Chapter 1

Introduction to Ethical Hacking

In This Chapter

- Understanding hacker objectives
- Outlining the differences between ethical hackers and malicious hackers
- Examining how the ethical hacking process has come about
- Understanding the dangers that your computer systems face
- Starting the ethical hacking process

This book is about hacking ethically — the science of testing your computers and network 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, the Merriam-Webster dictionary defines *ethical* perfectly for the context of this book and the professional security testing techniques that I cover — that is, *conforming to accepted professional standards of conduct*. IT practitioners are obligated to perform all 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 the introduction.

How Hackers Beget Ethical Hackers

We've all heard of hackers. Many of us have even suffered the consequences of hacker actions. So who are these hackers? Why is it important to know about them? The next few sections give you the lowdown on hackers.

Defining hacker

Hacker is a word that has two meanings:

- Traditionally, a hacker is someone who likes to tinker with software or electronic systems. Hackers enjoy exploring and learning how computer systems operate. They love discovering new ways to work electronically.

- Recently, *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 *(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. (These terms come from Western movies where the good guys wore white cowboy hats and the bad guys wore black cowboy hats.) Whatever the case, most people give *hacker* a negative connotation.

Many malicious hackers claim that they don't cause damage but instead are altruistically helping others. Yeah, right. Many malicious hackers are electronic thieves.

In this book, I use the following terminology:

- *Hackers* (or *bad guys*) try to compromise computers.
- *Ethical hackers* (or *good guys*) protect computers against illicit entry.

Hackers go for almost any system they think they can compromise. Some prefer prestigious, well-protected systems, but hacking into anyone's system increases their status in hacker circles.

Ethical Hacking 101

You need protection from hacker shenanigans. 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.

If you perform ethical hacking tests for customers or simply want to add another certification to your credentials, you may want to consider the ethical hacker certification Certified Ethical Hacker, which is sponsored by EC-Council. See `www.eccouncil.org/CEH.htm` for more information.

Ethical hacking — also known as *penetration testing* or *white-hat hacking* — involves the same tools, tricks, and techniques that hackers use, but with one major difference: Ethical hacking is legal. Ethical hacking is performed with the target's permission. The intent of ethical hacking is to discover vulnerabilities from a hacker's viewpoint so systems can be better secured. It's 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.

To hack your own systems like the bad guys, you must think like they think. It's absolutely critical to know your enemy; see Chapter 2 for details.

Understanding the Need to Hack Your Own Systems

To catch a thief, think like a thief. That's the basis for ethical hacking.

The law of averages works against security. With the increased numbers and expanding knowledge of hackers combined with the growing number of system vulnerabilities and other unknowns, the time will come when all computer systems are 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 can see how vulnerable your systems are.

Hacking preys on weak security practices and undisclosed vulnerabilities. Firewalls, encryption, and virtual private networks (VPNs) can create a false feeling of safety. These security systems often focus on high-level vulnerabilities, such as viruses and traffic through a firewall, without affecting how hackers work. Attacking your own systems to discover vulnerabilities is a step to making them more secure. This is the only proven method of greatly hardening your systems from attack. If you don't identify weaknesses, it's a matter of time before the vulnerabilities are exploited.

As hackers expand their knowledge, so should you. You must think like them to protect your systems from them. You, as the ethical hacker, must know activities hackers carry out and how to stop their efforts. You should know what to look for and how to use that information 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. That's not the best approach to information security. What's important is to protect your systems from known vulnerabilities and common hacker attacks.

It's impossible to buttress all possible vulnerabilities on all your systems. You can't plan for all possible attacks — especially the ones that are currently unknown. However, the more combinations you try — the more you test whole systems instead of individual units — the better your chances of discovering vulnerabilities that affect everything as a whole.

Don't take ethical hacking too far, though. It makes little sense to harden your systems from unlikely attacks. For instance, if you don't have a lot of foot traffic

in your office and no internal Web server running, you may not have as much to worry about as an Internet hosting provider would have. However, don't forget about insider threats from malicious employees!

Your overall goals as an ethical hacker should be as follows:

- Hack your systems in a nondestructive fashion.
- Enumerate vulnerabilities and, if necessary, prove to upper management that vulnerabilities exist.
- Apply results to remove vulnerabilities and better secure your systems.

Understanding the Dangers Your Systems Face

It's one thing to know that your systems generally are under fire from hackers around the world. It's another to understand specific attacks against your systems that are possible. This section offers some well-known attacks but is by no means a comprehensive listing. That requires its own book: *Hack Attacks Encyclopedia,* by John Chirillo (Wiley Publishing, Inc.).

Many information-security vulnerabilities aren't critical by themselves. However, exploiting several vulnerabilities at the same time can take its toll. For example, a default Windows OS configuration, a weak SQL Server administrator password, and a server hosted on a wireless network may not be major security concerns separately. But exploiting all three of these vulnerabilities at the same time can be a serious issue.

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 defined as the exploitation of the trusting nature of human beings to gain information for malicious purposes. I cover social engineering in depth in Chapter 5.

Other common and effective attacks against information systems are physical. Hackers break into buildings, computer rooms, or other areas containing critical information or property. Physical attacks can 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. Here are some examples of network-infrastructure attacks:

- Connecting into a network through a rogue modem attached to a computer behind a firewall
- Exploiting weaknesses in network transport mechanisms, 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
- Piggybacking onto a network through an insecure 802.11b wireless configuration

Operating-system attacks

Hacking operating systems (OSs) is a preferred method of the bad guys. OSs comprise a large portion of hacker attacks simply because every computer has one and so many well-known exploits can be used against them.

Occasionally, some operating systems that are more secure out of the box — such as Novell NetWare and the flavors of BSD UNIX — are attacked, and vulnerabilities turn up. But hackers prefer attacking operating systems like Windows and Linux because they are widely used and better known for their vulnerabilities.

Here are some examples of attacks on operating systems:

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

Application and other specialized attacks

Applications take a lot of hits by hackers. Programs such as e-mail server software and Web applications often are 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 programs from the Internet.
- Malicious software *(malware)* includes viruses, worms, Trojan horses, and spyware. Malware clogs networks and takes down systems.
- *Spam* (junk e-mail) is wreaking havoc on system availability and storage space. And it can carry malware.

Ethical hacking helps reveal such attacks against your computer systems. Parts II through V of this book cover these attacks in detail, along with specific countermeasures you can implement against attacks on your systems.

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.

Working ethically

The word *ethical* in this context can be defined as 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 are allowed!

Trustworthiness is the ultimate tenet. The misuse of information is absolutely forbidden. That's what the bad guys do.

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 — must be kept private. Don't use this information to snoop into confidential corporate information or private lives. If you sense that someone should know there's a problem, consider sharing that information with the appropriate manager.

Involve others in your process. This is a "watch the watcher" system that can build trust and support your ethical hacking projects.

Not crashing your systems

One of the biggest mistakes I've seen when people try to hack their own systems is inadvertently crashing their systems. The main reason for this is poor planning. These testers have not read the documentation or misunderstand the usage and power of the security tools and techniques.

You can easily create DoS conditions on your systems when testing. Running too many tests too quickly on a system causes many system lockups. I know because I've done this! Don't rush things and assume that a network or specific host can handle the beating that network scanners and vulnerability-assessment tools can dish out.

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

You can even create an account or system lockout condition by social engineering someone into changing a password, not realizing that doing so might create a system lockout condition.

The Ethical Hacking Process

Like practically any IT or security project, ethical hacking needs to be planned in advance. Strategic and tactical issues in the ethical hacking process should be determined and agreed upon. Planning is important for any amount of testing — from a simple password-cracking test to an all-out penetration test on a Web application.

Formulating your plan

Approval for ethical hacking is essential. Make what you're doing known and visible — at least to the decision makers. Obtaining *sponsorship* of the project is the first step. This could be your manager, an executive, a customer, 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 claims they never authorized you to perform the tests.

The authorization can be as simple as an internal memo from your boss if you're performing these tests on your own systems. If you're testing for a customer, have a signed contract in place, stating the customer'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.

You need a detailed plan, but that doesn't mean you have to have volumes of testing procedures. One slip can crash your systems — not necessarily what anyone wants. A well-defined scope includes the following information:

- Specific systems to be tested
- Risks that are involved
- When the tests are performed and your overall timeline
- How the tests are performed
- How much knowledge of the systems you have before you start testing
- What is done when a major vulnerability is discovered
- The specific deliverables — this includes security-assessment reports and a higher-level report outlining the general vulnerabilities to be addressed, along with countermeasures that should be implemented

When selecting systems to test, start with the most critical or vulnerable systems. For instance, you can test computer passwords or attempt social-engineering attacks before drilling down into more detailed systems.

It pays to 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 could cause loss of data integrity, loss of data, and bad publicity.

Handle social-engineering and denial-of-service attacks carefully. Determine how they can affect the systems you're testing and your entire organization.

Determining when the tests are performed is something that you must think long and hard about. Do you test 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, wherein any type of test is possible. The bad guys aren't hacking your systems within a limited scope, so why should you? Some exceptions to this approach are performing DoS, social-engineering, and physical-security tests.

Don't stop with one security hole. This can lead to a false sense of security. 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).

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

You don't need extensive knowledge of the systems you're testing — just a basic understanding. This will help you protect 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 hacking a customer's systems, you may have to dig deeper. In fact, I've never had a customer ask for a fully blind assessment. Most people are scared of these assessments. Base the type of test you will perform on your organization's or customer's needs.

Chapter 19 covers hiring "reformed" hackers.

Selecting tools

As with any project, if you don't have the right tools for ethical hacking, accomplishing the task effectively is difficult. Having said that, just because you use the right tools doesn't mean that you will discover all vulnerabilities.

Know the personal and technical limitations. Many security-assessment tools generate false positives and negatives (incorrectly identifying vulnerabilities). Others may miss vulnerabilities. If you're performing tests such as social-engineering or physical-security assessments, you may miss weaknesses.

Many tools focus on specific tests, but no one tool can test for everything. For the same reason that you wouldn't drive in a nail with a screwdriver, you shouldn't use a word processor to scan your network for open ports. This is why you need a set of specific tools that you can call on for the task at hand. The more tools you have, the easier your ethical hacking efforts are.

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

- To crack passwords, you need a cracking tool such as LC4, John the Ripper, or pwdump.

  A general port scanner, such as SuperScan, may not crack passwords.

- For an in-depth analysis of a Web application, a Web-application assessment tool (such as Whisker or WebInspect) is more appropriate than a network analyzer (such as Ethereal).

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 (`www.google.com`) or perusal of security portals, such as SecurityFocus.com, SearchSecurity.com, and ITsecurity.com, often produces great feedback from other security experts.

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

- Nmap
- EtherPeek
- SuperScan
- QualysGuard
- WebInspect
- LC4 (formerly called L0phtcrack)
- LANguard Network Security Scanner
- Network Stumbler
- ToneLoc

Here are some other popular tools:

- Internet Scanner
- Ethereal
- Nessus
- Nikto
- Kismet
- THC-Scan

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 shed negative light on some excellent tools, such as SATAN (Security Administrator Tool for Analyzing Networks) and Nmap (Network Mapper).

Some of these 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 for your tools.
- Study the user's guide for your commercial tools.
- 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 may be exploited and fixed.
- Updates and support when needed.
- High-level reports that can be presented to managers or nontechie types.

These features can save you time and effort when you're writing the report.

Executing the plan

Ethical hacking can take 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 may watch what's going on. This person could use this information against you.

It's not practical to make sure that no hackers are on your systems before you start. Just make sure you keep everything as quiet and private as possible. This is especially critical when transmitting and storing your test results. If possible, encrypt these e-mails and files using Pretty Good Privacy (PGP) or something similar. At a minimum, password-protect them.

You're now on a reconnaissance mission. Harness as much information as possible about your organization and systems, which is what 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 for this.

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

   Whether physical-security structures or Web applications, a casual assessment can turn up much information about your systems.

3. **Further narrow your focus with a more critical eye. Perform actual scans and other detailed tests on your systems.**

4. **Perform the attacks, if that's what you choose to do.**

Evaluating results

Assess your results to see what you uncovered, assuming that the vulnerabilities haven't been made obvious before now. This is where knowledge counts. Evaluating the results and correlating the specific vulnerabilities discovered is a skill that gets better with experience. You'll end up knowing your systems as well as anyone else. This makes the evaluation process much simpler moving forward.

Submit a formal report to upper management or to your customer, outlining your results. Keep these other parties in the loop to show that your efforts and their money are well spent. Chapter 17 describes this process.

Moving on

When you've finished your ethical hacking tests, you still need to implement your analysis and recommendations to make sure your systems are secure.

New security vulnerabilities continually appear. Information systems constantly change and become more complex. New hacker exploits and security vulnerabilities are regularly uncovered. You may discover new ones! Security tests are a snapshot of the security posture of your systems. At any time, everything can change, especially after software upgrades, adding computer systems, or applying patches. Plan to test regularly (for example, once a week or once a month). Chapter 19 covers managing security changes.