to block. For example, if management wants to ensure that users are not running games on their system, administrators can enable tools to block these games.

Firewalls

Chapter 11 discussed firewalls in greater depth, but a few things are worth emphasizing when discussing detection and preventive measures. First, firewalls are preventive and technical controls. They attempt to prevent security incidents using technical methods.

These basic guidelines can provide a lot of protection against attacks:

Block directed broadcasts on routers A directed broadcast acts as a unicast packet until it reaches the destination network. Attackers have used these to flood targeted networks with broadcasts, so it's common to block directed broadcasts. Many routers have the option to change this setting, but it's to block directed broadcasts.

Block private IP addresses at the border Internal networks use private IP address ranges (discussed in Chapter 12), and the Internet uses public IP address ranges. If traffic from the Internet has a source address in a private IP address range, it is a spoofed address, and the firewall should block it.

Basic network firewalls filter traffic based on IP addresses, ports, and some protocols using protocol numbers. It's common to place firewalls at the border or edge of a network (between the Internet and the internal network). This allows it to monitor all incoming and outgoing traffic.

Firewalls include rules within an ACL to allow specific traffic and end with an implicit deny rule. The implicit deny rule blocks all traffic not allowed by a previous rule. For example, a firewall can allow HTTP and HTTPS traffic by allowing traffic using TCP ports 80 and 443, respectively. (Chapter 11 covers logical ports in more depth.)

Many attackers use ping to discover systems or to launch DoS attacks. For example, an attacker can launch a ping flood attack by flooding a system with pings. Ping uses ICMP, so it's common to block pings by blocking ICMP echo requests at border firewalls. This prevents the pings from reaching the internal network from the Internet.

There are other methods of blocking ping. For example, all ICMP traffic uses a protocol number of 1. A firewall can block ping traffic by blocking protocol number 1. However, this method blocks all ICMP traffic, which is similar to using a bazooka to remove an ant from a picnic table.

> NOTE The Internet Assigned Numbers Authority (IANA) maintains a list of well-known ports matched to protocols. IANA also maintains lists of assigned protocol numbers for IPv4 and IPv6. These pages have changed a few times over the years, but a search for "IANA ports protocol numbers" will get you there.

Second-generation firewalls add additional filtering capabilities. For example, an application-level gateway firewall filters traffic based on specific application requirements and *circuit-level gateway firewalls* filter traffic based on the communications circuit. Third-generation firewalls (also called *stateful inspection firewalls* and dynamic packet filtering firewalls) filter traffic based on its state within a stream of traffic.

Application firewalls control traffic going to or from a specific application or service. As an example, a *web application firewall (WAF)* is a specialized application firewall that protects a web server. It inspects all traffic going to a web server and can block malicious traffic such as SQL injection attacks and cross-site scripting (XSS) attacks. This can be processor intensive, so the WAF filters traffic going to the web server but not all network traffic.

A *next-generation firewall (NGFW)* functions as a *unified threat management (UTM)* device and combines several filtering capabilities. It includes traditional functions of a firewall such as packet filtering and stateful inspection. However, an NGFW is able to perform packet inspection techniques, allowing it to identify and block malicious traffic. It can filter malware using definition files and/or whitelists and blacklists. It also includes intrusion detection and/or intrusion prevention capabilities.

Sandboxing

Sandboxing is a virtualization technique that provides a security boundary for applications and prevents the application from interacting with other applications. Anti-malware applications use sandboxing techniques to test unknown applications. If the application displays suspicious characteristics, the sandboxing technique prevents the application from infecting other applications or the operating system.

Application developers often use virtualization techniques to test applications. They create a virtual machine and then isolate it from the host machine and the network. They can then test the application within this sandbox environment without affecting anything outside the virtual machine. Similarly, many anti-malware vendors use virtualization as a sandboxing technique to observe the behavior of malware.

Third-Party Security Services

Some organizations outsource security services to a third party, which is an individual or organization outside the organization. This can include many different types of services, such as auditing and penetration testing.

In some cases, an organization must provide assurances to an outside entity that third-party service providers comply with specific security requirements. For example, organizations processing transactions with major credit cards must comply with the Payment Card Industry Data Security Standard (PCI DSS). These organizations often outsource some of the services, and PCI DSS requires organizations to ensure that service providers also comply with PCI DSS requirements. In other words, PCI DSS doesn't allow organizations to outsource their responsibilities.

Some software-as-a-service (SaaS) vendors provide security services via the cloud. This can include cloud-based solutions similar to next-generation firewalls, UTM devices, and email gateways for spam and malware filtering.

Logging and Monitoring

Logging and monitoring procedures help an organization prevent incidents and provide an effective response when they occur. Logging records events into various logs, and monitoring reviews these events. Combined, they allow an organization to track, record, and review activity, providing overall accountability.

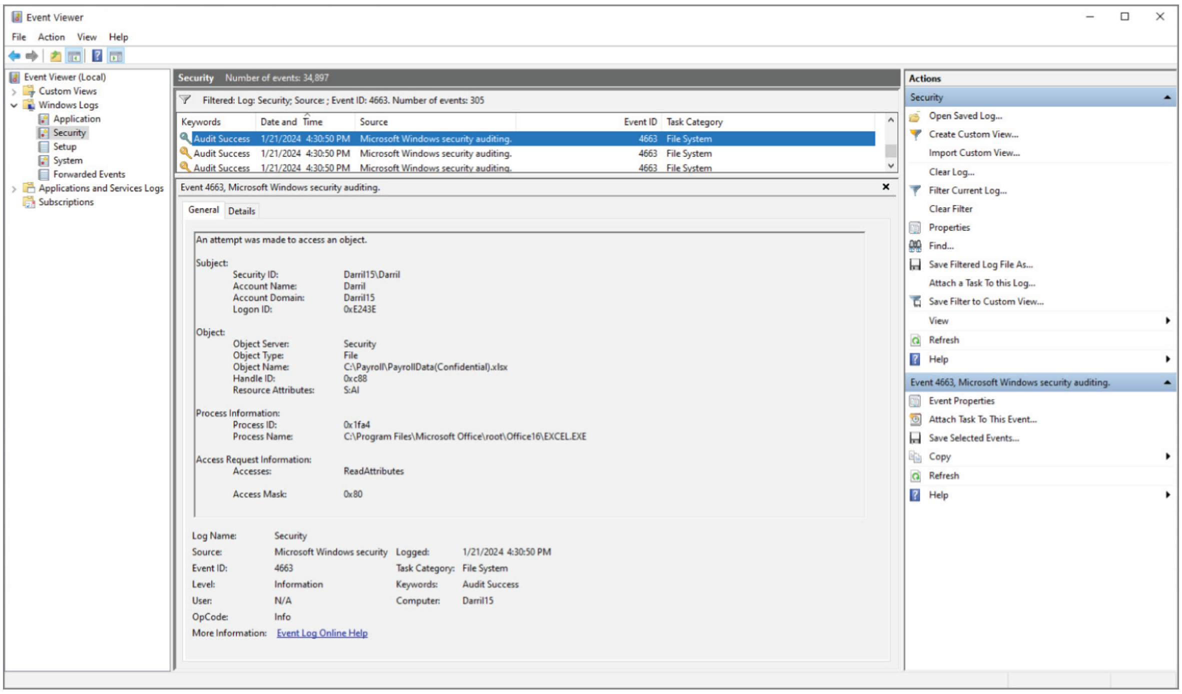
This helps an organization detect undesirable events that can negatively affect confidentiality, integrity, and system availability. It is also useful in reconstructing activity after an event has occurred to identify what happened and sometimes to prosecute those responsible for the activity. The following sections cover common logging and monitoring topics.

Logging Techniques

Logging is the process of recording information about events to a log file or database. Logging captures events, changes, messages, and other data describing activities on a system. Logs will commonly record details such as what happened, when it happened, where it happened, who did it, and sometimes how it happened. When you need to find information about an incident that occurred in the recent past, logs are a good place to start.

For example, Figure 17.5 shows Event Viewer on a Microsoft Windows system with a Security log entry selected and expanded. This log entry shows that a user named Darril accessed a file named `PayrollData (Confidential).xlsx` located in a folder named `C:\Payroll`. It shows that the user accessed the file at 4:30 p.m. on January 21, 2024.

As long as the identification and authentication processes are secure, this is enough to hold Darril accountable for accessing the file. On the other hand, if the organization doesn't use secure authentication processes, and it's easy for someone to impersonate another user, Darril may be wrongly accused. This reinforces the requirement for secure identification and authentication practices as a prerequisite for accountability.



Security Number of events: 34,897

Filtered: Log: Security; Source: ; Event ID: 4663. Number of events: 305

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	1/21/2024 4:30:50 PM	Microsoft Windows security auditing.	4663	File System
Audit Success	1/21/2024 4:30:50 PM	Microsoft Windows security auditing.	4663	File System
Audit Success	1/21/2024 4:30:50 PM	Microsoft Windows security auditing.	4663	File System

Event 4663, Microsoft Windows security auditing.

```
An attempt was made to access an object.

Subject:
        Security ID:            Darril15\Darril
        Account Name:           Darril
        Account Domain:         Darril15
        Logon ID:               0xE243E

Object:
        Object Server:          Security
        Object Type:            File
        Object Name:            C:\Payroll\PayrollData(Confidential).xlsx
        Handle ID:              0xc88
        Resource Attributes:    S:AI

Process Information:
        Process ID:             0x1fa4
        Process Name:           C:\Program Files\Microsoft Office\root\Office16\EXCEL.EXE

Access Request Information:
        Accesses:               ReadAttributes

        Access Mask:            0x80
```

```
Log Name:          Security
Source:            Microsoft Windows security   Logged:          1/21/2024 4:30:50 PM
Event ID:          4663                         Task Category:   File System
Level:             Information                  Keywords:        Audit Success
User:              N/A                          Computer:        Darril15
OpCode:            Info
More Information:  Event Log Online Help
```

FIGURE 17.5 Viewing a log entry

NOTE Logs are often referred to as audit logs, and logging is often called audit logging. However, it's important to realize that auditing (described in Chapter 15, “Security Assessment and Testing”) is more than just logging. Logging will record events, and auditing examines or inspects an environment for compliance.

Common Log Types

There are many different types of logs. The following is a short list of common logs available within an IT environment:

Security Logs Security logs record access to resources such as files, folders, printers, and so on. For example, they can record when a user accessed, modified, or deleted a file, as shown earlier in Figure 17.5. Many systems automatically record access to key system files but require an administrator to enable auditing on other resources before logging access. For example, administrators might configure logging for proprietary data but not for public data posted on a website.

System Logs System logs record system events such as when a system starts or stops, when services start or stop, or when service attributes are modified. If attackers are able to shut down a system and reboot it with a CD or USB flash drive, they can steal data from the system without any record of the data access. Similarly, if attackers are able to stop a service that is monitoring the system, they may be able to access the system without the logs recording their actions. Additionally, attackers sometimes modify the attributes of logs. For example, a service might be set to Disabled, but the attacker can change it to Manual, allowing the attacker to start it at will. Logs that detect when systems reboot, or when services stop or are modified, can help administrators discover potentially malicious activity.

Application Logs These logs record information for specific applications. Application developers choose what to record in the application logs. For example, a database developer can choose to record when anyone accesses specific data objects such as tables or views.

Firewall Logs Firewall logs can record events related to any traffic that reaches a firewall. This includes traffic that the firewall allows and traffic that the firewall blocks. These logs commonly log key packet information such as source and destination IP addresses and source and destination ports but not the packets' actual contents.

Proxy Logs Proxy servers improve internet access performance for users and can control what websites users can visit. Proxy logs include the ability to record details such as what sites specific users visit and how much time they spend on these sites. They can also record when users attempt to visit known prohibited sites.

Change Logs Change logs record change requests, approvals, and actual changes to a system as a part of an overall change management process. A change log can be manually created or created from an internal web page as personnel record activity related to a change. Change logs are useful to track approved changes. They can also be helpful as part of a disaster recovery program. For example, administrators and technicians can use change logs to return a system to its last known state after a disaster. This will include all previously applied changes.

Logging is usually a native feature in an operating system and for most applications and services, which makes it easy for administrators and technicians to configure a system to record specific types of events. Events from privileged accounts, such as Administrator and root user accounts, should be included in any logging plan. Doing so helps deter attacks from a malicious insider and will document activity for prosecution if necessary.

Protecting Log Data

Personnel within the organization can use logs to re-create events leading up to and during an incident, but only if the logs haven't been modified. If attackers can modify the logs, they can erase their activity, effectively nullifying the value of the data. The files may no longer include accurate information and may not be admissible as evidence to prosecute attackers. With this in mind, it's important to protect log files against unauthorized access and unauthorized modification.

It's common to store copies of logs on a central system, such as a security information and event management (SIEM) system, to protect it. Even if an attacker modifies or corrupts the original files, personnel can still use the copy to view the events. One way to protect log files is by assigning permissions to limit their access.

Organizations often have strict policies mandating backups of log files. Additionally, these policies define retention times. For example, organizations might keep archived log files for a year, three years, or any other length of time. Some government regulations require organizations to keep archived logs indefinitely. Security controls such as setting logs to read-only, assigning permissions, and implementing physical security controls protect archived logs from unauthorized access and modifications. It's important to destroy logs when they are no longer needed.

TIP Keeping unnecessary logs can cause excessive labor costs if the organization experiences legal issues. For example, if regulations require an organization to keep logs for one year but the organization has 10 years of logs, a court order can force personnel to retrieve relevant data from these 10 years of logs. In contrast, if the organization keeps only one year of logs, personnel need only search a year's worth of logs, which will take significantly less time and effort.

NIST publishes a significant amount of information on IT security, including Federal Information Processing Standards (FIPS) publications. The Minimum Security Requirements for Federal Information and Information Systems (FIPS 200) specifies the following as the minimum security requirements for audit data:

> Create, protect, and retain information system audit records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful, unauthorized, or inappropriate information system activity.
>
> Ensure that the actions of individual information system users can be uniquely traced to those users so they can be held accountable for their actions.

NOTE You'll find it useful to review NIST documents when preparing for the CISSP exam to give you a broader idea of different security concepts. They are freely available, and you can access them here: http://csrc.nist.gov. You can download the FIPS 200 document here: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.200.pdf.

The Role of Monitoring

Monitoring provides several benefits for an organization, including increasing accountability, help with investigations, and basic troubleshooting. The following sections describe these benefits in more depth.

Audit Trails

Audit trails are records created when information about events and occurrences is stored in one or more databases or log files. They provide a record of system activity and can reconstruct activity leading up to and during security events. Security professionals extract information about an incident from an audit trail to prove or disprove culpability, and much more. Audit trails allow security professionals to examine and trace events in forward or reverse order. This flexibility helps when tracking down problems, performance issues, attacks, intrusions, security breaches, coding errors, and other potential policy violations.

> TIP Audit trails provide a comprehensive record of system activity and can help detect a wide variety of security violations, software flaws, and performance problems.

Using audit trails is a passive form of detection security control. They serve as a deterrent in the same manner that closed-circuit television (CCTV) or security guards do. If personnel know they are being watched and their activities are being recorded, they are less likely to engage in illegal, unauthorized, or malicious activity —at least in theory. Some criminals are too careless or clueless for this to apply consistently. However, more and more advanced attackers take the time to locate and delete logs that might have recorded their activity. This has become a standard practice with many advanced persistent threats (APTs).

Audit trails are also essential as evidence in the prosecution of criminals. They provide a before-and-after picture of the state of resources, systems, and assets. This, in turn, helps determine whether a change or alteration was caused by a user action, the operating system, a software application, or some other source, such as hardware failure. Because data in audit trails can be so valuable, it is important to ensure that the logs are protected to prevent modification or deletion.

Monitoring and Accountability

Monitoring is necessary to ensure that subjects (such as users and employees) can be held accountable for their actions and activities. Users claim an identity (such as with a username) and prove their identity (by authenticating), and audit trails record their activity while they are logged in. Monitoring and reviewing the audit trail logs provide accountability for these users. It is possible to promote positive user behavior and compliance with the organization's security policy by monitoring activity. Users who are aware that logs are recording their IT activities are less likely to try to circumvent security controls or perform unauthorized or restricted activities.

Once a security policy violation or a breach occurs, the source of that violation should be determined. If it is possible to identify the individuals responsible, they should be held accountable based on the organization's security policy. Severe cases can result in terminating employment or legal prosecution.

Legislation often requires specific monitoring and accountability practices. This includes laws such as the Sarbanes–Oxley Act of 2002, the Health Insurance Portability and Accountability Act (HIPAA), and the European Union (EU)'s General Data Protection Regulation (GDPR) that many organizations must abide by.

Real World Scenario

Monitoring Activity

Accountability is necessary at every level of business, from the frontline infantry to the high-level commanders overseeing daily operations. If you don't monitor users' actions and activities on a given system, you cannot hold them accountable for mistakes or misdeeds they commit.

Consider Duane, a quality assurance supervisor for the data entry department at an oil-drilling data-mining company. He sees many highly sensitive documents that include the kind of valuable information that can earn a heavy tip or a bribe from interested parties during his daily routine. He also corrects the kind of mistakes that could cause serious backlash from his clientele. Sometimes, a minor clerical error can cause serious issues for a client's entire project.

Whenever Duane touches or transfers such information on his workstation, his actions leave an electronic trail of evidence that his supervisor, Nicole, can examine if Duane's actions should come under scrutiny. She can observe where he obtained or placed pieces of sensitive information, when he accessed and modified such information, and just about anything else related to the data's handling and processing as it flows in from the source and out to the client.

This accountability protects the company should Duane misuse this information. It also provides Duane with protection against anyone falsely accusing him of misusing the data he handles.

Monitoring and Investigations

Audit trails give investigators the ability to reconstruct events long after they have occurred. They can record access abuses, privilege violations, attempted intrusions, and many different types of attacks. After detecting a security violation, security professionals can reconstruct the conditions and system state leading up to the event, during the event, and after the event through a close examination of the audit trail.

One important consideration is ensuring that logs have accurate timestamps and that these timestamps remain consistent throughout the environment. A common method is to set up an internal Network Time Protocol (NTP) server synchronized to a trusted time source such as a public NTP server. Other systems can then synchronize with this internal NTP server.

NIST operates several time servers that support authentication. Once an NTP server is properly configured, the NIST servers will respond with encrypted and authenticated time messages. The authentication provides assurances that the response came from a NIST server.

> NOTE Systems should have their time synchronized against a centralized or trusted public time server. This ensures that all audit logs record accurate and consistent times for recorded events.

Monitoring and Problem Identification

Audit trails offer details about recorded events that are useful for administrators. They can record system failures, OS bugs, and software errors in addition to malicious attacks. Some log files can even capture the contents of memory when an application or system crashes. This information can help pinpoint the cause of the event and eliminate it as a possible attack. For example, if a system keeps crashing due to faulty memory, crash dump files can help diagnose the problem.

Using log files for this purpose is often labeled as problem identification. Once a problem is identified, performing problem resolution involves little more than following up on the disclosed information.

Monitoring and Tuning Techniques

Monitoring is the process of reviewing information logs, looking for something specific. Personnel can manually review logs or use tools to automate the process. Monitoring is necessary to detect malicious actions by subjects as well as attempted intrusions and system failures. It can help reconstruct events, provide evidence for prosecution, and create reports for analysis.

Tuning is the process of adjusting security controls to better match the needs of the organization and their operational environment. For example, intrusion detection and prevention systems require tuning to reduce the number of false positive alerts that they generate. If a system is too sensitive, it will generate many alerts that will cause administrators to begin to mistrust, and possibly ignore, the system alerts. If a system is not sensitive enough, it may miss a potential intrusion.

It's important to understand that monitoring and tuning are a continuous process. Continuous monitoring ensures that all events are recorded and can be investigated later if necessary. Many organizations increase logging in response to an incident or a suspected incident to gather additional intelligence on attackers. Continuous tuning ensures that the log entries and alerts are relevant and sufficient to meet the organization's security needs.

Log analysis is a detailed and systematic form of monitoring in which the logged information is analyzed for trends and patterns as well as abnormal, unauthorized, illegal, and policy-violating activities. Log analysis isn't necessarily in response to an incident but instead a periodic task, which can detect potential issues.

When manually analyzing logs, administrators simply open the log files and look for relevant data. This process can be very tedious and time-consuming. For example, searching 10 different archived logs for a specific event or ID code can take some time, even when using built-in search tools.

In many cases, logs can produce so much information that important details can get lost in the sheer volume of data, so administrators often use automated tools to analyze the log data. For example, intrusion detection systems (IDSs) actively monitor multiple logs to detect and respond to malicious intrusions in real time. An IDS can help detect and track attacks from external attackers, send alerts to administrators, and record attackers' access to resources.

Multiple vendors sell operations management software that actively monitors systems' security, health, and performance throughout a network. This software automatically looks for suspicious or abnormal activities that indicate problems such as an attack or unauthorized access.

Security Information and Event Management

Many organizations use a centralized application to automate the monitoring of systems on a network. Several terms are used to describe these tools, including security information and event management (SIEM), security event management (SEM), and security information management (SIM). These tools provide centralized logging and real-time analysis of events occurring on systems throughout an organization. They include agents installed on remote systems that monitor for specific events known as alarm triggers. When the trigger occurs, the agents report the event back to the central monitoring software.

Many IDSs and IPSs send collected data to a SIEM system. The system also collects data from many other sources within the network, providing real-time monitoring of traffic and analysis and notification of potential attacks. Additionally, it provides long-term storage of data, allowing security professionals to analyze the data later.

A SIEM typically includes several features. Because it collects data from dissimilar devices, it includes a correlation and aggregation feature converting this data into useful information. Advanced analytic tools within the SIEM can analyze the data and raise alerts and/or trigger responses based on preconfigured rules.

For example, a SIEM can monitor a group of email servers. Each time one of the email servers logs an event, a SIEM agent examines the event to determine whether it is an item of interest. If it is, the SIEM agent forwards the event to a central SIEM server. Depending on the event, it can raise an alarm for an administrator or take some other action. For example, if the send queue of an email server starts backing up, a SIEM application can detect the issue and alert administrators before the problem is serious.

Most SIEMs are configurable, allowing personnel within the organization to specify what items are of interest and need to be forwarded to the SIEM server. SIEMs have agents for just about any type of server or network device, and in some cases, they monitor network flows for traffic and trend analysis. The tools can also collect all the logs from target systems and use machine learning and artificial intelligence techniques to retrieve relevant data. Security professionals can then create reports and analyze the data.

SIEMs often include sophisticated correlation engines. These engines are a software component that collects the data and aggregates it looking for common attributes. It then uses advanced analytic tools to detect abnormalities and sends alerts to security administrators.

Some monitoring tools are also used for inventory and status purposes. For example, tools can query all the available systems and document details, such as system names, IP addresses, operating systems, installed patches, updates, and installed software. These tools can then create reports of any system based on the needs of the organization. For example, they can identify how many systems are active, identify systems with missing patches, and flag systems that have unauthorized software installed.

Software monitoring watches for attempted or successful installations of unapproved software, use of unauthorized software, or unauthorized use of approved software. Software monitoring thus reduces the risk of users inadvertently installing a virus or Trojan horse.

Syslog

RFC 5424 describes the syslog protocol, which is used to send event notification messages. A centralized syslog server receives these syslog messages from devices on a network. The protocol defines how to format the messages and how to send them to the syslog server but not how to handle them.

Syslog has historically been used in Unix and Linux systems. These systems include the syslogd daemon, which handles all incoming syslog messages, similar to how a SIEM server provides centralized logging. Some syslogd extensions, such as syslog-ng and Rsyslog, allow the syslog server to accept messages from any source, not just Unix and Linux systems.

Sampling

Sampling, or *data extraction*, is the process of extracting specific elements from a large collection of data to construct a meaningful representation or summary of the whole. In other words, sampling is a form of data reduction that allows someone to glean valuable information by looking at only a small sample of data in an audit trail.

Statistical sampling uses precise mathematical functions to extract meaningful information from a large volume of data and is thus similar to the science used by pollsters to learn the opinions of large populations without interviewing everyone in the population. There is always a risk that sampled data is not an accurate representation of the whole body of data, and statistical sampling can identify the margin of error.

Clipping Levels

Clipping is a form of nonstatistical sampling. It selects only events that exceed a *clipping level*, which is a predefined threshold for the event. The system ignores events until they reach this threshold.

For example, failed logon attempts are common in any system, since users can easily enter the wrong password once or twice. Instead of raising an alarm for every single failed logon attempt, a clipping level can be set to raise an alarm only if it detects five failed logon attempts within a 30-minute period. Many account lockout controls use a similar clipping level. They don't lock the account after a single failed logon. Instead, they count the failed logons and lock the account only when the predefined threshold is reached.

Clipping levels are widely used in the process of auditing events to establish a baseline of routine system or user activity. The monitoring system raises an alarm to signal abnormal events only if the baseline is exceeded. In other words, the clipping level causes the system to ignore routine events and only raise an alert when it detects serious intrusion patterns.

In general, nonstatistical sampling is discretionary sampling, or sampling at the auditor's discretion. It doesn't offer an accurate representation of the whole body of data and will ignore events that don't reach the clipping level threshold. However, it is effective when used to focus on specific events. Additionally, nonstatistical sampling is less expensive and easier to implement than statistical sampling.

NOTE Both statistical and nonstatistical sampling are valid mechanisms to create summaries or overviews of large bodies of audit data. However, statistical sampling is more reliable and mathematically defensible.

Other Monitoring Tools

Although logs are the primary tools used for monitoring, some additional tools are used within organizations that are worth mentioning. For example, a CCTV system can automatically record events onto tape for later review. Security personnel can also watch a live CCTV system for unwanted, unauthorized, or illegal activities in real time. This system can work alone or in conjunction with security guards, who themselves can be monitored by the CCTV and held accountable for any illegal or unethical activity. Other tools include the following:

Keystroke Monitoring *Keystroke monitoring* is the act of recording the keystrokes a user performs on a physical keyboard. The monitoring is commonly done via technical means such as a hardware device or a software program known as a keylogger. However, a video recorder can perform visual monitoring. In most cases, attackers use keystroke monitoring for malicious purposes. In extreme circumstances and highly restricted environments, an organization might implement keystroke monitoring to monitor and analyze user activity.

Keystroke monitoring is often compared to wiretapping. There is some debate about whether keystroke monitoring should be restricted and controlled in the same manner as telephone wiretaps. Many organizations that employ keystroke monitoring notify both authorized and unauthorized users of such monitoring through employment agreements, security policies, or warning banners at sign-on or login areas.

> NOTE Companies can and do use keystroke monitoring in some situations. However, in almost all cases, they are required to inform employees of the monitoring.

Traffic Analysis and Trend Analysis *Traffic analysis* and *trend analysis* are forms of monitoring that examine the flow of packets rather than actual packet contents. These processes are sometimes referred to as *network flow monitoring*. It can infer a lot of information, such as primary and backup communication routes, the location of primary servers, sources of encrypted traffic and the amount of traffic supported by the network, typical direction of traffic flow, frequency of communications, and much more.

These techniques can sometimes reveal questionable traffic patterns, such as when an employee's account sends a massive amount of email to others. This might indicate the employee's system is part of a botnet controlled by an attacker at a remote location. Similarly, traffic analysis might detect if an unscrupulous insider forwards internal information to unauthorized parties via email. These types of events often leave detectable signatures.

Log Management

Log management refers to all the methods used to collect, process, and protect log entries. As discussed previously, a SIEM system collects and aggregates log entries from multiple systems. It then analyzes these entries and reports any suspicious events.

After a system forwards log entries to a SIEM system, it's acceptable to delete the log entries. However, these usually aren't deleted from the original system right away. Instead, systems typically use *rollover logging*, sometimes called circular logging or log cycling. Rollover logging allows administrators to set a maximum log size. When the log reaches that size, the system begins overwriting the oldest events in the log.

Windows systems allow administrators to archive logs, which is useful if a SIEM system isn't available. When the option to archive logs is selected and the log reaches the maximum size, the system will save the log as a new file and start a new log. The danger here is that the system disk drive could fill with these archived log files.

Another option is to create and schedule a PowerShell script to regularly archive the files and copy them to another location, such as a backup server using a UNC path. The key is to implement a method that will save the log entries and prevent the logs from filling a disk drive.

Egress Monitoring

Monitoring traffic isn't limited to traffic within a network or entering a network. It's also important to monitor traffic leaving a network to the Internet, also called egress monitoring. This can detect the unauthorized transfer of data outside the organization, often referred to as data exfiltration. Some common methods used to detect or prevent data exfiltration are data loss prevention (DLP) techniques and monitoring for steganography.

> Chapter 7 covers steganography and watermarking in more depth and Chapter 5, "Protecting Security of Assets," covers DLP in more depth.

Steganography allows attackers to embed messages within other files such as graphic or audio files. It is possible to detect steganography attempts if you have both the original file and a file you suspect has a hidden message. If you use a hashing algorithm such as Secure Hash Algorithm 3 (SHA-3), you can create a hash of both files. If the hashes are the same, the file does not have a hidden message. However, if the hashes are different, it indicates the second file has been modified. Forensic analysis techniques might be able to retrieve the message.

An organization can periodically capture hashes of internal files that rarely change. For example, graphics files such as JPEG and GIF files generally stay the same. Imagine security experts suspect that a malicious insider is embedding additional data within these files and emailing them outside the organization. In that case, they can compare the original hashes with the hashes of the files the malicious insider sent out. If the hashes are different, it indicates the files are different and may contain hidden messages.

An advanced implementation of watermarking is digital watermarking. A *digital watermark* is a secretly embedded marker in a digital file. For example, some movie studios digitally mark copies of movies sent to different distributors. Each copy has a different mark, and the studios track which distributor received which copy. If any of the distributors release pirated copies of the movie, the studio can identify which distributor did so.

DLP systems can detect watermarks in unencrypted files. When a DLP system identifies sensitive data from these watermarks, it can block the transmission and raise an alert for security personnel. This prevents the transmission of the files outside the organization.

Advanced attackers, such as advanced persistent threats sponsored by nation-states, commonly encrypt data before sending it out of the network. This can thwart some common tools that attempt to detect data exfiltration. Although a DLP system can't examine content from encrypted data, it can monitor the volume of encrypted data going out of a network, where it's going, and which system sent it. Administrators can configure DLP systems to look for abnormalities related to encrypted traffic, such as an increase in volume.

However, it's also possible to include tools that monitor the amount of encrypted data sent out of the network.

Automating Incident Response

Incident response automation has improved considerably over the years, and it continues to improve. The following sections describe some of these improvements, such as security orchestration, automation, and response (SOAR), artificial intelligence (AI), and threat intelligence techniques.

Understanding SOAR

Security orchestration, automation, and response (SOAR) refers to a group of technologies that allow organizations to respond to some incidents automatically. Organizations have a variety of tools that warn about potential incidents. Traditionally, security administrators respond to each warning manually. This typically requires them to verify the warning is valid and then respond. Many times, they perform the same rote actions that they've done before.

As an example, imagine attackers have launched a SYN flood attack on servers in a screened subnet (sometimes referred to as a demilitarized zone). Network tools detect the attack and raise alerts. The organization has policies in place where security administrators verify the alerts are valid. If so, they manually change the amount of time a server will wait for an ACK packet. After the attack has stopped, they manually change the time back to its original setting.

Depending on the event, it can raise an alarm for an administrator or take some other action. For example, if an email server's send queue starts backing up, a SIEM application can detect the issue and alert administrators before the problem is serious.

SOAR allows security administrators to define these incidents and the response, typically using playbooks and runbooks:

Playbook A playbook is a document or checklist that defines how to verify an incident. Additionally, it gives details on the response. A playbook for the SYN flood attack would list the same actions security administrators take to verify a SYN flood is under way. It would also list the steps administrators take after verifying it is a SYN flood attack.

Runbook A runbook implements the playbook data into an automated tool. For example, if an IDS alerts on the traffic, it implements a set of conditional steps to verify that the traffic is a SYN flood attack using the playbook's criteria. If the IDS confirms the attack, it then performs specified actions to mitigate the threat.

> NOTE It's worth noting that there aren't definitive definitions of a playbook and a runbook that all companies use. For example, some BCP experts say that a runbook refers to computers and networks, whereas a playbook refers to the business in general. However, within the context of incident response, a playbook is a document that defines actions, and the runbook implements those actions.

This scenario shows a single attack and response, but SOAR technologies can respond to any attacks. The hard part is documenting all known incidents and responses in the playbooks and then configuring tools to respond automatically.

It's important to realize that the playbooks' primary purpose is to document what the runbooks should do. However, playbooks can be used as a manual backup if the SOAR system fails. In other words, if a runbook fails to run after an incident, administrators can still refer to the playbook to complete the steps manually.

Machine Learning and AI Tools

Many companies (especially those with something to sell) use the terms artificial intelligence (AI) and machine learning (ML) interchangeably, as though they are synonymous. However, they aren't. Unfortunately, there aren't strict definitions of these terms that everyone agrees on and follows. Marketers may use them synonymously. Scientists creating ML and AI systems have much more complex definitions that have morphed over time. However, the following bullets provide general descriptions of the term:

- Machine learning is a part of artificial intelligence and refers to a system that can improve automatically through experience. ML gives computer systems the ability to learn.
- Artificial intelligence is a broad field that includes ML. It gives machines the ability to do things that a human can do better or allows a machine to perform tasks that we previously thought required human intelligence. This is a moving target, though. The idea of a car parking itself or coming to you from a parking spot was once thought to require human intelligence. Cars can now do these tasks without human interaction.

A key point is that machine learning is a part of the broad topic of AI. From a simple perspective, consider machine learning and AI applied to the game of Go.

A machine-learning algorithm will outline the rules of the game, such as how the pieces move, legal moves, and what a win looks like. The machine will use these rules to play games against itself repeatedly. With each game, it adds to its experience level, and it progressively gets better and better. Over time, it learns what strategies work and what strategies don't work.

In contrast, an AI system starts with zero knowledge of the game. It doesn't know how the pieces move, what moves are legal, or even what a win looks like. However, a separate algorithm outside of the AI system enforces the rules. It tells the AI system when it makes an illegal move and when it wins or loses a game. The AI system uses this feedback to create its own algorithms as it is learning the rules. As it creates these algorithms, it applies machine-learning techniques to teach itself winning strategies.

These two examples demonstrate the major difference between machine learning and AI. A machine-learning system (part of AI) starts with a set of rules or guidelines. An AI system starts with nothing and progressively learns the rules. It then creates its own algorithms as it learns the rules and applies machine-learning techniques based on these rules.

Think of a behavior-based detection system as one way machine learning and artificial intelligence can apply to cybersecurity. As a reminder, administrators need to create a baseline of normal activities and traffic on a network. If the network is modified, administrators need to re-create the baseline. In this case, the baseline is similar to a set of rules given to a machine-learning system.

A machine-learning system would use this baseline as a starting point. During normal operation, it detects anomalies and reports them. If an administrator investigates and reports it as a false positive, the machine-learning system learns from this feedback. It modifies the initial baseline based on feedback it receives about valid alarms and false positives.

An AI system starts without a baseline. Instead, it monitors traffic and slowly creates its own baseline based on the traffic it observes. As it creates the baseline, it also looks for anomalies. An AI system also relies on feedback from administrators to learn if alarms are valid or false positives.

Threat Intelligence

Threat intelligence refers to gathering data on potential threats. It includes using various sources to get timely information on current threats. Many organizations used it to hunt out threats.

Understanding the Kill Chain

The military has used a kill chain model to disrupt attacks for decades. The military model has a lot of depth, but in short, it includes the following phases:

1. Find or identify a target through reconnaissance.
2. Get the target's location.
3. Track the target's movement.
4. Select a weapon to use on the target.
5. Engage the target with the selected weapon.
6. Evaluate the effectiveness of the attack.

It's important to know that the military uses this model for both offense and defense. When attacking, they go through each of the phases as an ordered chain of events. However, they know that the enemy is likely using a similar model, so they attempt to break the chain. If the attacker fails at any stage of the attack chain, the attack will not succeed.

Several organizations have adapted the military kill chain to create cyber kill chain models. For example, Lockheed Martin created the Cyber Kill Chain Framework. It includes seven ordered stages of an attack:

Reconnaissance Attackers gather information on the target.

Weaponization Attackers identify an exploit that the target is vulnerable to, along with methods to send the exploit.

Delivery Attackers send the weapon to the target via phishing attacks, malicious email attachments, compromised websites, or other common social engineering methods.

Exploitation. The weapon exploits a vulnerability on the target system.

Installation. Code that exploits the vulnerability then installs malware. The malware typically includes a backdoor, allowing the target to access the system remotely.

Command and Control. Attackers maintain a command-and-control system, which controls the target and other compromised systems.

Actions on objectives. Attackers execute their original goals such as theft of money, theft of data, data destruction, or installing additional malicious code such as ransomware.

As with the military model, the goal is to disrupt the chain by stopping the attacker at any phase of the attack. As an example, if users avoid all the social engineering methods, the attacker can't deliver the weapon, and the attacker can't succeed.

Understanding the MITRE ATT&CK

The MITRE ATT&CK Matrix (created by MITRE and viewable at http://attack.mitre.org) is a knowledge base of identified tactics, techniques, and procedures (TTPs) used by malicious actors in various attacks. It is complementary to kill chain models, such as the Cyber Kill Chain. However, unlike kill chain models, the tactics are not an ordered set of attacks. Instead, MITRE ATT&CK lists the TTPs within a matrix. Additionally, malicious actors are constantly modifying their attack methods, so the ATT&CK Matrix is a living document that is updated at least twice a year.

The matrix includes the following tactics:

- Reconnaissance
- Resource development
- Initial access
- Execution
- Persistence

- Privilege escalation
- Defense evasion
- Credential access
- Discovery
- Lateral movement
- Collection
- Command and control
- Exfiltration
- Impact

Each of the tactics includes techniques used by attackers. For example, the Reconnaissance tactic consists of multiple techniques. Clicking any of these takes you to another page describing it, along with mitigation and detection techniques. Some techniques include layers of subtechniques. If you drill down on Reconnaissance, you'll see Vulnerability Scanning under Active Scanning. This documents specific things you can look for to detect unauthorized scans.

> NOTE Chapter 15 covers vulnerability scans and vulnerability scanners in more depth.

Threat Feeds

On the Internet, a feed is a steady stream of content that users can scroll through. Users can subscribe to various content, such as news articles, weather, blog content, and more. As an example, Really Simple Syndication (RSS) allows users to subscribe to different content, and a single aggregator collects the content and displays it to users.

A *threat feed* is a steady stream of raw data related to current and potential threats. However, in its raw form, it can be difficult to extract meaningful data. A threat intelligence feed attempts to extract actionable intelligence from the raw data. Here is some of the information included in a threat intelligence feed:

- Suspicious domains
- Known malware hashes
- Code shared on internet sites
- IP addresses linked to malicious activity

By comparing data in a threat feed with data going to and from the Internet, security experts can identify potentially malicious traffic. Imagine an attacker stands up a website and uses it to attempt drive-by downloads of new malware. If an organization detects this website's domain name (or IP address) in incoming or outgoing traffic, it is readily apparent that this is malicious and should be investigated.

Although it's possible to manually cross-check the data from a threat feed with logs tracking incoming and outgoing traffic, doing so can be quite tedious. Instead, many organizations use an additional tool to cross-check this data automatically.

Some security organizations sell platforms that integrate with threat feeds and automatically provide organizations with the data they need to respond quickly.

Threat Hunting

Threat hunting is the process of actively searching for cyberthreats in a network. This goes beyond waiting for traditional network tools to detect and report attacks. It starts with the premise that attackers are in the network now, even if none of the preventive and detection controls have detected them and raised warnings. Instead, security professionals aggressively search systems looking for indicators of threats.

As an example, imagine that a threat feed indicates that a botnet has been launching several DDoS attacks recently. It shows the TTPs commonly used to join computers to the botnet. More, it lists the specific things to look for to identify computers joined to this botnet. This might be the existence of specific files or log entries showing specific traffic into or out of the network. Once administrators know what to look for, it becomes a simple matter to craft scripts to look for these files on all internal computers or to send alerts for any network traffic with log entries matching the threat feed information.

Many years ago, attackers often caused damage almost immediately after entering a network. However, many attackers now attempt to remain in a network as long as possible. As an example, APTs often stay undetected in networks for months.

There isn't a single method used for threat hunting. However, many methods attempt to analyze the phases of an attack and then look for signs of the attack at individual phases. One popular method of threat hunting is to use a kill chain model.

The Intersection of SOAR, Machine Learning, AI, and Threat Feeds

These technologies are all advancing rapidly, and things are likely to continue improving. As they do so, it is important to see how these concepts are intertwined.

Think of SOAR technologies. These include playbooks that are the written guidelines administrators use to verify and respond to incidents. Personnel then implement these guidelines in runbooks that implement the guidelines. Strictly speaking, these are not using machine learning or AI because someone must implement the guidelines, and the systems don't deviate from these rules. However, computers are great at performing repetitive steps and eliminating human errors, so they are welcomed by most administrators.

IDPSs often send out false positives (an alert indicating a problem where none exists). After implementing SOAR technologies, they will automatically deal with these false positives using the same guidelines documented in the playbook. Of course, the danger arises when an IDPS has false negatives (indicating a problem that has gone undetected by the IDPS). One way to avoid this is to keep IDPSs informed of new threats.

Enter threat feeds. If the SOAR technologies can receive and process the threat feeds, they can ensure all prevention and detection systems know about new threats and automatically respond to them. Compatible threat feeds can keep systems updated in real time. When a threat feed reports a suspicious domain (website), firewalls can immediately block access to it. When new malware hashes are known, IDPSs can monitor incoming traffic looking for these hashes.

Many companies claim that their security solutions leverage machine learning and AI. However, many of their methods are proprietary, so we can't see them. It could be that their systems are using these advanced techniques. They could also have a team of dedicated professionals working around the clock, identifying threats and manually creating runbooks to detect and mitigate the threats. Either way, SOAR technologies are constantly improving and reducing the workload of administrators.

Summary

The CISSP Security Operations domain lists several specific incident management steps. Detection is the first step and can come from automated tools or employee observations. Personnel investigate alerts to determine whether an actual incident has occurred, and if so, the next step is a response. Containment of the incident is essential during the mitigation stage. It's also important to protect any evidence during all stages of incident management. Reporting may be required based on governing laws or an organization's security policy. In the recovery stage, the system is restored to full operation, and it's important to ensure that it is restored to at least as secure a state as it was before the attack. The remediation stage includes a root cause analysis and will often include recommendations to prevent a reoccurrence. Last, the lessons learned stage examines the incident and the response to determine whether there are any lessons to be learned.

Preventive and detection measures help prevent security incidents and detect them if they occur. This includes basic preventive measures such as keeping systems and applications up-to-date with current patches, removing or disabling unneeded services and protocols, using intrusion detection and prevention systems, using anti-malware software with up-to-date signatures, and enabling both host-based and network-based firewalls. It also includes using advanced tools such as intrusion detection and prevention systems, honeypots, and honeynets.

Logging and monitoring provide overall accountability when combined with effective identification and authentication practices. Logging involves recording events in logs and database files. Security logs, system logs, application logs, firewall logs, proxy logs, and change management logs are all common log files. Log files include valuable data and should be protected to ensure that they aren't modified, deleted, or corrupted. If they are not protected, attackers will often try to modify or delete them, and they will not be admissible as evidence to prosecute an attacker.

Automating incident response techniques helps reduce the workload of administrators. These include the use of SOAR technologies, along with machine learning and automated intelligence tools. Using threat intelligence helps find threats within a network before traditional security tools locate them.

Study Essentials

List and describe incident management steps. The CISSP Security Operations domain lists incident management steps as detection, response, mitigation, reporting, recovery, remediation, and lessons learned. After detecting and verifying an incident, the first response is to limit or contain the scope of the incident while protecting evidence. Based on governing laws, an organization may need to report an incident to official authorities, and if PII is affected, individuals need to be informed. The remediation and lessons learned stages include root cause analysis to determine the cause and recommend solutions to prevent a reoccurrence.

Understand basic preventive measures. Basic preventive measures can prevent many incidents from occurring. These include keeping systems up-to-date, removing or disabling unneeded protocols and services, using intrusion detection and prevention systems, using anti-malware software with up-to-date signatures, and enabling both host-based and network-based firewalls.

Know the difference between whitelisting and blacklisting. Software whitelists provide a list of approved software and prevent the installation of any other software not on the list. Blacklists provide a list of unapproved software and prevent the installation of any software on the list.

Understand sandboxing. Sandboxing provides an isolated environment and prevents code running in a sandbox from interacting with elements outside of a sandbox.

Know about third-party provided security services. Third-party security services help an organization augment security services provided by internal employees. Many organizations use cloud-based solutions to augment their internal security.

Know about denial-of-service (DoS) attacks. DoS attacks prevent a system from responding to legitimate requests for service. A common DoS attack is the SYN flood attack, which disrupts the TCP three-way handshake. Even though older attacks are not as common today because basic precautions block them, you still need to know them because many newer attacks are often variations on older methods. Smurf attacks employ an amplification network to send numerous response packets to a victim. Ping-of-death attacks send numerous oversized ping packets to the victim, causing the victim to freeze, crash, or reboot.

Understand zero-day exploits. A zero-day exploit is an attack that uses a vulnerability that is either unknown to anyone but the attacker or known only to a limited group of people. On the surface, it sounds like you can't protect against an unknown vulnerability, but basic security practices go a long way toward preventing zero-day exploits. Removing or disabling unneeded protocols and services reduces the attack surface, enabling firewalls to block many access points, and using intrusion detection and prevention systems helps detect and block potential attacks. Additionally, using tools such as honeypots helps protect live networks.

Understand man-in-the-middle attacks. A man-in-the-middle attack (sometimes called an on-path attack) occurs when a malicious user is able to gain a logical position between the two endpoints of a communications link. Although it takes a significant amount of sophistication on the part of an attacker to complete a man-in-the middle attack, the amount of data obtained from the attack can be significant.

Understand intrusion detection and intrusion prevention. IDSs and IPSs are important detection and preventive measures against attacks. Know the difference between knowledge-based detection (using a database similar to anti-malware signatures) and behavior-based detection. Behavior-based detection starts with a baseline to recognize normal behavior and compares activity with the baseline to detect abnormal activity. The baseline can be outdated if the network is modified, so it must be updated when the environment changes.

Describe honeypots and honeynets. A honeypot is a system that typically has pseudo flaws and fake data to lure intruders. A honeynet is two or more honeypots in a network. Administrators can observe attackers' activity while they are in the honeypot, and as long as attackers are in the honeypot, they are not in the live network.

Understand the methods used to block malicious code. Malicious code is thwarted with a combination of tools. The obvious tool is anti-malware software with up-to-date definitions installed on each system, at the boundary of the network, and on email servers. However, policies that enforce basic security principles, such as the least privilege principle, prevent regular users from installing potentially malicious software. Additionally, educating users about the risks and the methods attackers commonly use to spread viruses helps users understand and avoid dangerous behaviors.

Know the types of log files. Log data is recorded in databases and different types of log files. Common log files include security logs, system logs, application logs, firewall logs, proxy logs, and change management logs. Log files should be protected by centrally storing them and using permissions to restrict access, and archived logs should be set to read-only to prevent modifications.

Understand monitoring and uses of monitoring tools. Monitoring is a form of auditing that focuses on active review of the log file data. Monitoring is used to hold subjects accountable for their actions and to detect abnormal or malicious activities. It is also used to monitor system performance. Monitoring tools such as IDSs or SIEMs automate continuous monitoring and provide real-time analysis of events, including monitoring what happens inside a network, traffic entering a network, and traffic leaving a network (also known as egress monitoring). Log management includes analyzing logs and archiving logs.

Be able to explain audit trails. Audit trails are the records created by recording information about events and occurrences into one or more databases or log files. They are used to reconstruct an event, extract information about an incident, and prove or disprove culpability. Using audit trails is a passive form of detection security control, and audit trails are essential evidence in criminals' prosecution.

Understand how to maintain accountability. Accountability is maintained for individual subjects through the use of auditing. Logs record user activities and users can be held accountable for their logged actions. This directly promotes good user behavior and compliance with the organization's security policy.

Describe threat feeds and threat hunting. Threat feeds provide organizations with a steady stream of raw data. By analyzing threat feeds, security administrators can learn of current threats. They can then use this knowledge to search through the network, looking for signs of these threats.

Know the benefits of SOAR. SOAR technologies automate responses to incidents. One of the primary benefits is that this reduces the workload of administrators. It also removes the possibility of human error by having computer systems respond.

Written Lab

1. Define an incident.
2. List the different phases of incident management identified in the CISSP Security Operations domain.
3. Describe the primary types of intrusion detection systems.
4. Discuss the benefits of a SIEM system.
5. Describe the purpose of SOAR technologies.

Review Questions

1. Which of the following are valid incident management steps or phases as listed in the CISSP objectives? (Choose all that apply.)

    A. Prevention

    B. Detection

    C. Reporting

    D. Lessons learned

    E. Backup

2. A technician is troubleshooting a problem on a user's computer. After viewing the host-based intrusion detection system (HIDS) logs, he determines that the computer has been compromised by malware. Of the following choices, what should he do next?

    A. Isolate the computer from the network.

    B. Review the HIDS logs of neighboring computers.

    C. Run an antivirus scan.