Chapter 1

Introduction to Ethical Hacking

In This Chapter

- Understanding hackers' and malicious users' objectives
- Differentiating between ethical hackers and malicious attackers
- Examining how the ethical hacking process came about
- Understanding the dangers that your computer systems face
- Starting to use the ethical hacking process

This book is about hacking ethically — the methodology of testing your computers and networks for security vulnerabilities and plugging the holes you find before the bad guys get a chance to exploit them.

Although *ethical* is an often overused and misunderstood word, *Webster's New World Dictionary* defines *ethical* perfectly for the context of this book and the professional security testing techniques that I cover — that is, "conforming to the standards of conduct of a given profession or group." IT and information security practitioners are obligated to perform the tests covered in this book aboveboard and only after permission has been obtained by the owner(s) of the systems — hence the disclaimer in this book's Introduction.

Straightening Out the Terminology

Most people have heard of hackers and malicious users. Many have even suffered the consequences of hackers' criminal actions. So who are these people? And why do you need to know about them? The next few sections give you the lowdown on these attackers.

In this book, I use the following terminology:

- *Hackers* (or external attackers) try to compromise computers and sensitive information for ill-gotten gains — usually from the outside — as an unauthorized user. 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 internal users* (or internal attackers) try to compromise computers and sensitive information from the inside as authorized and "trusted" users. Malicious users go for systems they believe they can compromise for ill-gotten gains or revenge.

  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 drill down further into their tools, techniques, and ways of thinking.
- *Ethical hackers* (or good guys) hack systems to discover vulnerabilities to protect against unauthorized access, abuse, and misuse.

Defining 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.
- In recent years, hacker has taken on a new meaning — someone who maliciously breaks into systems for personal gain. Technically, these criminals are *crackers* (criminal hackers). Crackers break into, or crack, systems with malicious intent. They are out for personal gain: fame, profit, and even revenge. They modify, delete, and steal critical information, often making other people miserable.

The good-guy *(white hat)* hackers don't like being in the same category as the bad-guy *(black hat)* hackers. (In case you're curious, the white hat and black hat terms 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 little bit of both. Whatever the case, most people have a negative connotation for the word *hacker*.

Many malicious hackers claim that they don't cause damage but instead help others. Yeah, right. Malicious hackers are electronic thieves and deserve the consequences of their actions.

Defining malicious user

Malicious users — meaning a rogue employee, contractor, intern, or other user who abuses his or her privileges — is a common term in security circles and in headlines about information breaches. A long-standing statistic states that insiders carry out 80% of all security breaches. Whether this number is accurate is still questionable, but based on what I've seen and numerous annual surveys, undoubtedly an insider problem makes up the majority of all computer breaches.

The issue is not necessarily users "hacking" internal systems, but rather users who abuse the computer access privileges they've been given. Users ferret through critical database systems to glean sensitive information, e-mail confidential client information to the competition or other third parties, or delete sensitive files from servers that they probably didn't need to have access to in the first place. There's also the occasional ignorant insider whose intent is not malicious but who still causes security problems by moving, deleting, or corrupting sensitive information.

Malicious users are often ethical hackers' worst enemies 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 the management trusts them without question.

Recognizing How Malicious Attackers Beget Ethical Hackers

You need protection from hacker shenanigans; you need (or need to become) an ethical hacker. An ethical hacker possesses the skills, mindset, and tools of a hacker but is also trustworthy. Ethical hackers perform the hacks as security tests for their systems based on how hackers might work.

Ethical hacking — which encompasses formal and methodical penetration testing, white hat hacking, and vulnerability testing — involves the same tools, tricks, and techniques that hackers use, but with one major difference: Ethical hacking is performed with the target's permission. The intent of ethical hacking is to discover vulnerabilities from a malicious attacker's viewpoint to better secure systems. Ethical hacking is part of an overall information risk management program that allows for ongoing security improvements. Ethical hacking can also ensure that vendors' claims about the security of their products are legitimate.

If you perform ethical hacking tests for clients or simply want to add another certification to your credentials, you might want to consider becoming a Certified Ethical Hacker (C|EH), through a certification program sponsored by EC-Council. See `www.eccouncil.org/CEH.htm` for more information.

Ethical hacking versus auditing

Many people confuse ethical hacking with security auditing but there are big differences. Security auditing involves comparing a company's security policies to what's actually taking place. The intent of security auditing is to validate that security controls exist — typically using a risk-based approach. Auditing often involves reviewing business processes and might not be very technical. I often refer to security audits as "security checklists" because they're usually based off (you guessed it) checklists.

Conversely, ethical hacking focuses on vulnerabilities that can be exploited. It validates that security controls *do not* exist. Ethical hacking can be both highly technical and nontechnical and, although you do use a formal methodology, it tends to be a bit less structured than formal auditing. If auditing continues to take place in your organization, you might consider integrating the ethical hacking techniques I outline into your auditing process.

Policy considerations

If you choose to make ethical hacking an important part of your business's risk management program, you really need to have a documented security testing policy. Such a policy outlines the type of ethical hacking that is done, which systems (such as servers, Web applications, laptops, and so on) are tested, and how often the testing is performed. Specific procedures for carrying out your security tests could outline the ethical hacking methodology I cover in this book. You might also consider creating a security standards document that outlines the specific security testing tools that are used and specific dates your systems are tested each year. You might list standard testing dates, such as once per quarter for external systems and biannual tests for internal systems.

Compliance and regulatory concerns

Your own internal policies might dictate how company management views security testing, but you also need to consider the state, federal, and global laws and regulations that affect your business. Many of the federal laws and regulations, such as the Health Insurance Portability and Accountability Act

(HIPAA), Gramm-Leach-Bliley Act (GLBA), North American Electric Reliability Corporation (NERC) CIP requirements, and Payment Card Industry Data Security Standard (PCI DSS) require periodic and consistent security evaluations. Incorporating your ethical hacking into these required tests is a great way to meet the state and federal regulations and beef up your overall privacy and security compliance program.

Understanding the Need to Hack Your Own Systems

To catch a thief, you must think like a thief. That's the basis for ethical hacking. Knowing your enemy is absolutely critical. See Chapter 2 for details about how malicious attackers work.

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, eventually, all computer systems and applications will be hacked or compromised in some way. Protecting your systems from the bad guys — and not just the generic vulnerabilities that everyone knows about — is absolutely critical. When you know hacker tricks, you find out how vulnerable your systems really are.

Hacking preys on weak security practices and undisclosed vulnerabilities. Firewalls, encryption, and passwords can create a false feeling of safety. These security systems often focus on high-level vulnerabilities, such as basic access control, without affecting how the bad guys work. Attacking your own systems to discover vulnerabilities helps make them more secure. Ethical hacking is the only proven method of 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 the ethical hacker, you must know the activities that hackers carry out and how to stop their efforts. Knowing what to look for and how to use that information helps you to thwart hackers' efforts.

You don't have to protect your systems from *everything*. You can't. The only protection against everything is to unplug your computer systems and lock them away so no one can touch them — not even you. But doing so is not the best approach to information security and it's certainly not good for business. What's important is to protect your systems from known vulnerabilities and common attacks.

Anticipating all the possible vulnerabilities you'll have in your systems and business processes is impossible. You certainly can't plan for all possible attacks — especially the unknown ones. However, the more combinations you try and the more 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 ethical hacking too far, though; hardening your systems from unlikely attacks makes little sense. For instance, if you don't have a lot of foot traffic in your office and no internal Web server running, you might not have as much to worry about as an Internet hosting provider might have. Your overall goals as an ethical hacker are

- Prioritize your systems so you can focus your efforts on what matters.
- Hack your systems in a nondestructive fashion.
- Enumerate vulnerabilities and, if necessary, prove to management that vulnerabilities exist and can be exploited.
- Apply results to remove 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 the specific attacks against your systems that are possible. This section offers some well-known attacks but is by no means a comprehensive listing.

Many information security vulnerabilities aren't critical by themselves. However, exploiting several vulnerabilities at the same time can take its toll on a system. For example, a default Windows OS configuration, a weak SQL Server administrator password, or a server hosted on a wireless network might not be major security concerns separately — but a hacker exploiting all three of these vulnerabilities at the same time could lead to sensitive information disclosure and more.

Nontechnical attacks

Exploits that involve manipulating people — end users and even yourself — are the greatest vulnerability within any computer or network infrastructure. Humans are trusting by nature, which can lead to social engineering exploits.

Social engineering is the exploitation of the trusting nature of human beings to gain information for malicious purposes. Check out Chapter 5 for more information about social engineering and how to guard your systems against it.

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

Network infrastructure attacks

Hacker attacks against network infrastructures can be easy because many networks can be reached from anywhere in the world via the Internet. Some examples of network infrastructure attacks include the following:

- Connecting to a network through an unsecured wireless router attached behind a firewall
- Exploiting weaknesses in network protocols, such as TCP/IP and NetBIOS
- Flooding a network with too many requests, creating a denial of service (DoS) for legitimate requests
- Installing a network analyzer on a network and capturing every packet that travels across it, revealing confidential information in clear text

Operating system attacks

Hacking an operating system (OS) is a preferred method of the bad guys. OS attacks make up a large portion of hacker attacks simply because every computer has an operating system and OSes are susceptible to many well-known exploits.

Occasionally, some operating systems that tend to be more secure out of the box — such as Novell NetWare and OpenBSD— are attacked, and vulnerabilities turn up. But hackers often prefer attacking Windows and Linux because they are widely used and better known for their weaknesses.

Here are some examples of attacks on operating systems:

- Exploiting specific network protocol implementations
- Attacking built-in authentication systems
- Breaking file system security
- Cracking passwords and weak encryption implementations

Application and other specialized attacks

Applications take a lot of hits by hackers. Programs, such as e-mail server software and Web applications, are often beaten down:

- Hypertext Transfer Protocol (HTTP) and Simple Mail Transfer Protocol (SMTP) applications are frequently attacked because most firewalls and other security mechanisms are configured to allow full access to these services from the Internet.
- Voice over Internet Protocol (VoIP) faces increasing attacks as it finds its way into more and more businesses.
- Unsecured files containing sensitive information are scattered throughout workstation and server shares, and database systems contain numerous vulnerabilities that malicious users can exploit.

Ethical hackers carry out such attacks against computer systems, physical controls, and people and highlight any associated weaknesses. Parts II through V of this book cover these attacks in detail, along with specific countermeasures you can implement against attacks against your business.

Obeying the Ethical Hacking Commandments

Every ethical hacker must abide by a few basic commandments. If not, bad things can happen. I've seen these commandments ignored or forgotten when planning or executing ethical hacking tests. The results weren't positive — trust me.

Working ethically

The word ethical in this context means working with high professional morals and principles. Whether you're performing ethical hacking tests against your own systems or for someone who has hired you, everything you do as an ethical hacker must be aboveboard and must support the company's goals. No hidden agendas allowed!

Trustworthiness is the ultimate tenet. The misuse of information is absolutely forbidden. That's what the bad guys do. Let them receive a fine or go to prison because of their poor choices.

Respecting privacy

Treat the information you gather with the utmost respect. All information you obtain during your testing — from Web application log files to clear text passwords to personally identifiable information and beyond — must be kept private. Don't snoop into confidential corporate information or employees' private lives. If you sense that a colleague or team member breaches privacy and you feel like someone should know about it, consider sharing that information with the appropriate manager or project sponsor.

Involve others in your process. Employ a watch-the-watcher system that can help build trust and support for your ethical hacking projects.

Not crashing your systems

One of the biggest mistakes I've seen people make when trying to hack their own systems is inadvertently crashing the systems they're trying to keep running. Poor planning is the main cause of this mistake. These testers either have not read the documentation or misunderstand the usage and power of the security tools and techniques at their disposal.

Although it's not likely, you can create DoS conditions on your systems when testing. Running too many tests too quickly can cause system lockups, data corruption, reboots, and more. I should know: I've done it! Don't rush and assume that a network or specific host can handle the beating that network tools and vulnerability scanners can dish out.

Many vulnerability scanners can control how many tests are performed on a system at the same time. These tools are especially handy when you need to run the tests on production systems during regular business hours.

You can even accidentally create an account or system lockout condition by socially engineering someone into changing a password, not realizing the consequences of your actions.

Using the Ethical Hacking Process

Like practically any IT or security project, ethical hacking needs to be planned. It's been said that action without planning is at the root of every failure. Strategic and tactical issues in the ethical hacking process need to be determined and agreed upon. To ensure the success of your efforts, spend time up front planning for any amount of testing — from a simple password-cracking test to an all-out penetration test on a Web application.

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, cons, do's, and don'ts associated with hiring an ethical hacker in Chapter 18.

Formulating your plan

Getting approval for ethical hacking 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. 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 might be called off unexpectedly if someone claims you were never authorized to perform the tests.

The authorization can be as simple as an internal memo or an e-mail 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 on this sponsorship as soon as possible to ensure that none of your time or effort is wasted. This documentation is your Get Out of Jail Free card if anyone questions what you're doing, or worse, if the authorities come calling. Don't laugh — it wouldn't be the first time it happened.

One slip can crash your systems — not necessarily what anyone wants. You need a detailed plan, but that doesn't mean you need volumes of testing procedures to make things 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 you suspect are the most vulnerable. For instance, you can test server OS passwords, an Internet-facing Web application, or attempt social engineering attacks before drilling down into all your systems.
- **Risks involved:** Have a contingency plan for your ethical hacking process in case something goes awry. What if you're assessing your firewall or Web application and you take it down? This can cause system unavailability, which can reduce system performance or employee productivity. Even worse, it might cause loss of data integrity, loss of data itself, and even bad publicity. It'll most certainly tick off a person or two and make you look bad.

Handle social engineering and DoS attacks carefully. Determine how they affect the systems you test and your entire organization.

- **Dates the tests will be performed and your overall timeline:** Determining when the tests are performed is something that you must think long and hard about. Do you perform tests during normal business hours? How about late at night or early in the morning so that production systems aren't affected? Involve others to make sure they approve of your timing.

  The best approach is an *unlimited attack,* where 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 DoS attacks, social engineering, and physical security tests.

- **Knowledge of the systems you have before you start testing:** You don't need extensive knowledge of the systems you're testing — just a basic understanding. This basic understanding helps protect you and the tested systems.

  Understanding the systems you're testing shouldn't be difficult if you're hacking your own in-house systems. If you're testing a client's systems, you may have to dig deeper. In fact, I've only had one or two clients ask for a fully blind assessment. Most IT managers and others responsible for security are scared of these assessments — and they can take more time and cost more. Base the type of test you perform on your organization or client's needs.

- **Actions you will take when a major vulnerability is discovered:** Don't stop after you find one security hole. Keep going to see what else you can discover. I'm not saying to keep hacking until the end of time or until you crash all your systems; simply pursue the path you're going down until you can't hack it any longer (pun intended). If you haven't found any vulnerability, you haven't looked hard enough. If you uncover something big, you do need to share that information with the key players as soon as possible to plug the hole before it's exploited.

- **The specific deliverables:** This includes vulnerability scanner reports and a higher-level report outlining the important vulnerabilities to address, along with countermeasures to implement.

One of your goals might be to perform the tests without being detected. For example, you might perform your tests on remote systems or on a remote office, and you don't want the users aware of what you're doing. Otherwise, the users might catch on to you and be on their best behavior — instead of their normal behavior.

Selecting tools

As with any project, if you don't have the right tools for ethical hacking, you might 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.

Know the personal and technical limitations. Many vulnerability scanners generate false positives and negatives (incorrectly identifying vulnerabilities). Others just skip right over vulnerabilities altogether. In certain situations, you might need to run multiple vulnerability scanners to find the most vulnerabilities.

Many tools focus on specific tests, and no tool can test for everything. For the same reason you wouldn't drive a nail with a screwdriver, don't use a word processor to scan your network for open ports. This is why you need a set of specific tools for the task. The more (and better) tools you have, the easier your ethical hacking efforts are.

Make sure you're using the right tool for the task:

- To crack passwords, you need cracking tools, such as ophcrack and Proactive Password Auditor.

  A general port scanner, such as SuperScan or Nmap, won't work for cracking passwords and rooting out detailed vulnerabilities.
- For an in-depth analysis of a Web application, a Web application assessment tool (such as N-Stalker or WebInspect) is more appropriate than a network analyzer (such as Wireshark).

When selecting the right security tool for the task, ask around. Get advice from your colleagues and from other people online. A simple groups search on Google (`http://groups.google.com`), LinkedIn (`www.linkedin.com`) or a perusal of security portals, such as `http://SecurityFocus.com` and `http://SearchSecurity.com`, often produces great feedback from other security experts about what works and what doesn't.

Hundreds, if not thousands, of tools can be used for ethical hacking — from software-based vulnerability scanner programs to hardware-based network analyzers. The following list runs down some of my favorite commercial, freeware, and open source security tools:

- Cain & Abel
- OmniPeek

- SuperScan
- QualysGuard
- WebInspect
- Proactive Password Auditor
- Metasploit
- LANguard
- AirMagnet WiFi Analyzer

I discuss these tools and many others in Parts II through V when I go into the specific hack attacks. Appendix A contains a more comprehensive listing of these tools for your reference.

The capabilities of many security and hacking tools are often misunderstood. This misunderstanding has cast a negative light on otherwise excellent and legitimate tools.

Some of these security testing tools are complex. Whichever tools you use, familiarize yourself with them before you start using them. Here are ways to do that:

- Read the readme and/or online help files and FAQs.
- Study the user's guides.
- Use the tools in a lab or test environment.
- Consider formal classroom training from the security tool vendor or another third-party training provider, if available.

Look for these characteristics in tools for ethical hacking:

- Adequate documentation
- Detailed reports on the discovered vulnerabilities, including how they might be exploited and fixed
- General industry acceptance
- Availability of updates and support
- High-level reports that can be presented to managers or nontechnical types

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 ethical hacking takes persistence. Time and patience are important. Be careful when you're performing your ethical hacking tests. A hacker in your network or a seemingly benign employee looking over your shoulder might 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. Be sure you keep everything as quiet and private as possible. This is especially critical when transmitting and storing your test results. If possible, encrypt any e-mails and files containing sensitive test information by using Pretty Good Privacy (PGP) (`www.pgp.com`), encrypted Zip file, or similar technology.

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

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

   Google is a great place to start.

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 with a more critical eye. Perform actual 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 Chapter 4 to find out more information and tips on using this process.

Evaluating results

Assess your results to see what you've uncovered, assuming that the vulnerabilities haven't been made obvious before now. This is where knowledge counts. Your skill at 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. This makes the evaluation process much simpler moving forward.

Submit a formal report to upper management or to your client, outlining your results and any recommendations you wish to share. Keep these parties in the loop to show that your efforts and their money are well spent. Chapter 16 describes the ethical hacking reporting process.

Moving on

When you finish your ethical hacking tests, you (or your client) still need to implement your recommendations to make sure the systems are secure. Otherwise, all the time, money, and effort spent on ethical hacking goes to waste.

New security vulnerabilities continually appear. Information systems constantly change and become more complex. New hacker exploits and security vulnerabilities are regularly uncovered. You might even discover new ones yourself! Vulnerability scanners get better and better. Security tests are a snapshot of the security posture of your systems. At any time, everything can change, especially after upgrading software, adding computer systems, or applying patches. Plan to test regularly and consistently (for example, once a month, once a quarter, or biannually). Chapter 18 covers managing security changes.