Chapter 1

Penetration Testing

THE COMPTIA PENTEST+ EXAM OBJECTIVES COVERED IN THIS CHAPTER INCLUDE:

Domain 1: Planning and Scoping

✓ **1.3 Given a scenario, demonstrate an ethical hacking mindset by maintaining professionalism and integrity.**

- Background checks of penetration testing team
- Adhere to specific scope of engagement
- Identify criminal activity
- Immediately report breaches/criminal activity
- Limit the use of tools to a particular engagement
- Limit invasiveness based on scope
- Maintain confidentiality of data/information
- Risks to the professional

Hackers employ a wide variety of tools to gain unauthorized access to systems, networks, and information. Automated tools, including network scanners, software debuggers, password crackers, exploitation frameworks, and malware, do play an important role in the attacker's toolkit. Cybersecurity professionals defending against attacks should have access to the same tools in order to identify weaknesses in their own defenses that an attacker might exploit.

These automated tools are not, however, the most important tools at a hacker's disposal. The most important tool used by attackers is something that cybersecurity professionals can't download or purchase. It's the power and creativity of the human mind. Skilled attackers leverage quite a few automated tools as they seek to defeat cybersecurity defenses, but the true test of their ability is how well they are able to synthesize the information provided by those tools and pinpoint potential weaknesses in an organization's cybersecurity defenses.

What Is Penetration Testing?

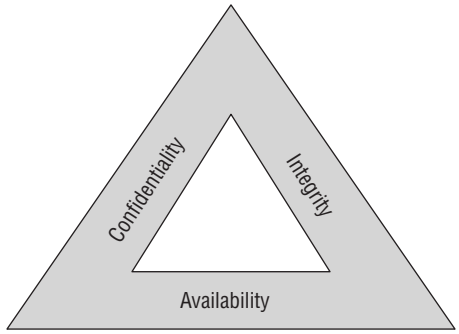
Penetration testing seeks to bridge the gap between the rote use of technical tools to test an organization's security and the power of those tools when placed in the hands of a skilled and determined attacker. Penetration tests are authorized, legal attempts to defeat an organization's security controls and perform unauthorized activities. The tests are time-consuming and require staff who are as skilled and determined as the real-world attackers who will attempt to compromise the organization. However, they're also the most effective way for an organization to gain a complete picture of its security vulnerability.

Cybersecurity Goals

Cybersecurity professionals use a well-known model to describe the goals of information security. The CIA triad, shown in Figure 1.1, includes the three main characteristics of information that cybersecurity programs seek to protect:

- *Confidentiality* measures seek to prevent unauthorized access to information or systems.
- *Integrity* measures seek to prevent unauthorized modification of information or systems.
- *Availability* measures seek to ensure that legitimate use of information and systems remains possible.

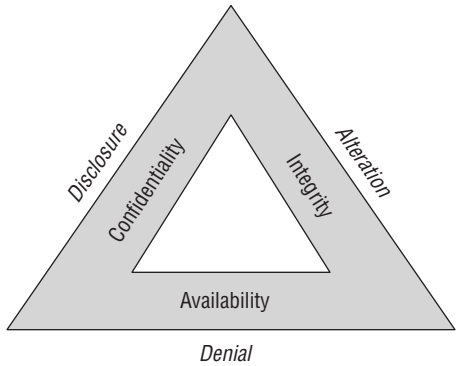
FIGURE 1.1 The CIA triad



Attackers, and therefore penetration testers, seek to undermine these goals and achieve three corresponding goals of their own. The attackers' goals are known as the DAD triad, shown in Figure 1.2:

- *Disclosure* attacks seek to gain unauthorized access to information or systems.
- *Alteration* attacks seek to make unauthorized changes to information or systems.
- *Denial* attacks seek to prevent legitimate use of information and systems.

FIGURE 1.2 The DAD triad



These two models, the CIA and DAD triads, are the cornerstones of cybersecurity. As shown in Figure 1.2, the elements of both models are directly correlated, with each leg of the attackers' DAD triad directly corresponding to a leg of the CIA triad that is designed to counter those attacks. Confidentiality controls seek to prevent disclosure attacks. Integrity controls seek to prevent alteration attacks. Availability controls seek to keep systems running, preventing denial attacks.

Adopting the Hacker Mindset

If you've been practicing cybersecurity for some time, you're probably intimately familiar with the elements of the CIA triad. Cybersecurity defenders spend the majority of their time thinking in these terms, designing controls and defenses to protect information and systems against a wide array of known and unknown threats.

Penetration testers must take a very different approach in their thinking. Instead of trying to defend against all possible threats, they only need to find a single vulnerability that they might exploit to achieve their goals. To find these flaws, they must think like the adversary who might attack the system in the real world. This approach is commonly known as adopting the *hacker mindset*.

Before we explore the hacker mindset in terms of technical systems, let's explore it using an example from the physical world. If you were responsible for the physical security of an electronics store, you might consider a variety of threats and implement controls designed to counter those threats. You'd be worried about shoplifting, robbery, and employee embezzlement, among other threats, and you might build a system of security controls that seeks to prevent those threats from materializing. These controls might include the following items:

- Security cameras in high-risk areas
- Auditing of cash register receipts
- Theft detectors at the main entrance/exit of the store
- Exit alarms on emergency exits
- Burglar alarm wired to detect the opening of doors outside of business hours

Now, imagine that you've been engaged to conduct a security assessment of this store. You'd likely examine each one of these security controls and assess its ability to prevent each of the threats identified in your initial risk assessment. You'd also look for gaps in the existing security controls that might require supplementation. Your mandate is broad and high-level.

Penetration tests, on the other hand, have a much more focused mandate. Instead of adopting the approach of a security professional, you adopt the mindset of an attacker. You don't need to evaluate the effectiveness of each security control. You simply need to find either one flaw in the existing controls or one scenario that was overlooked in planning those controls.

In this example, a penetration tester might enter the store during business hours and conduct reconnaissance, gathering information about the security controls that are in place and the locations of critical merchandise. They might notice that although the burglar alarm is tied to the doors, it does not include any sensors on the windows. The tester might then return in the middle of the night, smash a window, and grab valuable merchandise. Recognizing that the store has security cameras in place, the attacker might wear a mask and park a vehicle outside of the range of the cameras. That's the hacker mindset. You need to think like a criminal.

There's an important corollary to the hacker mindset that is important for both attackers and defenders to keep in mind. When conducting a penetration test (or a real-world attack), the attacker needs to win only once. They might attempt hundreds or thousands of potential

attacks against a target. The fact that an organization's security defenses block 99.99 percent of those attacks is irrelevant if one of the attacks succeeds. Cybersecurity professionals need to win *every* time. Attackers need to win only once.

Ethical Hacking

While penetration testers certainly must be able to adopt the hacker mindset, they must do so in a manner that demonstrates their own professionalism and integrity. *Ethical hacking* is the art of using hacking tools and techniques but doing so within a code of ethics that regulates activity. Some of the key components of ethical hacking programs are:

- Performing background checks on all members of the penetration testing team to identify and resolve any potential issues
- Adhering to the defined scope of a penetration testing engagement
- Immediately reporting any active security breaches or criminal activity detected during a penetration test
- Limiting the use of penetration testing tools to approved engagements
- Limiting the invasiveness of a penetration test based on the scope of the engagement
- Protecting the confidentiality of data and information related to or uncovered during a penetration test

Cybersecurity professionals engaged in penetration testing work that exceeds the bounds of ethical hacking may find themselves subject to fees, fines, or even criminal charges depending on the nature of the violation.

Reasons for Penetration Testing

The modern organization dedicates extensive time, energy, and funding to a wide variety of security controls and activities. We install firewalls, intrusion prevention systems, security information and event management devices, vulnerability scanners, and many other tools. We equip and staff 24-hour security operations centers (SOCs) to monitor those technologies and watch our systems, networks, and applications for signs of compromise. There's more than enough work to completely fill our days twice over. Why on Earth would we want to take on the additional burden of performing penetration tests? After all, they are time-consuming to perform internally and expensive to outsource.

The answer to this question is that penetration testing provides us with visibility into the organization's security posture that simply isn't available by other means. Penetration testing does not seek to replace all of the other cybersecurity activities of the organization. Instead, it complements and builds on those efforts. Penetration testers bring their unique skills and perspective to the table and can take the outputs of security tools and place them within the attacker's mindset, asking the question, "If I were an attacker, how could I use this information to my advantage?"

Benefits of Penetration Testing

We've already discussed *how* penetration testers carry out their work at a high level, and the remainder of this book is dedicated to exploring penetration testing tools and techniques in detail. Before we dive into that, let's take a moment to consider *why* we conduct penetration testing. What benefits does it bring to the organization?

First and foremost, penetration testing provides us with knowledge that we can't obtain elsewhere. By conducting thorough penetration tests, we learn whether an attacker with the same knowledge, skills, and information as our testers would likely be able to penetrate our defenses. If they can't gain a foothold, we can then be reasonably confident that our networks are secure against attack by an equivalently talented attacker under the present circumstances.

Second, in the event that attackers are successful, penetration testing provides us with an important blueprint for remediation. As cybersecurity professionals, we can trace the actions of the testers as they progressed through the different stages of the attack and close the series of open doors the testers passed through. Doing so provides us with a more robust defense against future attacks.

Finally, penetration tests can provide us with essential, focused information about specific attack targets. We might conduct a penetration test prior to the deployment of a new system that is specifically focused on exercising the security features of that new environment. Unlike an open-ended penetration test, which is broad in nature, focused tests can drill into the defenses around a specific target and provide actionable insight that can prevent a vulnerability from initial exposure.

Threat Hunting

The discipline of *threat hunting* is closely related to penetration testing but has a separate and distinct purpose. Like penetration testers, cybersecurity professionals engaged in threat hunting seek to adopt the attacker's mindset and imagine how hackers might seek to defeat an organization's security controls. The two disciplines diverge in what they accomplish with this information.

Penetration testers seek to evaluate the organization's security controls by testing them in the same manner an attacker might, whereas threat hunters use the attacker mindset to search the organization's technology infrastructure for the artifacts of a successful attack. They ask themselves what a hacker might do and what type of evidence they might leave behind and then go in search of that evidence.

Threat hunting builds on a cybersecurity philosophy known as the *presumption of compromise*. This approach assumes that attackers have already successfully breached an organization and searches out the evidence of successful attacks. When threat hunters discover a potential compromise, they then kick into incident-handling mode, seeking to contain, eradicate, and recover from the compromise. They also conduct a postmortem analysis of the factors that contributed to the compromise in an effort to remediate deficiencies. This post-event remediation is another similarity between penetration testing and threat hunting: organizations leverage the output of both processes in similar ways.

Regulatory Requirements for Penetration Testing

There is one last reason that you might conduct a penetration test—because you must! The most common regulatory requirement for penetration testing comes from the Payment Card Industry Data Security Standard (PCI DSS). This regulation is a private contractual obligation that governs all organizations involved in the storage, processing, or transmission of credit and debit card transactions. Nestled among the more than 100 pages of detailed security requirements for cardholder data environments (CDEs) is section 11.3, which reads as follows:

> Implement a methodology for penetration testing that includes the following:
>
> - Is based on industry accepted penetration testing approaches (for example, NIST SP800-115)
> - Includes coverage for the entire CDE perimeter and critical systems
> - Includes testing from both inside and outside the network
> - Includes testing to validate any segmentation and scope-reduction controls
> - Defines application-layer penetration tests to include, at a minimum, the vulnerabilities listed in Requirement 6.5
> - Defines network-layer penetration tests to include components that support network functions as well as operating systems
> - Includes review and consideration of threats and vulnerabilities experienced in the last 12 months
> - Specifies retention of penetration testing results and remediation activities results
>
> Source: Payment Card Industry Data Security Standard Version 3.2

Requirement 6.5 includes a listing of common vulnerabilities, such as SQL injection, buffer overflow, insecure cryptographic storage, insecure communications, improper error handling, cross-site scripting, improper access controls, cross-site request forgery, broken authentication, and other "high-risk" vulnerabilities.

That section of PCI DSS provides a useful set of requirements for anyone conducting a penetration test. It's also a nice blueprint for penetration testing, even for organizations that don't have PCI DSS compliance obligations.

The standard goes on to include four additional requirements that describe the frequency and scope of penetration tests:

> 11.3.1. Perform *external* penetration testing at least annually and after any significant infrastructure or application upgrade or modification (such as an operating system upgrade, a sub-network added to the environment, or a web server added to the environment).

> 11.3.2 Perform *internal* penetration testing at least annually and after any significant infrastructure or application upgrade or modification (such as an operating system upgrade, a sub-network added to the environment, or a web server added to the environment).
>
> 11.3.3. Exploitable vulnerabilities found during penetration testing are corrected and the testing is repeated to verify the corrections.
>
> 11.3.4 If segmentation is used to isolate the CDE from other networks, perform penetration tests at least annually and after any changes to segmentation controls/methods to verify that the segmentation methods are operational and effective, and isolate all out-of-scope systems from systems in the CDE.

Again, though these requirements are only mandatory for organizations subject to PCI DSS, they provide an excellent framework for any organization attempting to plan the frequency and scope of their own penetration tests. We'll cover compliance requirements for penetration testing in greater detail in Chapter 2, "Planning and Scoping Penetration Tests."

Organizations that must comply with PCI DSS should also read the detailed *Information Supplement: Penetration Testing Guidance* available from the PCI Security Standards Council at `www.pcisecuritystandards.org/documents/Penetration-Testing-Guidance-v1_1.pdf`. This document covers in great detail how organizations should interpret these requirements.

Who Performs Penetration Tests?

Penetration testing is a highly skilled discipline, and organizations often try to have experienced penetration testers for their testing efforts. Given that you're reading this book and are preparing for the PenTest+ certification, you likely already understand and recognize this.

If you don't have experience conducting penetration tests, that doesn't mean that all hope is lost. You may be able to participate in a test under the mentorship of an experienced penetration tester, or you may be able to conduct penetration testing in your organization simply because there's nobody with experience available to conduct the test.

Penetration tests may be conducted by either internal teams, consisting of cybersecurity employees from the organization being tested, or external teams, consisting of contractors.

Internal Penetration Testing Teams

Internal penetration testing teams consist of cybersecurity professionals from within the organization who conduct penetration tests on the organization's systems and applications. These teams may be dedicated to penetration testing on a full-time basis or they may be convened periodically to conduct tests on a part-time basis.

There are two major benefits of using internal teams to conduct penetration testing. First, they have contextual knowledge of the organization that can improve the effectiveness of testing by providing enhanced subject matter expertise. Second, it's generally less expensive to conduct testing using internal employees than it is to hire a penetration testing firm, provided that you have enough work to keep your internal team busy!

The primary disadvantages to using internal teams to conduct penetration testing stem from the fact that you are using internal employees. These individuals may have helped to design and implement the security controls that they are testing, which may introduce conscious or unconscious bias toward demonstrating that those controls are secure. Similarly, the fact that they were involved in designing the controls may make it more difficult for them to spot potential flaws that could provide a foothold for an attacker.

There's a bit of tricky language surrounding the use of the words *internal* and *external* when it comes to penetration tests. If you see these words used on the exam (or in real life!), be sure that you understand the context. Internal penetration tests may refer either to tests conducted by internal teams (as described in this section) or to tests conducted from an internal network perspective. The latter tests are designed to show what activity a malicious insider could engage in and may be conducted by either internal or external teams. Similarly, an external penetration test may refer to a test that is conducted by an external team or a test that is conducted from an external network perspective.

If you do choose to use an internal penetration testing team, it is important to recognize that team members might be limited by a lack of independence. If at all possible, the penetration testing team should be organizationally separate from the cybersecurity team that designs and operates controls. However, this is usually not possible in any but the largest organizations due to staffing constraints.

External Penetration Testing Teams

External penetration testing teams are hired for the express purpose of performing a penetration test. They may come from a general cybersecurity consulting firm or one that specializes in penetration testing. These individuals are usually highly skilled at conducting penetration tests because they perform these tests all day, every day. When you hire a professional penetration testing team, you generally benefit from the use of very talented attackers.

If you are subject to regulatory requirements that include penetration testing, be sure to understand how those requirements impact your selection of a testing team.

External penetration testing teams also generally bring a much higher degree of independence than internal teams. However, organizations using an external team should still be aware of any potential conflicts of interest the testers may have. It might not be the

best idea to hire the cybersecurity consultants that helped you design and implement your security controls to perform an independent test of those controls. They may be inclined to feel that any negative report they provide is a reflection on the quality of their own work.

Selecting Penetration Testing Teams

Penetration testing is not a one-time process. Organizations may wish to require penetration testing for new systems upon deployment, but it is important to repeat those tests on a periodic basis for three reasons.

First, the technology environment changes. Systems are reconfigured, patches are applied, updates and tweaks are made on a regular basis. Considered in isolation, each of these changes may have only a minor impact on the environment and may not reach the threshold for triggering a "significant change" penetration test, but collectively they may change the security posture of the environment. Periodic penetration tests have a good chance of detecting security issues introduced by those environmental changes.

Second, attack techniques evolve over time as well, and updated penetration tests should reflect changing attack techniques. A system developed and tested today may receive a clean bill of health, but the exact same system tested two years from now may be vulnerable to an attack technique that simply wasn't known at the time of the initial test.

Finally, each team member brings a unique set of skills, talents, and experiences to the table. Different team members may approach the test in different ways, and a team conducting a follow-on test differently may discover a vulnerability that went unnoticed by the initial team. To maximize your chances of discovering these issues, you should take care when you select the members of a penetration testing team. When possible, rotating team members so they are testing systems, environments, and applications that they have never tested before helps bring a fresh perspective to each round of penetration tests.

The CompTIA Penetration Testing Process

The CompTIA PenTest+ curriculum divides the penetration testing process into four stages, as shown in Figure 1.3.

FIGURE 1.3 CompTIA penetration testing stages

This process captures the major activities involved in conducting a penetration test and will be the way that we approach organizing the content in the remainder of this book.

If you look at CompTIA's PenTest+ Certification Exam Objectives document, you'll find that there are actually five domains of material covered by the exam. The four domains shown in Figure 1.3 each map to one of the stages of the penetration testing process. Domain 5 is titled "Tools and Code Analysis" and includes coverage of the many tools used during all stages of the penetration testing process.

Planning and Scoping

The military has a saying that resonates in the world of cybersecurity: "Prior planning prevents poor performance!" Although this sentiment is true for almost any line of work, it's especially important for penetration testing. Testers and their clients must have a clear understanding of what will occur during the penetration test, outline clear rules of engagement, and decide what systems, data, processes, and activities are within the authorized scope of the test. There's a fine line between penetration testing and hacking, and a written statement of work that includes clear authorization for penetration testing activities is crucial to ensuring that testers stay on the right side of the law and meet client expectations.

We cover this topic in great detail in Chapter 2. Specifically, you'll learn how to meet the three objectives of this domain:

- Compare and contrast governance, risk, and compliance concepts.
- Explain the importance of scoping and organizational/customer requirements.
- Given a scenario, demonstrate an ethical hacking mindset by maintaining professionalism and integrity.

Information Gathering and Vulnerability Scanning

Once a penetration testing team has a clearly defined scope and authorization to proceed with their work, they move on to the reconnaissance phase. During this stage, they gather as much information as possible about the target environment and perform testing designed to identify vulnerabilities in that environment.

This information-gathering process is crucial to the remainder of the penetration test, as the vulnerabilities identified during this stage provide the road map for the remainder of the test, highlighting weak links in an organization's security chain and potential paths of entry for attackers.

We cover information gathering and vulnerability scanning across four chapters of this book. In Chapter 3, "Information Gathering," you'll learn about the use of open source intelligence and the Nmap scanning tool. In Chapter 4, "Vulnerability Scanning," we begin a two-chapter deep dive into vulnerability scanning, perhaps the most important

information-gathering tool available to penetration testers. Chapter 4 covers how testers can design and perform vulnerability scans. In Chapter 5, "Analyzing Vulnerability Scans," we move on to the analysis of vulnerability reports and their application to the penetration testing process. Finally, in Chapter 6, "Exploiting and Pivoting," we discuss how to apply information learned during scans and exploit vulnerabilities. Together, these chapters cover the four objectives of this domain:

- Given a scenario, perform passive reconnaissance.
- Given a scenario, perform active reconnaissance.
- Given a scenario, analyze the results of a reconnaissance exercise.
- Given a scenario, perform vulnerability scanning.

As you plan your cybersecurity certification journey, you should know that there is significant overlap between the material covered in this domain and the material covered in Domain 2 (which is Vulnerability Management) of the Cybersecurity Analyst+ (CySA+) exam. There is also quite a bit of overlap between the basic security concepts and tools covered by both exams. If you successfully pass the PenTest+ exam, you might want to consider immediately moving on to the CySA+ exam because you'll already have mastered about a third of the material covered on that test.

Attacks and Exploits

After developing a clear testing plan and conducting reconnaissance activities, penetration testers finally get the opportunity to move on to what most of us consider the fun stuff! It's time to break out the white hat and attempt to exploit the vulnerabilities discovered during reconnaissance and penetrate an organization's network as deeply as possible, staying within the bounds established in the rules of engagement.

The specific attack techniques used during a penetration test will vary based on the nature of the environment and the scope agreed to by the client, but there are some common techniques used in most tests. Half of this book is dedicated to exploring each of those topics in detail.

In Chapter 6, "Exploiting and Pivoting," you'll learn how attackers establish a foothold on a network and then try to leverage that initial breach to gain as much access as possible. Chapter 7, "Exploiting Network Vulnerabilities," dives into attack techniques that focus on network devices and protocols. Chapter 9, "Exploiting Application Vulnerabilities," is about software attacks, and Chapter 10, "Attacking Hosts, Cloud Technologies, and Specialized Systems," examines issues on servers and endpoints. Chapter 8, "Exploiting Physical and Social Vulnerabilities," reminds us that many vulnerabilities aren't technical at all and that a penetration test that gains physical access to a facility or compromises members of an organization's staff can be even more dangerous than those that arrive over a network.

Combined, these chapters cover the seven objectives of this domain:

- Given a scenario, research attack vectors and perform network attacks.
- Given a scenario, research attack vectors and perform wireless attacks.
- Given a scenario, research attack vectors and perform application-based attacks.
- Given a scenario, research attack vectors and perform attacks on cloud technologies.
- Explain common attacks and vulnerabilities against specialized systems.
- Given a scenario, perform a social engineering or physical attack.
- Given a scenario, perform post-exploitation techniques.

Reporting and Communication

Once the glamor and excitement of the attack and exploitation phase passes, the work of the penetration testing team is not yet complete. A key requirement for a successful penetration test is that it provide useful information to the client about the security of their information technology environment. This should come in the form of clear, actionable recommendations for implementing new security controls and enhancing existing controls.

Chapter 11, "Reporting and Communication," explains the best practices for sharing penetration testing results with clients. Specifically, it covers the four objectives of this domain:

- Compare and contrast important components of written reports.
- Given a scenario, analyze the findings and recommend the appropriate remediation within a report.
- Explain the importance of communication during the penetration testing process.
- Explain post-report delivery activities.

Tools and Code Analysis

Chapter 12, "Scripting for Penetration Testing," covers a topic that's extremely important to penetration testers: applying coding skills to automate aspects of a penetration test. Although this chapter won't turn you into a software developer, it will introduce you to the analysis of basic penetration testing scripts written in Bash, Python, Ruby, and PowerShell. This chapter covers two of the objectives for Domain 5:

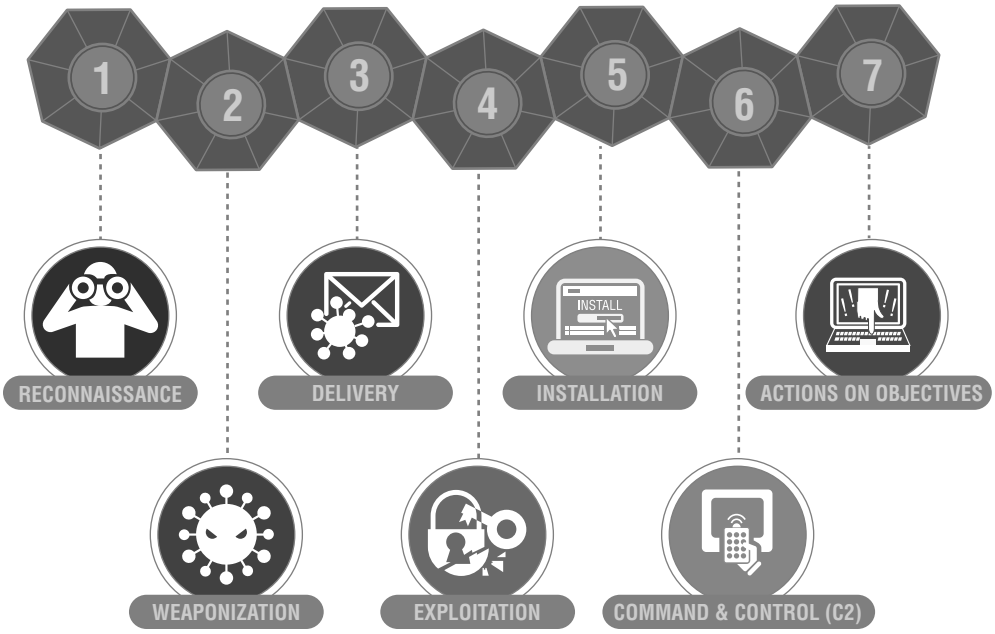
- Explain the basic concepts of scripting and software development.
- Given a scenario, analyze a script or code sample for use in a penetration test.

There is one final objective in Domain 5. Objective 5.3 is "Explain use cases of the following tools during the phases of a penetration test." This objective then lists the many tools covered by the PenTest+ exam. You will find coverage of each of those tools throughout the book in the chapters most closely related to each tool.

The Cyber Kill Chain

The CompTIA penetration testing model described in the previous sections is an important way for penetration testers to structure their activities. There is an equally important counterpart to this model that describes how sophisticated attackers typically organize their work: the Cyber Kill Chain model. This approach, pioneered by Lockheed Martin, consists of the seven stages shown in Figure 1.4.

FIGURE 1.4 The Cyber Kill Chain model

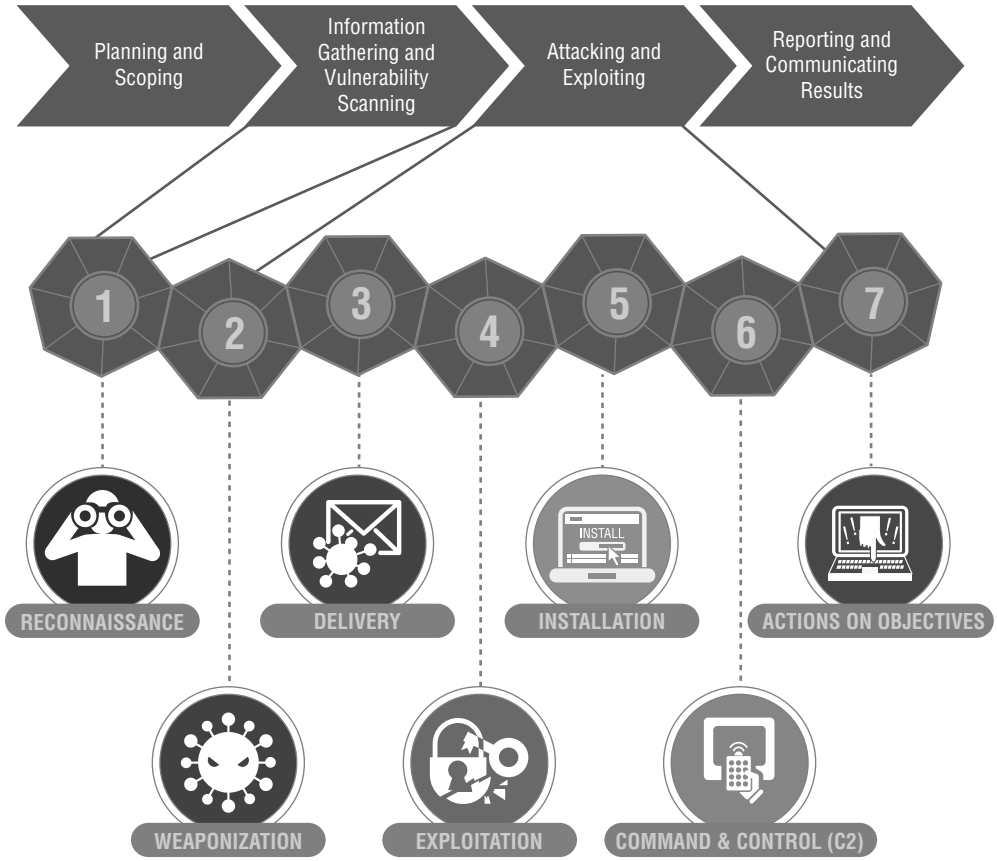


Source: Lockheed Martin

Cybersecurity professionals seeking to adopt the hacker mindset can only do so if they understand how attackers plan and structure their work. The Cyber Kill Chain provides this model. As you seek to reconcile it with the CompTIA process, you might choose to think of it as expanding the Information Gathering and Vulnerability Scanning and Attacking and Exploiting stages into seven more detailed steps, as shown in Figure 1.5.

Captain Chesley "Sully" Sullenberger recently gave a talk on his heroic landing of US Airways Flight 1549 on New York's Hudson River in January 2009. In addition to being an outstanding pilot, Sully is a noted expert on aviation safety. One portion of his talk particularly resonated with this author and made him think of the Cyber Kill Chain. When describing the causes of aviation accidents, Sully said, "Accidents don't happen as the result of a single failure. They occur as the result of a series of unexpected events."

FIGURE 1.5 Cyber Kill Chain in the context of the CompTIA model



Security incidents follow a similar pattern, and penetration testers must be conscious of the series of events that lead to cybersecurity failures. The Cyber Kill Chain illustrates this well, showing the many stages of failure that must occur before a successful breach.

Reconnaissance

The reconnaissance phase of the Cyber Kill Chain maps directly to the Information Gathering and Vulnerability Scanning phase of the penetration testing process. During this phase, attackers gather open source intelligence and conduct initial scans of the target environment to detect potential avenues of exploitation.

Weaponization

After completing the Reconnaissance phase of an attack, attackers move into the remaining six steps, which expand on the Attacking and Exploiting phase of the penetration testing process.

The first of these phases is Weaponization. During this stage, the attackers develop a specific attack tool designed to exploit the vulnerabilities identified during reconnaissance. They often use automated toolkits to develop a malware strain specifically tailored to infiltrate their target.

Delivery

After developing and testing their malware weapon, attackers next must deliver that malware to the target. Delivery may occur through a variety of means, including exploiting a network or application vulnerability, conducting a social engineering attack, distributing malware on an infected USB drive or other media, or sending it as an email attachment or through other means.

Exploitation

Once the malware is delivered to the target organization, the attacker or the victim takes some action that triggers the malware's payload, beginning the Exploitation phase of the Cyber Kill Chain. During this phase, the malware gains access to the targeted system. This may occur when the victim opens a malicious file or when the attacker exploits a vulnerability over the network or otherwise gains a foothold on the target network.

Installation

The initial malware installation is designed only to enable temporary access to the target system. During the next phase of the Cyber Kill Chain, Installation, the attacker uses the initial access provided by the malware to establish permanent, or persistent, access to the target system. For this reason, many people describe the objective of this phase as establishing persistence in the target environment. Attackers may establish persistence by creating a back door that allows them to return to the system at a later date, by creating Registry entries that reopen access once an administrator closes it, or by installing a web shell that allows them to access the system over a standard HTTPS connection.

Command and Control

After establishing persistent access to a target system and network, the attacker may then use a remote shell or other means to remotely control the compromised system. The attacker may manually control the system using the shell or may connect it to an automated command-and-control (C2C) network that provides it with instructions. This automated approach is common in distributed denial-of-service (DDoS) attacks where the attacker simultaneously directs the actions of thousands of compromised systems, known as a botnet.

Actions on Objectives

With a C2C mechanism in place, the attacker may then use the system to advance the original objectives of their attack. This may involve pivoting from the compromised system to other systems operated by the same organization, effectively restarting the Cyber Kill Chain.

The Actions on Objectives stage of the attack may also include the theft of sensitive information, the unauthorized use of computing resources to engage in denial-of-service attacks or to mine cryptocurrency, or the unauthorized modification or deletion of information.

Tools of the Trade

Penetration testers use a wide variety of tools as they conduct their testing. The specific tools chosen for each assessment will depend on the background of the testers, the nature of the target environment, and the rules of engagement, among many other factors.

The PenTest+ exam requires that candidates understand the purposes of a wide variety of tools. In fact, the official exam objectives include a listing of over 50 tools that you'll need to understand before taking the exam. Although you do need to be familiar with these tools, you don't have to be an expert in their use. The official exam objective for these tools says that you must be able to "Explain use cases of the following tools during the phases of a penetration test." It then goes on to state that "The intent of this objective is NOT to test specific vendor feature sets."

This guidance can be frustrating and confusing for test candidates. As you prepare for the exam, you should certainly understand the purpose of each tool. Table 1.1 provides a summary of the tools, broken out by the categories used in the exam objectives. You should be able to describe the purpose of each of these tools in a coherent sentence.

Additionally, the exam objectives include a series of use cases. You should be able to read a scenario covering one of these use cases and then name the appropriate tool(s) for meeting each objective. These use cases include the following topics:

- Reconnaissance
- Enumeration
- Vulnerability scanning
- Credential attacks (offline password cracking and brute-forcing services)
- Persistence
- Configuration compliance
- Evasion
- Decompilation
- Forensics
- Debugging
- Software assurance

TABLE 1.1 Penetration testing tools covered by the PenTest+ exam

Scanners

- Nikto
- Open vulnerability assessment scanner (OpenVAS)
- SQLmap
- Nessus
- Open Security Content Automation Protocol (SCAP)
- Wapiti
- WPScan
- Brakeman
- Scout Suite

OSINT

- WHOIS
- Nslookup
- Fingerprinting Organization with Collected Archives (FOCA)
- theHarvester
- Shodan
- Maltego
- Recon-ng
- Censys

Remote Access Tools

- Secure Shell (SSH)
- Ncat
- Netcat
- ProxyChains

Credential Testing Tools

- Hashcat
- Medusa
- Hydra
- CeWL
- John the Ripper
- Cain
- Mimikatz
- Patator
- DirBuster

Wireless

- Aircrack-ng suite
- Kismet
- WiFite
- Rogue access point
- EAPHammer
- mdk4
- Spooftooph
- Reaver
- Wireless Geographic Logging Engine (WiGLE)
- Fern

Networking Tools

- Wireshark
- Hping

Debuggers

- OllyDbg
- Immunity Debugger
- GNU Debugger (GDB)
- WinDbg
- Interactive Disassembler (IDA)
- Covenant
- SearchSploit

Web Application Tools

- OWASP ZAP
- Burp Suite
- Gobuster
- W3AF

Social Engineering Tools

- Social Engineering Toolkit (SET)
- BeEF

Miscellaneous Tools

- SearchSploit
- PowerSploit
- Responder
- Impacket tools
- Empire
- Metasploit
- Mitm6
- CrackMapExec
- TruffleHog

Steganography Tools

- Open steg
- Steghide
- Snow
- Coagula
- Sonic Visualiser
- TinEye
- Metagoofil
- Online SSL checkers

Cloud Tools

- Scout Suite
- CloudBrute
- Pacu
- Cloud Custodian

In the remainder of this chapter, you'll learn about some of these tools at a very high level. We will then revisit each tool and use case as we progress through the remainder of the book. You'll find references in the following sections that help you locate the more detailed explanations of each tool later in the book.

You'll want to return to Table 1.1 as a reference as you continue through your test preparation. It's also a great review sheet to use the night before you take the exam.

Now, let's discuss these tools briefly in the context of the penetration testing process. We're going to deviate from the CompTIA categories a bit here to help put this information into the easiest context for you to understand. Remember, this is just an overview and we'll return to each of these tools later in the book.

Reconnaissance

During the Information Gathering and Vulnerability Scanning phase of a penetration test, the testing team spends a large amount of time gathering information. Most of this information is collected using open source intelligence (OSINT) tools and techniques that simply comb through publicly available information for organizational and technical details that might prove useful during the penetration test.

A variety of tools assist with OSINT collection:

- *WHOIS* tools gather information from public records about domain ownership.
- *Nslookup* tools help identify the IP addresses associated with an organization.
- *theHarvester* scours search engines and other resources to find email addresses, employee names, and infrastructure details about an organization.
- *Recon-ng* is a modular web reconnaissance framework that organizes and manages OSINT work.
- *Censys* is a web-based tool that probes IP addresses across the Internet and then provides penetration testers with access to that information through a search engine.
- *FOCA* (Fingerprinting Organizations with Collected Archives) is an open source tool used to find metadata within Office documents, PDFs, and other common file formats.
- *Shodan* is a specialized search engine to provide discovery of vulnerable Internet of Things (IoT) devices from public sources.
- *Maltego* is a commercial product that assists with the visualization of data gathered from OSINT efforts.

In addition to these OSINT tools, penetration testers must be familiar with the Nmap network scanning tool. Nmap is the most widely used network port scanner and is a part of almost every cybersecurity professional's toolkit.

You'll find coverage of all of these tools in Chapter 3, "Information Gathering."

In most cases, you don't need to know the detailed use of cybersecurity tools covered by the PenTest+ exam. However, Nmap is an exception to this general rule. You do need to know the syntax and common options used with Nmap, as they are described in an exam objective. Don't worry; you'll learn everything you need to know in Chapter 3.

Vulnerability Scanners

Vulnerability scanners also play an important role in the information-gathering stages of a penetration test. Once testers have identified potential targets, they may use vulnerability scanners to probe those targets for weaknesses that might be exploited during future stages of the test.

You'll need to be familiar with these vulnerability scanning tools for the exam:

- *Nessus* is a commercial vulnerability scanning tool used to scan a wide variety of devices.
- *OpenVAS* is an open source alternative to commercial tools such as Nessus. OpenVAS also performs network vulnerability scans.
- *Sqlmap* is an open source tool used to automate SQL injection attacks against web applications with database back ends.
- *Nikto*, *Wapiti*, and *W3AF* are open source web application vulnerability scanners. *WPScan* is a web application testing tool designed to work with websites running the WordPress content management system.
- *Security Content Automation Protocol (SCAP)* is a set of tools designed to help organizations manage compliance with security standards.

You'll learn more about these tools in Chapter 4, "Vulnerability Scanning," and Chapter 5, "Analyzing Vulnerability Scans."

Social Engineering

Social engineering plays an important role in many attacks. As penetration testers move into the Attacking and Exploiting phase of their work, they often begin with social engineering attacks to harvest credentials.

The PenTest+ exam includes coverage of two toolkits used by social engineers:

- The *Social Engineer Toolkit (SET)* provides a framework for automating the social engineering process, including sending spear phishing messages, hosting fake websites, and collecting credentials.

- Similarly, the *Browser Exploitation Framework* (*BeEF*) provides an automated toolkit for using social engineering to take over a victim's web browser.

Both of these tools are described in more detail in Chapter 8, "Exploiting Physical and Social Vulnerabilities."

Credential Testing Tools

If attackers aren't able to gain access to credentials through social engineering techniques, they may be able to use tools to reverse-engineer hashed passwords.

The PenTest+ exam includes coverage of a large set of tools designed to assist with these activities:

- *Hashcat*, *John the Ripper*, *Hydra*, *Medusa*, *Patator*, and *Cain* are password-cracking tools used to reverse-engineer hashed passwords stored in files.
- *CeWL* is a custom wordlist generator that searches websites for keywords that may be used in password-guessing attacks.
- *Mimikatz* retrieves sensitive credential information from memory on Windows systems.
- *DirBuster* is a brute-forcing tool used to enumerate files and directories on a web server.

We'll cover all of these tools in more detail in Chapter 10, "Attacking Hosts, Cloud Technologies, and Specialized Systems."

Debuggers and Software Testing Tools

Debugging tools provide insight into software and assist with reverse engineering activities. Penetration testers preparing for the exam should be familiar with these debugging tools:

- *Immunity Debugger* is designed specifically to support penetration testing and the reverse engineering of malware.
- *GDB* is a widely used open source debugger for Linux that works with a variety of programming languages.
- *OllyDbg* is a Windows debugger that works on binary code at the assembly language level.
- *WinDbg* is another Windows-specific debugging tool that was created by Microsoft.
- *IDA* is a commercial debugging tool that works on Windows, Mac, and Linux platforms.
- *Brakeman* is a static software analysis tool used for scanning Ruby on Rails applications.

- *Covenant* is a software security testing tool used for testing .NET applications.
- *TruffleHog* is a tool that scans through code repositories for accidentally published secrets.

We'll provide detailed coverage of these tools in Chapter 9, "Exploiting Application Vulnerabilities."

Network Testing

In addition to exploiting software vulnerabilities, penetration testers often exploit flaws in networks as they seek access to systems. The network testing tools covered by the PenTest+ exam include:

- *Wireshark* is a protocol analyzer that allows penetration testers to eavesdrop on and dissect network traffic.
- *Hping* is a command-line tool that allows testers to artificially generate network traffic.
- *Aircrack-ng*, *WiFite*, *mdk4*, *Fern*, and *Kismet* are wireless network security testing tools.
- *Rogue wireless access points* are used to attract connections from unsuspecting users.
- *EAPHammer* is used to conduct evil twin attacks against WPA2-Enterprise wireless networks.
- *Reaver* is used to conduct attacks against networks that support Wi-Fi Protected Setup (WPS).
- *Spooftooph* is used to perform attacks against Bluetooth-enabled devices.
- The *Wireless Geographic Logging Engine (WiGLE)* is an open database of wireless network information collected by the community and published for open access.
- *Online SSL checkers* are used to determine whether websites are susceptible to SSL and/or TLS vulnerabilities.

You'll learn more about each of these tools in Chapter 7, "Exploiting Network Vulnerabilities."

Remote Access

After gaining initial access to a network, penetration testers seek to establish persistence so that they may continue to access a system. These are some of the tools used to assist with this task:

- *Secure Shell (SSH)* provides secure encrypted connections between systems.
- *Ncat* and *Netcat* provide an easy way to read and write data over network connections.

- *Proxychains* allows testers to force connections through a proxy server where they may be inspected and altered before being passed on to their final destination.

You'll learn more about each of these tools in Chapter 10, "Attacking Hosts, Cloud Technologies, and Specialized Systems."

Exploitation

As attackers work their way through a network, they use a variety of exploits to compromise new systems and escalate the privileges they have on systems they've already compromised. Exploitation toolkits make this process easy and automated. For the exam, you should be familiar with the following exploitation tools:

- *Metasploit* is, by far, the most popular exploitation framework and supports thousands of plug-ins covering different exploits.
- *SearchSploit* is a command-line tool that allows you to search through a database of known exploits.
- *PowerSploit* and *Empire* are Windows-centric sets of PowerShell scripts that may be used to automate penetration testing tasks.
- *Responder* is a toolkit used to answer NetBIOS queries from Windows systems on a network.
- *Impacket* is a set of network tools that provide low-level access to network protocols.
- *Mitm6* is a tool used to conduct attacks against IPv6 networks.
- *CrackMapExec* is a set of tools used after gaining access to a network to assess the security of an Active Directory environment.

You'll learn more about each of these tools in Chapter 6, "Exploiting and Pivoting."

Steganography

Steganography is the art of hiding information in plain sight, normally within image files or other large binary files. For the PenTest+ exam, you should be familiar with the following steganography tools:

- *Open Steg* and *Steghide* are general-purpose steganography tools used to hide text within images and other binary files.
- *Coagula* is used to embed text within audio files. *Sonic Visualiser* is an audio analysis tool that may be used to detect alterations made by steganography tools.
- *Snow* uses whitespace and tabs within a document to hide information.
- *TinEye* is a reverse image search tool that allows security researchers to identify the original image when they suspect steganography is being used.
- *Metagoofil* is used to extract metadata from a large variety of file types.

You'll learn more about each of these tools in Chapter 9.

Cloud Tools

As organizations move operations to the cloud, security teams need new tools designed to assess the security of those cloud services. For the PenTest+ exam, you should be familiar with the following cloud security tools:

- *ScoutSuite* is a cloud security auditing tool that can work across commonly used cloud environments.
- *CloudBrute* is a scanner used to identify the cloud components used by an organization.
- *Pacu* is a cloud exploitation framework focused on Amazon Web Services (AWS)-hosted environments.
- *Cloud Custodian* is a rule enforcement engine that allows the consistent application of security policies across cloud environments.

You'll learn more about each of these tools in Chapter 10, "Attacking Hosts, Cloud Technologies, and Specialized Systems."

Summary

Penetration testing is an important practice that allows cybersecurity professionals to assess the security of environments by adopting the hacker mindset. By thinking like an attacker, testers are able to identify weaknesses in the organization's security infrastructure and potential gaps that may lead to future security breaches.

The CompTIA penetration testing process includes four phases: Planning and Scoping, Information Gathering and Vulnerability Scanning, Attacking and Exploiting, and Reporting and Communication. Penetration testers follow each of these phases to ensure that they have a well-designed test that operates using agreed-upon rules of engagement.

Penetration testers use a wide variety of tools to assist in their work. These are many of the same tools used by cybersecurity professionals, hackers, network engineers, system administrators, and software developers. Tools assist with all stages of the penetration testing process, especially information gathering, vulnerability identification, and exploiting vulnerabilities during attacks.

Exam Essentials

Know how the CIA and DAD triads describe the goals of cybersecurity professionals and attackers. Cybersecurity professionals strive to protect the confidentiality, integrity, and availability of information and systems. Attackers seek to undermine these goals by achieving the goals of destruction, alteration, and denial.

Be able to name several important benefits of penetration testing. Penetration testing provides knowledge about an organization's security posture that can't be obtained elsewhere.

It also provides a blueprint for the remediation of security issues. Finally, penetration tests provide focused information on specific attack targets.

Understand that penetration testing may be conducted to meet regulatory requirements. The Payment Card Industry Data Security Standard (PCI DSS) requires that organizations involved in the processing of credit card transactions conduct both internal and external penetration tests on an annual basis.

Describe how both internal and external teams may conduct penetration tests. Internal teams have the benefit of inside knowledge about the environment. They also operate more cost-effectively than external teams. External penetration testers have the benefit of organizational independence from the teams who designed and implemented the security controls.

Know the four phases of the penetration testing process. Penetration testers begin in the Planning and Scoping phase, where they develop a statement of work and agree with the client on rules of engagement. They then move into reconnaissance efforts during the Information Gathering and Vulnerability Scanning phase. The information collected is then used to conduct attacks during the Attacking and Exploiting phase. During the final phase, Reporting and Communicating Results, the team shares its findings with the target organization.

Describe the tools used by penetration testers. Tools designed for use by cybersecurity professionals and other technologists may also assist penetration testers in gathering information and conducting attacks. Penetration testers use specialized exploitation frameworks, such as Metasploit, to help automate their work.

Lab Exercises

Activity 1.1: Adopting the Hacker Mindset

Before we dive into the many technical examples throughout this book, let's try an example of applying the hacker mindset to everyday life.

Think about the grocery store where you normally shop. What are some of the security measures used by that store to prevent the theft of cash and merchandise? What ways can you think of to defeat those controls?

Activity 1.2: Using the Cyber Kill Chain

Choose a real-world example of a cybersecurity incident from recent news. Select an example in which there is a reasonable amount of technical detail publicly available.

Describe this attack in terms of the Cyber Kill Chain. How did the attacker carry out each step of the process? Were any steps skipped? If there is not enough information available to definitively address an element of the Cyber Kill Chain, offer some assumptions about what may have happened.

Review Questions

You can find the answers in the Appendix.

1. Tom is running a penetration test in a web application and discovers a flaw that allows him to shut down the web server remotely. What goal of penetration testing has Tom most directly achieved?
   - **A.** Disclosure
   - **B.** Integrity
   - **C.** Alteration
   - **D.** Denial

2. Brian ran a penetration test against a school's grading system and discovered a flaw that would allow students to alter their grades by exploiting a SQL injection vulnerability. What type of control should he recommend to the school's cybersecurity team to prevent students from engaging in this type of activity?
   - **A.** Confidentiality
   - **B.** Integrity
   - **C.** Alteration
   - **D.** Availability

3. Edward Snowden gathered a massive quantity of sensitive information from the National Security Agency and released it to the media without permission. What type of attack did he wage?
   - **A.** Disclosure
   - **B.** Denial
   - **C.** Alteration
   - **D.** Availability

4. Assuming no significant changes in an organization's cardholder data environment, how often does PCI DSS require that a merchant accepting credit cards conduct penetration testing?
   - **A.** Monthly
   - **B.** Semiannually
   - **C.** Annually
   - **D.** Biannually

5. Which one of the following is *not* a benefit of using an internal penetration testing team?
   - **A.** Contextual knowledge
   - **B.** Cost
   - **C.** Subject matter expertise
   - **D.** Independence

6. Which one of the following is *not* a reason to conduct periodic penetration tests of systems and applications?
   - **A.** Changes in the environment
   - **B.** Cost
   - **C.** Evolving threats
   - **D.** New team members

7. Rich recently got into trouble with a client for using an attack tool during a penetration test that caused a system outage. During what stage of the penetration testing process should Rich and his clients have agreed on the tools and techniques that he would use during the test?
   - **A.** Planning and Scoping
   - **B.** Information Gathering and Vulnerability Scanning
   - **C.** Attacking and Exploiting
   - **D.** Reporting and Communication Results

8. Which one of the following steps of the Cyber Kill Chain does *not* map to the Attacking and Exploiting stage of the penetration testing process?
   - **A.** Weaponization
   - **B.** Reconnaissance
   - **C.** Installation
   - **D.** Actions on Objectives

9. Beth recently conducted a phishing attack against a penetration testing target in an attempt to gather credentials that she might use in later attacks. What stage of the penetration testing process is Beth in?
   - **A.** Planning and Scoping
   - **B.** Attacking and Exploiting
   - **C.** Information Gathering and Vulnerability Scanning
   - **D.** Reporting and Communication

10. Which one of the following security assessment tools is not commonly used during the Information Gathering and Vulnerability Scanning phase of a penetration test?
    - **A.** Nmap
    - **B.** Nessus
    - **C.** Metasploit
    - **D.** Nslookup

11. During what phase of the Cyber Kill Chain does an attacker steal information, use computing resources, or alter information without permission?

A. Weaponization

B. Installation

C. Actions on Objectives

D. Command and Control

12. Grace is investigating a security incident where the attackers left USB drives containing infected files in the parking lot of an office building. What stage in the Cyber Kill Chain describes this action?

A. Weaponization

B. Installation

C. Delivery

D. Command and Control

13. Which one of the following is *not* an open source intelligence gathering tool?

A. WHOIS

B. Nslookup

C. Nessus

D. FOCA

14. Which one of the following tools is an exploitation framework commonly used by penetration testers?

A. Metasploit

B. Wireshark

C. Aircrack-ng

D. SET

15. Which one of the following tools is *not* a password-cracking utility?

A. OWASP ZAP

B. Cain and Abel

C. Hashcat

D. Jack the Ripper

16. Which one of the following vulnerability scanners is specifically designed to test the security of web applications against a wide variety of attacks?

A. OpenVAS

B. Nessus

C. SQLmap

D. Nikto

17. Which one of the following debugging tools does not support Windows systems?
    - **A.** GDB
    - **B.** OllyDbg
    - **C.** WinDbg
    - **D.** IDA

18. What is the final stage of the Cyber Kill Chain?
    - **A.** Weaponization
    - **B.** Installation
    - **C.** Actions on Objectives
    - **D.** Command and Control

19. Which one of the following activities assumes that an organization has already been compromised?
    - **A.** Penetration testing
    - **B.** Threat hunting
    - **C.** Vulnerability scanning
    - **D.** Software testing

20. Alan is creating a list of recommendations that his organization can follow to remediate issues identified during a penetration test. In what phase of the testing process is Alan participating?
    - **A.** Planning and Scoping
    - **B.** Reporting and Communication
    - **C.** Attacking and Exploiting
    - **D.** Information Gathering and Vulnerability Scanning