IN THIS CHAPTER

- Understanding hackers' and malicious users' objectives
- Examining how the security testing process came about
- Recognizing what endangers your computer systems
- Understanding how artificial intelligence can help
- Starting to use the process for security testing

Chapter 1
Introduction to Vulnerability and Penetration Testing

This book is about testing your computers and networks for security vulnerabilities and plugging the holes you find before the bad guys get a chance to exploit them. Understanding the concepts in this chapter is your first step in this process.

Straightening Out the Terminology

Everyone has heard of hackers and malicious users. Many people have even suffered the consequences of their criminal actions. Who are these people, and why do you need to know about them? The next few sections give you the lowdown on these attackers.

REMEMBER

In this book, I use the following terminology:

- **Hackers** (or *external attackers, often called black-hat hackers)* try to compromise computers, sensitive information, and even entire networks for ill-gotten gains — usually from the outside — as unauthorized users. Hackers go for almost any system they think they can compromise. Some prefer prestigious, well-protected systems, but hacking into anyone's system increases an attacker's status in hacker circles.
- **Malicious users** (*internal attackers)* try to compromise computers and sensitive information from the inside as authorized and trusted users. Malicious users go for systems that they believe they can compromise for ill-gotten gains or revenge, because they may have access or knowledge of a system that gives them a leg up. Plus, they know that their efforts will often go undetected because they usually already have credentials into the systems they wish to exploit.

  Malicious attackers are, generally speaking, both hackers and malicious users. For the sake of simplicity, I refer to both as *hackers* and specify *hacker* or *malicious user* only when I need to differentiate and drill down further into their unique tools, techniques, and ways of thinking.
- **Ethical hackers** (or *good guys),* often referred to as white-hat hackers or penetration testers, hack systems to discover vulnerabilities to protect against unauthorized access, abuse, and misuse. Information security researchers, consultants, and internal staff fall into this category. These ethical hackers often work as part of a *red team* within an organization whose purpose is to find and exploit vulnerabilities across the entire network, companywide. Sometimes red team security professionals will work in conjunction with what's referred to as a *blue team* that specializes in security defense or a purple team that does both to provide a unified approach to security offense and defense.

Hacker

Hacker has two meanings:

- Traditionally, hackers like to tinker with software or electronic systems. Hackers enjoy exploring and learning how computer systems operate. They love discovering new ways to work — both mechanically and electronically.
- Over the years, *hacker* has taken on a new meaning: someone who maliciously breaks into systems for personal gain. Technically, these criminals are *crackers* (criminal hackers). These "crackers" break into — or crack — systems with malicious intent. They seek fame, intellectual property, profit, or even revenge. They modify, delete, and steal critical information, and they spread ransomware and take entire networks offline, often bringing large corporations and government agencies to their knees.

WARNING

Don't get me started on how pop culture and the media have hijacked the word *hack*, from *life hacking* to so-called election meddling. Marketers, politicians, and media strategists know that the average person doesn't understand the term *hacking,* so many of them use it however they desire to achieve their goals. Don't be distracted.

The good-guy *(white-hat)* hackers don't like being lumped in the same category as the bad-guy *(black-hat)* hackers. (In case you're curious, the *white hat* and *black hat* come from old Western TV shows in which the good guys wore white cowboy hats and the bad guys wore black cowboy hats.) *Gray-hat* hackers are a bit of both. Whatever the case, the word *hacker* often has a negative connotation.

Many malicious hackers claim that they don't cause damage but help others for the greater good of society. Yeah, whatever. Malicious hackers are electronic miscreants and deserve the consequences of their actions.

Be careful not to confuse criminal hackers with security researchers. Researchers not only hack aboveboard and develop the amazing tools that we get to use in our work, but they also (usually) take responsible steps to disclose their findings and publish their code. Unfortunately, a war is going on against legitimate information security research, and the tools and techniques are often questioned by government agencies. Some people are even forced to remove these tools from their websites.

Malicious user

A malicious user — meaning a rogue employee, contractor, intern, or other user who abuses their trusted privileges — is a common term in security circles and in

headlines about information breaches. The issue isn't necessarily users hacking internal systems but users who abuse the computer access privileges they've been given. Users ferret through critical database systems to glean sensitive information, email confidential client information to the competition or elsewhere to the cloud to save for later, or delete sensitive files from servers that they probably didn't need to have access to in the first place.

Sometimes, an innocent (or ignorant) insider whose intent isn't malicious still causes security problems by moving, deleting, or corrupting sensitive information. Even an innocent fat finger on the keyboard can have dire consequences in the business world. Think about all the ransomware infections affecting businesses around the world. All it takes is one click by a careless user for your entire network to be affected.

Malicious users are often the worst enemies of IT and information security professionals because they know exactly where to go to get the goods and don't need to be computer-savvy to compromise sensitive information. These users have the access they need, and management trusts them — often without question.

Recognizing How Malicious Attackers Beget Ethical Hackers

You need protection from hacker shenanigans. Along the lines of what my father taught me about being smarter than the machine you're working on, you have to become as savvy as the guys who are trying to attack your systems. A true IT or security professional possesses the skills, mindset, and tools of a hacker but is trustworthy. They perform hacks as security tests against systems based on how hackers think and work and make tireless efforts to protect the organizations' network and information assets.

REMEMBER

Ethical hacking (more commonly known as vulnerability and penetration testing in the business world) involves the same tools, tricks, and techniques that criminal hackers use, with one major difference: It's performed with the target's permission in a professional setting. The intent of this testing is to discover vulnerabilities from a malicious attacker's viewpoint to better secure systems. Vulnerability and penetration testing is part of an overall information risk management program that allows for ongoing security improvements. This security testing can also ensure that vendors' claims about the security of their products are legitimate.

SECURITY TESTING CERTIFICATIONS

If you perform vulnerability and penetration tests and want to add another certification to your credentials, you may want to consider becoming a Certified Ethical Hacker (C|EH) through a certification program by EC-Council. See `www.eccouncil.org` for more information. Like Certified Information Systems Security Professional (CISSP), the C|EH certification is a well-known, respected certification in the industry, accredited by the American National Standards Institute (ANSI 17024).

Other options include the SANS Global Information Assurance Certification (GIAC) program and the Offensive Security Certified Professional (OSCP) program, a hands-on security testing certification. I love the approach of the certifications, as all too often, people who perform this type of work don't have the proper hands-on experience with the tools and techniques to do it well. See `www.giac.org`, and `www.offsec.com` for more information.

Vulnerability and penetration testing versus auditing

Many people confuse security testing via vulnerability and penetration testing with security auditing, but *big* differences exist in the objectives. Security auditing involves comparing a company's security policies (or compliance requirements) with what's actually taking place. The intent of security auditing is to validate that security controls exist, typically by using a risk-based approach. Auditing often involves reviewing business processes, and in some cases, it isn't as technical. Some security audits, in fact, can be as basic as security checklists that simply serve to meet a specific compliance requirement.

REMEMBER

Not all audits are high-level, but many of the ones I've seen — especially those involving compliance with the Payment Card Industry Data Security Standard (PCI DSS) and the Health Insurance Portability and Accountability Act (HIPAA) — are quite simplistic. Often, these audits are performed by people who have no technical security experience — or, worse, work outside IT altogether!

Conversely, security assessments based on ethical hacking focus on vulnerabilities that can be exploited. This testing approach validates that security controls *don't* exist or are ineffectual. This formal vulnerability and penetration testing can be both highly technical and nontechnical, and although it involves the use of formal methodology, it tends to be a bit less structured than formal auditing. Where auditing is required (such as for SSAE 18 SOC reports and the ISO 27001 certification) in your organization, you might consider integrating the

vulnerability and penetration testing techniques I outline in this book into your IT/security audit program. You might actually be required to do so. Auditing and vulnerability and penetration testing complement one another really well.

Policy considerations

If you choose to make vulnerability and penetration testing an important part of your business's information risk management program, you need to have a documented security testing policy. Such a policy outlines who's doing the testing, the general type of testing that's performed, and how often the testing takes place. Specific procedures for carrying out your security tests could outline the methodologies I cover in this book. You should also consider creating security standards documented along with your policy that outline the specific security testing tools used and the specific people performing the testing. You could establish standard testing dates, such as once per quarter for external systems and biannual tests for internal systems — whatever works for your business.

Compliance and regulatory concerns

Your own internal policies may dictate how management views security testing, but you also need to consider the state, federal, and international laws and regulations that affect your business. In particular, the Digital Millennium Copyright Act (DMCA) sends chills down the spines of legitimate researchers. See `www.eff.org/issues/dmca` for everything that the DMCA has to offer.

Many federal laws and regulations in the United States — such as the Health Insurance Portability and Accountability Act (HIPAA) and the associated Health Information Technology for Economic and Clinical Health (HITECH) Act, Gramm-Leach-Bliley Act (GLBA), North American Electric Reliability Corporation (NERC) Critical Infrastructure Protection (CIP) requirements, and the Payment Card Industry Data Security Standard (PCI DSS) — require strong security controls and consistent security assessments. There's also the Cybersecurity Maturity Model Certification (CMMC). CMMC is a follow-on to NIST Special Publication 800-171 Protecting Controlled Unclassified Information in Nonfederal Systems and Organizations. This certification is intended to ensure that the U.S. Department of Defense's (DoD's) Defense Industrial Base (DIB) of suppliers/contractors are adequately protecting the DOD's information assets.

Related international laws —such as the Canadian Personal Information Protection and Electronic Documents Act (PIPEDA), the European Union's General Data Protection Regulation (GDPR), and Japan's Personal Information Protection Act (JPIPA) — are no different. Incorporating your security tests into these compliance requirements is a great way to meet state and federal regulations and to beef up your overall information security and privacy program.

Understanding the Need to Hack Your Own Systems

To catch a thief, you must think like a thief. That adage is the basis of vulnerability and penetration testing. Knowing your enemy is critical. The law of averages works against security. With the increased number of hackers and their expanding knowledge and the growing number of system vulnerabilities and other unknowns, all computer systems and applications are likely to be hacked or compromised somehow. Protecting your systems from the bad guys — not just addressing general security best practices — is critical. When you know hacker tricks, you find out how vulnerable your systems really are and can take the necessary steps to make them secure.

Hacking preys on weak security practices and both disclosed and undisclosed vulnerabilities. More and more research, such as the annual Verizon Data Breach Investigations Report (`www.verizon.com/business/resources/reports/dbir/`), shows that long-standing, *known* vulnerabilities are continually being targeted. Firewalls, advanced endpoint security, security incident and event management (SIEM), and other fancy (and expensive) security technologies often create a false feeling of safety. Attacking your own systems to discover vulnerabilities — especially the low-hanging fruit that gets so many people into trouble — helps you go beyond security products to make them even more secure. Vulnerability and penetration testing is a proven method for greatly hardening your systems from attack. If you don't identify weaknesses, it's only a matter of time before the vulnerabilities are exploited.

As hackers expand their knowledge, so should you. You must think like them and work like them to protect your systems from them. As a security professional, you must know the activities that the bad guys carry out, as well as how to stop their efforts. Knowing what to look for and how to use that information helps you thwart their efforts.

TIP

You don't have to protect your systems from *everything*. You can't. The only protection against everything is unplugging your computer systems and locking them away so no one can touch them — not even you and especially not your users. But doing so is not the best approach to security, and it's certainly not good for business! What's important is protecting your systems from known vulnerabilities and common attacks — the 20 percent of the issues that create 80 percent of the risks, which happen to be some of the most overlooked weaknesses in most organizations. Seriously, you wouldn't believe the basic flaws I see in my work!

Anticipating all the possible vulnerabilities you'll have in your systems and business processes is impossible. You certainly can't plan for all types of attacks — especially the unknown ones. But the more combinations you try and the more often you test whole systems instead of individual units, the better your chances are of discovering vulnerabilities that affect your information systems in their entirety.

Don't take your security testing too far, though; hardening your systems from unlikely (or even *less* likely) attacks makes little sense and will probably get in the way of doing business.

REMEMBER

Your overall goals for security testing are to

- Prioritize your systems so that you can focus your efforts on what matters.
- Test your systems in a nondestructive fashion.
- Enumerate vulnerabilities and, if necessary, prove to management that business risks exist.
- Apply results to address the vulnerabilities and better secure your systems.

Understanding the Dangers Your Systems Face

It's one thing to know generally that your systems are under fire from hackers around the world and malicious users around the office; it's another to understand specific potential attacks against your systems. This section discusses some well-known attacks but is by no means a comprehensive listing.

Many security vulnerabilities aren't critical by themselves, but exploiting several vulnerabilities at the same time can take its toll on a system or network environment. A default Windows operating system (OS) configuration, a weak SQL Server administrator password, or a mission-critical workstation running on a wireless network may not be a major security concern by itself. But someone who exploits all three of these vulnerabilities simultaneously could enable unauthorized remote access and disclose sensitive information (among other things).

REMEMBER

Complexity is the enemy of security.

Vulnerabilities and attacks have grown enormously in recent years because of virtualization, cloud computing, and even social media. These three things alone add immeasurable complexity to your environment. On top of that, with the new ways of the world and so many people working from home, the complexities have grown exponentially.

Nontechnical attacks

Exploits that involve manipulating people — your users and even you — are often the greatest vulnerability. Humans are trusting by nature, which can lead to social engineering exploits. *Social engineering* is exploiting the trusting nature of human beings to gain information — often via email phishing — for malicious purposes. With dramatic increases in the size of the remote workforce, social engineering has become an even greater threat, especially with more personal devices being used that are likely much less secure. Check out Chapter 6 for more information about social engineering and how to guard your systems and users against it.

Other common, effective attacks against information systems are physical. Hackers break into buildings, computer rooms, or other areas that contain critical information or property to steal computers, servers, and other valuable equipment. Physical attacks can also include *dumpster diving* — rummaging through trash cans and bins for intellectual property, passwords, network diagrams, and other information.

Network infrastructure attacks

Attacks on network infrastructures can be easy to accomplish because many networks can be reached from anywhere in the world via the internet. Examples of network infrastructure attacks include the following:

- Connecting to a network through an unsecured wireless access point attached behind a firewall
- Exploiting weaknesses in network protocols, such as File Transfer Protocol (FTP) and Secure Sockets Layer (SSL)
- Flooding a network with too many requests, creating denial of service (DoS) for legitimate requests
- Installing a network analyzer on a network segment and capturing packets that travel across it, revealing confidential information in cleartext

Operating system attacks

Hacking an OS is a preferred method of the bad guys. OS attacks make up a large portion of attacks simply because every computer has an operating system. They are susceptible to many well-known exploits, including vulnerabilities that remain unpatched years later.

Occasionally, some OSes that tend to be more secure out of the box — such as the old-but-still-out-there Novell NetWare, OpenBSD, and IBM Series i — are attacked, and vulnerabilities turn up. But hackers tend to prefer attacking Windows, Linux, and macOS because they're more widely used.

Here are some examples of attacks on operating systems:

- Exploiting missing patches
- Attacking built-in authentication systems
- Breaking file system security
- Installing ransomware to lock down the system to extort money or other assets
- Cracking passwords and weak encryption implementations

Application and other specialized attacks

Applications take a lot of hits by hackers. Web applications and mobile apps, which are probably the most popular means of attack, are often beaten down. The following are examples of application attacks and related exploits that are often present on business networks:

- Websites and applications are everywhere. Thanks to what's called *shadow IT,* in which people in various areas of the business run and manage their own technology, website applications are in every corner of the internal network and out in the cloud. Unfortunately, many IT and security professionals are unaware of the presence of shadow IT and the risks it creates.
- Mobile apps face increasing attacks, given their popularity in business settings. There are also rogue apps discovered on the app stores that can create challenges in your environment.
- Unsecured files containing sensitive information are scattered across workstation and server shares as well as out into the cloud in places like Microsoft OneDrive and Google Drive. Database systems also contain numerous vulnerabilities that malicious users can exploit.

Integrating AI into the testing mix

Artificial intelligence (AI) is rapidly transforming the world of information security. There's certainly no shortage of resources and use cases for AI when it comes to vulnerability and penetration testing. AI allows us to learn about the online footprint of an organization along with the people involved such as executives and key employees whose roles can be exploited. AI can also help us get very technical with things such as script code for specific exploits as well as evaluating code, logs, and so on for areas of exploitation and compromise. AI can even help us write our reports, which can be a huge asset.

For now, we still can't do without our traditional tools and techniques for vulnerability and penetration testing. AI isn't going away! I believe this technology will help elevate us professionals — and the criminal hackers — so it pays to hone your AI skills. This addition of *Hacking For Dummies* will help you do just that.

Following the Security Assessment Principles

Security professionals must carry out the same attacks against computer systems, physical controls, and people that malicious hackers do. (I introduce those attacks in the preceding section.) A security professional's intent, however, is to highlight any associated weaknesses. Parts 2 through 5 of this book cover how you might proceed with these attacks in detail, along with specific countermeasures you can implement against attacks on your business.

To ensure that security testing is performed adequately and professionally, every security professional needs to follow a few basic tenets. The following sections introduce the important principles.

WARNING

If you don't heed these principles, bad things could happen. I've seen them ignored or forgotten by IT departments while planning and executing security tests. The results weren't positive; trust me.

Working ethically

The word *ethical* in this context means working with high professional morals and values. Whether you're performing security tests against your own systems or for someone who has hired you, everything you do must be aboveboard in support of the company's goals, with no hidden agenda — just professionalism. Being

ethical also means reporting all your findings, whether or not they may create political backlash. Don't laugh; on numerous occasions, I've witnessed people brushing off security vulnerability findings because they didn't want to rock the boat or to deal with difficult executives or vendors.

Trustworthiness is the ultimate tenet. It's also the best way to get (and keep) people on your side in support of your security program. Misusing information and power is forbidden; that's what the bad guys do, so let them be the ones who pay a fine or go to prison because of their poor choices.

Respecting privacy

Treat the information you gather with respect. All information you obtain during your testing — from web application flaws to clear text email passwords to personally identifiable information (PII) and beyond — must be kept private. Nothing good can come of snooping into confidential corporate information or employees' or customers' private lives.

TIP

Involve others in your process. Employ a peer review or similar oversight system that can help build trust and support for your security assessment projects.

Not crashing your systems

One of the biggest mistakes I've seen people make when trying to test their own systems is inadvertently crashing the systems they're trying to keep running. Crashing systems doesn't happen as often as it used to given the resiliency of today's hardware and software, but poor planning and timing can have negative consequences.

You can create DoS conditions on your systems when testing, including taking servers offline or flooding email inboxes through things like web contact forms not protected by a CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) challenge. Running too many tests too quickly can cause system lockups, data corruption, reboots, and similar problems, especially when you're testing older servers and web applications. (I should know; I've done it!) Don't assume that a network or specific host can handle the beating that network tools and vulnerability scanners can dish out.

You can even accidentally create accounts or lock users out of the network without realizing the consequences. Proceed with caution and common sense. Either way, be it you or someone else, these weaknesses likely exist on your network, and it's better that you discover them first!

TIP

Most vulnerability scanners can control how many requests are sent to each system simultaneously. These settings are especially handy when you need to run the tests on production systems during regular business hours. Don't be afraid to throttle back your scans. Completing your testing will take longer, but throttling back may save you a lot of grief if an unstable system is present.

Using the Vulnerability and Penetration Testing Process

As with practically any IT or security project, you need to plan security testing. It's been said that action without planning is the root of every failure. Strategic and tactical issues in vulnerability and penetration testing need to be determined and agreed on in advance. To ensure the success of your efforts, spend time planning for any amount of testing, from a simple OS password-cracking test against a few servers to a penetration test of a complex web environment.

WARNING

If you choose to hire a "reformed" hacker to work with you during your testing or to obtain an independent perspective, be careful. I cover the pros and cons and the do's and don'ts associated with hiring security resources in Chapter 19.

Formulating your plan

Getting approval for security testing is essential. Make sure that what you're doing is known and visible — at least to the decision-makers. Obtaining sponsorship of the project is the first step. This is how your testing objectives are defined. Sponsorship could come from your manager, an executive, your client, or even yourself if you're the boss. You need someone to back you up and sign off on your plan. Otherwise, your testing may be called off unexpectedly if someone (including third parties such as cloud service and hosting providers) claims that you were never authorized to perform the tests. Worse, you could be fired or charged with criminal activity.

The authorization can be as simple as an internal memo or an email from your boss when you perform these tests on your own systems. If you're testing for a client, have a signed contract stating the client's support and authorization. Get written approval of this sponsorship before you ever start working to ensure that none of your time or effort is wasted. This documentation is your "Get Out of Jail Free" card if anyone — such as your internet service provider (ISP), cloud service provider, or a related vendor — questions what you're doing or if the authorities come calling. Don't laugh — it wouldn't be the first time it has happened.

One slip can crash your systems, which isn't necessarily what anyone wants. You need a detailed plan, but you don't need volumes of testing procedures that make the plan overly complex. A well-defined scope includes the following information:

- **Specific systems to be tested:** When selecting systems to test, start with the most critical systems and processes or the ones that you suspect are the most vulnerable. You could test server OS passwords, test an internet-facing web application, or attempt social engineering via phishing before drilling down into all your systems. Another consideration is whether to test computer systems that are being used by employees who are working from home. Unless they are connected to the corporate environment over a VPN or are otherwise remotely accessible, you might not even be able to reach them. Furthermore, what are the ramifications of testing computers — especially personal systems — that are running on a home network? Are there medical devices, specific software, or Internet of Things (IoT) systems that might be disrupted? Of course, you may have forgotten about all those systems out in the cloud as well. Just be careful and make sure you have permission testing systems in AWS, Azure, and other cloud services! Thinking all of this through with all the right people is imperative.
- **Risks involved:** Have a contingency plan for your security testing process in case something goes awry. Suppose that you're assessing your firewall or a web application, and you take it down. This situation can cause system unavailability, which can reduce system performance or employee productivity. Worse, it might cause data integrity loss, loss of data itself, and even bad publicity. It'll most certainly tick off a person or two and make you look bad. All of these can create business risks. A use case for AI would be to simply ask your preferred tools such as ChatGPT, Copilot, and Grok what you should be concerned with and how to best handle it when something goes sideways.

  Handle social engineering and DoS attacks carefully as well. Determine how they might affect the people and systems you test. If you end up causing technical or business challenges, you could even ask AI how to best handle the situation and how you should best respond to stakeholders. It's absolutely amazing — like having a business coach at your beck and call. Just use AI carefully so that you're not dropping sensitive information that could lead to exposures down the road. And take AI's advice with a grain of salt because, well, AI has been known to hallucinate, and it's certainly not as intelligent as it think it is in many cases.
- **Dates when the tests will be performed and overall timeline:** Determining when the tests are to be performed is something you must think long and hard about. Decide whether to perform tests during normal business hours, late at night, or early in the morning so that production systems aren't affected. Involve others to make sure that they approve of your timing.

You may get pushback and suffer DoS-related consequences, but the best approach is an unlimited attack, in which any type of test is possible at any time of day. The bad guys aren't breaking into your systems within a limited scope, so why should you? Some exceptions to this approach are performing all-out DoS attacks, social engineering, and physical security tests.

TIP

- **Whether you intend to be detected:** One of your goals may be to perform the tests without being detected. You might perform your tests on remote systems or on a remote office and don't want the users to be aware of what you're doing. Otherwise, the users or IT staff may catch on to you and be on their best behavior instead of their normal behavior.
- **Whether to leave security controls enabled:** An important, yet often overlooked, issue is whether to leave enabled security controls such as firewalls, intrusion prevention systems (IPSes), and web application firewalls (WAFs) so that they block scans and exploit attempts. Leaving these controls enabled provides a real-world picture of where things stand. But I've found *much* more value in disabling these controls (in the form of whitelisting your source IP addresses) so that you can pull back the curtains and find the greatest number of vulnerabilities.

  Many people want to leave their security controls enabled. After all, that approach can make them look better, because many security checks will likely be blocked. To me, this defense-in-depth approach is great, but it can create a serious false sense of security and doesn't paint the entire picture of an organization's overall security posture. There's no right or wrong answer. Just make sure that everyone is on board with what is being tested and what the final outcomes and report represent.
- **Knowledge of the systems before testing:** You don't need extensive knowledge of the systems you're testing — just basic understanding, which protects both you and the tested systems. Understanding the systems you're testing shouldn't be difficult if you're testing your own in-house systems. If you're testing a client's systems, you may have to dig deeper. Only one or two clients have asked me for a fully blind assessment.

  Most IT managers and others who are responsible for security may be scared of blind assessments, which can take more time, cost more, and be less effective. Base the type of test you perform on the organization's or client's needs.
- **Actions to take when a major vulnerability is discovered:** Don't stop after you find one or two security holes; keep going to see what else you can discover. I'm not saying that you should keep testing until the end of time or until you crash all your systems; ain't nobody got time for that! Instead, simply pursue the path you're going down until you can't hack it any longer (pun intended). If you haven't found any vulnerabilities, you haven't looked hard enough. Be it external domain related issues, web applications, or internal

systems, I can say with confidence that vulnerabilities are there! If you uncover something big such as a weak password or SQL injection on an external system or compromised internal Active Directory credentials, you need to share that information with the key players (developers, database administrators, IT managers, and so on) as soon as possible to plug the hole. Don't wait until you publish your report as it could be too late by then.

- **The specific deliverables:** Deliverables may include vulnerability scanner reports and your own distilled report outlining important vulnerabilities to address, along with recommendations and countermeasures to implement.

Selecting tools

As in any project, if you don't have the right tools for your security testing, you'll have difficulty accomplishing the task effectively. Having said that, just because you use the right tools doesn't mean that you'll discover all the right vulnerabilities. Experience counts.

TIP

Know the limitations of your tools. Many vulnerability scanners and testing tools generate false positives and negatives (incorrectly identifying vulnerabilities). Others skip vulnerabilities. In certain situations, such as testing web applications, you have to run multiple vulnerability scanners to find all the vulnerabilities.

Many tools focus on specific tests, and no tool can test for everything. For the same reason that you wouldn't drive a nail with a screwdriver, don't use a port scanner to uncover specific network vulnerabilities or a wireless network analyzer to test a web application. You need a set of specific tools for the task. The more (and better) tools you have, the easier your security testing efforts will be.

Make sure that you're using tools like these for your tasks:

- To crack passwords, you need cracking tools such as Ophcrack and Proactive Password Auditor.
- For an in-depth analysis of a web application, a web vulnerability scanner (such as Acunetix Web Vulnerability Scanner or Probely) is more appropriate than a network analyzer (such as Wireshark or OmniPeek).

The capabilities of many security and hacking tools are misunderstood. This misunderstanding has cast a negative light on otherwise excellent and legitimate tools; even government agencies around the world are talking about making them illegal. Part of this misunderstanding is due to the complexity of some of these

security testing tools, but it's largely based in ignorance and the desire for control. Whichever tools you use, familiarize yourself with them before you start using them. That way, you're prepared to use the tools in the ways that they're intended to be used. Here are ways to do that:

- Read the readme and/or online help files and FAQs (frequently asked questions).
- Study the user guides.
- Use the tools in a lab or test environment.
- Watch tutorial videos on YouTube (if you can bear the poor production of most of them).
- Get your favorite AI tool to walk you through the usage and gotchas. You can even leverage AI to help optimize your network, OS, and application configurations.
- Consider formal classroom training from the security-tool vendor or another third-party training provider, if available.

Look for these characteristics in tools for security testing:

- Adequate documentation
- Detailed reports on discovered vulnerabilities, including how they might be exploited and fixed
- General industry acceptance
- Availability of updates and responsiveness of technical support
- High-level reports that can be presented to managers or nontechnical types (especially important in today's audit- and compliance-driven world)

These features can save you a ton of time and effort when you're performing your tests and writing your final reports.

Executing the plan

Good security testing takes persistence. Time and patience are important. Also, be careful when you're performing your tests. A criminal on your network or a seemingly benign employee looking over your shoulder may watch what's going on and use this information against you or your business.

Making sure that no hackers are on your systems before you start isn't practical. Just be sure to keep everything as quiet and private as possible, especially when you're transmitting and storing test results. If possible, encrypt any emails and files that contain sensitive test information or share them via a cloud-based file sharing service.

You're on a reconnaissance mission. Harness as much information as possible about your organization and systems — much as malicious hackers do. Start with a broad view and narrow your focus. Follow these steps:

1. **Search the internet for your organization's name, its computer and network system names, and its IP addresses.**

   Google is a great place to start. I have found ChatGPT, Microsoft Copilot, and other LLMs to work nicely for this as well. Agentic (autonomous) AI can work if you have such capabilities built into existing systems or use a tool such as AutoGPT.

2. **Narrow your scope, targeting the specific systems you're testing.**

   Whether you're assessing physical security structures or web applications, a casual assessment can turn up a lot of information about your systems.

3. **Further narrow your focus by performing scans and other detailed tests to uncover vulnerabilities on your systems.**

4. **Perform the attacks and exploit any vulnerabilities you find (if that's what you choose to do).**

Check out Chapters 4 and 5 for information and tips on this process.

Evaluating results

Assess your results to see what you've uncovered, assuming that the vulnerabilities haven't been made obvious before now. Knowledge counts. Your skill in evaluating the results and correlating the specific vulnerabilities discovered will get better with practice. You'll end up knowing your systems much better than anyone else does, which will make the evaluation process much simpler moving forward. Again, you know what tool to involve if more help is needed: AI.

TIP

Submit a formal report to management or to your client outlining your results and any recommendations you need to share. Keep these parties in the loop to show that your efforts and their money are well spent. Chapter 17 describes the security assessment reporting process.

Moving on

When you finish your security tests, you (or your business or your client) will still need to implement your recommendations to make sure that the systems are secure. Otherwise, all the time, money, and effort spent on testing goes to waste. Sadly, I see this very scenario fairly often in which vulnerabilities uncovered initially are still there 6 to 12 months later. It makes no sense, and it is certainly an indefensible approach to security governance.

REMEMBER

New security vulnerabilities continually appear. Information systems change and are becoming more complex. New security vulnerabilities and exploits are being uncovered. Vulnerability scanners and related testing tools get better. We, as security professionals are learning more each day. AI is on the scene and making things much better. Security tests provide a snapshot of the security posture of your systems. At any time, everything can change, especially after you upgrade software, add computer systems, or apply patches. This situation underscores the need to keep your tools updated — before each use, if possible. Plan to test regularly and consistently (such as quarterly, semi-annually, or annually). Chapter 19 covers managing security changes as you move forward.