Making Everything Easier!™

4th Edition

Hacking

FOR DUMMIES®

Learn to:

- Defend against the latest Windows® 8 and Linux® hacks
- Develop an effective ethical hacking plan
- Protect web applications, databases, laptops, and smartphones
- Use the latest testing tools and techniques

Kevin Beaver, CISSP

Independent Information Security Consultant

Hacking For Dummies, 4th Edition®

Chapter 7: Passwords

ISBN: 978-1-118-38093-2

WILEY

Copyright of John Wiley & Sons, Inc.
Hoboken, NJ

Posted with Permission

Chapter 7

Passwords

In This Chapter

- Identifying password vulnerabilities
- Examining password-hacking tools and techniques
- Hacking operating system passwords
- Hacking password-protected files
- Protecting your systems from password hacking

Password hacking is one of the easiest and most common ways attackers obtain unauthorized network, computer, or application access. You often hear about it in the headlines, and study after study such as the *Verizon Data Breach Investigations Report* reaffirms that weak passwords are at the root of many security problems. I have trouble wrapping my head around the fact that I'm *still* talking about (and suffering from) weak passwords, but it's a reality — and, as an information security testing professional, you can certainly do your part to minimize the risks.

Although strong passwords — ideally, longer and stronger passphrases that are difficult to *crack* (or guess) — are easy to create and maintain, network administrators and users often neglect this. Therefore, passwords are one of the weakest links in the information security chain. Passwords rely on secrecy. After a password is compromised, its original owner isn't the only person who can access the system with it. That's when accountability goes out the window and bad things start happening.

External attackers and malicious insiders have many ways to obtain passwords. They can glean passwords simply by asking for them or by looking over the shoulders of users *(shoulder surfing)* while they type their passwords. Hackers can also obtain passwords from local computers by using password-cracking software. To obtain passwords from across a network, attackers can use remote cracking utilities, keyloggers, or network analyzers.

This chapter demonstrates how easily the bad guys can gather password information from your network and computer systems. I outline common password vulnerabilities and describe countermeasures to help prevent these vulnerabilities from being exploited on your systems. If you perform the tests and implement the countermeasures outlined in this chapter, you'll be well on your way to securing your systems' passwords.

Understanding Password Vulnerabilities

When you balance the cost of security and the value of the protected information, the combination of a *user ID* and a *secret password* is usually adequate. However, passwords give a false sense of security. The bad guys know this and attempt to crack passwords as a step toward breaking into computer systems.

One big problem with relying solely on passwords for information security is that more than one person can know them. Sometimes, this is intentional; often, it's not. The tough part is that there's no way of knowing who, besides the password's owner, knows a password.

Remember that knowing a password doesn't make someone an authorized user.

Here are the two general classifications of password vulnerabilities:

- **Organizational or user vulnerabilities:** This includes lack of password policies that are enforced within the organization and lack of security awareness on the part of users.
- **Technical vulnerabilities:** This includes weak encryption methods and unsecure storage of passwords on computer systems.

I explore each of these classifications in more detail in the following sections.

Before computer networks and the Internet, the user's physical environment was an additional layer of password security that actually worked pretty well. Now that most computers have network connectivity, that protection is gone. Refer to Chapter 6 for details on managing physical security in this age of networked computers and mobile devices.

Organizational password vulnerabilities

It's human nature to want convenience, especially when it comes to remembering five, ten, and often dozens of passwords for work and daily life. This desire for convenience makes passwords one of the easiest barriers for an attacker to overcome. Almost 3 trillion (yes, trillion with a *t* and 12 zeros) eight-character password combinations are possible by using the 26 letters of the alphabet and the numerals 0 through 9. The keys to strong passwords are: 1) easy to remember and 2) difficult to crack. However, most people just focus on the easy-to-remember part. Users like to use such passwords as *password,* their login name, *abc123*, or no password at all! Don't laugh; I've seen these blatant weaknesses and guarantee they're on any given network this very moment.

A case study in Windows password vulnerabilities with Dr. Philippe Oechslin

In this case study, Dr. Philippe Oechslin, a researcher and independent information security consultant, shared with me his recent research findings on Windows password vulnerabilities.

The Situation

In 2003, Dr. Oechslin discovered a new method for cracking Windows passwords — now commonly referred to as *rainbow cracking.* While testing a brute-force password-cracking tool, Dr. Oechslin thought that everyone using the same tool to generate the same *hashes* (cryptographic representations of passwords) repeatedly was a waste of time. He believed that generating a huge dictionary of all possible hashes would make it easier to crack Windows passwords but then quickly realized that a dictionary of the LAN Manager (LM) hashes of all possible alphanumerical passwords would require over a terabyte of storage.

During his research, Dr. Oechslin discovered a technique called *time-memory trade-offs,* where hashes are computed in advance, but only a small fraction are stored (approximately one in a thousand). Dr. Oechslin discovered that how the LM hashes are organized allows you to find any password if you spend some time recalculating some of the hashes. This technique saves memory but takes a lot of time. Studying this method, Dr. Oechslin found a way to make the process more efficient, making it possible to find any of the 80 billion unique hashes by using a table of 250 million entries (1GB worth of data) and performing only 4 million hash calculations. This process is much faster than a brute-force attack, which must generate 50 percent of the hashes (40 billion) on average.

This research is based on the absence of a random element when Windows passwords are hashed. This is true for both the LM hash and the NTLM hash built in to Windows. As a result, the same password produces the same hash on any Windows machine. Although it is known that Windows hashes have no random element, no one has used a technique like the one that Dr. Oechslin discovered to crack Windows passwords.

Dr. Oechslin and his team originally placed an interactive tool on their website (`http://lasecwww.epfl.ch`) that enabled visitors to submit hashes and have them cracked. Over a six-day period, the tool cracked 1,845 passwords in an average of 7.7 seconds! You can try out the demo for yourself at `www.objectif-securite.ch/en/products.php`.

The Outcome

So what's the big deal, you say? This password-cracking method can crack practically any alphanumeric password in a few seconds, whereas current brute-force tools can take several hours. Dr. Oechslin and his research team have generated a table with which they can crack any password made of letters, numbers, and 16 other characters in less than a minute, demonstrating that passwords made up of letters and numbers aren't good enough (and thus should not exist in your environment). Dr. Oechslin also stated that this method is useful for ethical hackers who have only limited time to perform their testing. Unfortunately, malicious hackers have the same benefit and can perform their attacks before anyone detects them!

Philippe Oechslin, PhD, CISSP, is a lecturer and senior research assistant at the Swiss Federal Institute of Technology in Lausanne and is founder and CEO of Objectif Sécurité (`www.objectif-securite.ch/en`).

Unless users are educated and reminded about using strong passwords, their passwords usually are

- **Easy to guess.**
- **Seldom changed.**
- **Reused for many security points.** When bad guys crack one password, they can often access other systems with that same password and username.

  Using the same password across multiple systems and websites is nothing but a breach waiting to happen. Everyone is guilty of it, but that doesn't make it right. Do what you can to protect your own credentials and spread the word to your users about how this practice can get you into a real bind.
- **Written down in unsecure places.** The more complex a password is, the more difficult it is to crack. However, when users create complex passwords, they're more likely to write them down. External attackers and malicious insiders can find these passwords and use them against you and your business.

Technical password vulnerabilities

You can often find these serious technical vulnerabilities after exploiting organizational password vulnerabilities:

- **Weak password encryption schemes.** Hackers can break weak password storage mechanisms by using cracking methods that I outline in this chapter. Many vendors and developers believe that passwords are safe as long as they don't publish the source code for their encryption algorithms. *Wrong!* A persistent, patient attacker can usually crack this *security by obscurity* (a security measure that's hidden from plain view but can be easily overcome) fairly quickly. After the code is cracked, it is distributed across the Internet and becomes public knowledge.

  Password-cracking utilities take advantage of weak password encryption. These utilities do the grunt work and can crack any password, given enough time and computing power.
- **Programs that store their passwords in memory, unsecured files, and easily accessed databases.**
- **Unencrypted databases that provide direct access to sensitive information to anyone with database access, regardless of whether they have a business need to know.**
- **User applications that display passwords on the screen while the user is typing.**

The National Vulnerability Database (an index of computer vulnerabilities managed by the National Institute of Standards and Technology) currently identifies over 2,500 password-related vulnerabilities! You can search for these issues at `http://nvd.nist.gov` to find out how vulnerable some of your systems are from a technical perspective.

Cracking Passwords

Password cracking is one of the most enjoyable hacks for the bad guys. It fuels their sense of exploration and desire to figure out a problem. You might not have a burning desire to explore everyone's passwords, but it helps to approach password cracking with this mindset. So where should you start hacking the passwords on your systems? Generally, any user's password works. After you obtain one password, you can often obtain others — including administrator or root passwords.

Administrator passwords are the pot of gold. With unauthorized administrative access, you (or a criminal hacker) can do virtually anything on the system. When looking for your organization's password vulnerabilities, I recommend first trying to obtain the highest level of access possible (such as administrator) through the most discreet method possible. That's often what the bad guys do.

You can use low-tech ways and high-tech ways to exploit vulnerabilities to obtain passwords. For example, you can deceive users into divulging passwords over the telephone or simply observe what a user has written down on a piece of paper. Or you can capture passwords directly from a computer, over a network, and via the Internet with the tools covered in the following sections.

Cracking passwords the old-fashioned way

A hacker can use low-tech methods to crack passwords. These methods include using social engineering techniques, shoulder surfing, and simply guessing passwords from information that he knows about the user.

Social engineering

The most popular low-tech method for gathering passwords is *social engineering,* which I cover in detail in Chapter 5. Social engineering takes advantage of the trusting nature of human beings to gain information that later can be used maliciously. A common social engineering technique is simply to con people into divulging their passwords. It sounds ridiculous, but it happens all the time.

Techniques

To obtain a password through social engineering, you just ask for it. For example, you can simply call a user and tell him that he has some important-looking e-mails stuck in the mail queue, and you need his password to log in and free them up. This is often how hackers and rogue insiders try to get the information!

REMEMBER

If a user gives you his password during your testing, make sure that he changes it. You don't want to be held accountable if something goes awry after the password has been disclosed.

A common weakness that can facilitate such social engineering is when staff members' names, phone numbers, and e-mail addresses are posted on your company websites. Social media sites such as LinkedIn, Facebook, and Twitter can also be used against a company because these sites can reveal employees' names and contact information.

Countermeasures

User awareness and consistent security training are great defenses against social engineering. Security tools are a good fail-safe if they monitor for such e-mails and web browsing at the host-level, network perimeter, or in the cloud. Train users to spot attacks (such as suspicious phone calls or deceitful phishing e-mails) and respond effectively. Their best response is not to give out any information and to alert the appropriate information security manager in the organization to see whether the inquiry is legitimate and whether a response is necessary. Oh, and take that staff directory off your website or at least remove IT staff members' information.

Shoulder surfing

Shoulder surfing (the act of looking over someone's shoulder to see what the person is typing) is an effective, low-tech password hack.

Techniques

To mount this attack, the bad guys must be near their victims and not look obvious. They simply collect the password by watching either the user's keyboard or screen when the person logs in. An attacker with a good eye might even watch whether the user is glancing around his desk for either a reminder of the password or the password itself. Security cameras or a webcam can even be used for such attacks. Coffee shops and airplanes provide the ideal scenarios for shoulder surfing.

You can try shoulder surfing yourself. Simply walk around the office and perform random spot checks. Go to users' desks and ask them to log in to their computers, the network, or even their e-mail applications. Just don't tell them what you're doing beforehand, or they might attempt to hide what they're typing or where they're looking for their password — two things that they should've been doing all along! Just be careful doing this and respect other people's privacy.

Countermeasures

Encourage users to be aware of their surroundings and not to enter their passwords when they suspect that someone is looking over their shoulders. Instruct users that if they suspect someone is looking over their shoulders while they're logging in, they should politely ask the person to look away or, when necessary, hurl an appropriate epithet to show the offender that the user is serious. It's often easiest to just lean into the shoulder surfer's line of sight to keep them from seeing any typing and/or the computer screen. 3M Privacy Filters (`www.shop3m.com/3m-privacy-filters.html`) work great as well yet, surprisingly, I rarely see them being used.

Inference

Inference is simply guessing passwords from information you know about users — such as their date of birth, favorite television show, or phone numbers. It sounds silly, but criminals often determine their victims' passwords simply by guessing them!

The best defense against an inference attack is to educate users about creating secure passwords that don't include information that can be associated with them. Outside of certain password complexity filters, it's often not easy to enforce this practice with technical controls. So, you need a sound security policy and ongoing security awareness and training to remind users of the importance of secure password creation.

Weak authentication

External attackers and malicious insiders can obtain — or simply avoid having to use — passwords by taking advantage of older or unsecured operating systems that don't require passwords to log in. The same goes for a phone or tablet that isn't configured to use passwords.

Bypassing authentication

On older operating systems (such as Windows 9*x*) that prompt for a password, you can press Esc on the keyboard to get right in. Okay, it's hard to find any Windows 9*x* systems these days, but the same goes for any operating system — old or new — that's configured to bypass the login screen. After you're in, you can find other passwords stored in such places as dialup and VPN connections and screen savers. Such passwords can be cracked very easily using Elcomsoft's Proactive System Password Recovery tool (`www.elcomsoft.com/pspr.html`) and Cain & Abel (`www.oxid.it/cain.html`). These weak systems can serve as *trusted* machines — meaning that people assume they're secure — and provide good launching pads for network-based password attacks as well.

Countermeasures

The only true defense against weak authentication is to ensure your operating systems require a password upon boot. To eliminate this vulnerability, *at least* upgrade to Windows 7 or 8 or use the most recent versions of Linux or one of the various flavors of UNIX, including Mac OS X.

TIP

More modern authentication systems, such as Kerberos (which is used in newer versions of Windows) and directory services (such as Microsoft's Active Directory), encrypt user passwords or don't communicate the passwords across the network at all, which creates an extra layer of security.

Cracking passwords with high-tech tools

High-tech password cracking involves using a program that tries to guess a password by determining all possible password combinations. These high-tech methods are mostly automated after you access the computer and password database files.

The main password-cracking methods are dictionary attacks, brute-force attacks, and rainbow attacks. You find out how each of these work in the following sections.

Password-cracking software

You can try to crack your organization's operating system and application passwords with various password-cracking tools:

- **Brutus** (`www.hoobie.net/brutus`) cracks logons for HTTP, FTP, telnet, and more.
- **Cain & Abel** (`www.oxid.it/cain.html`) cracks LM and NT LanManager (NTLM) hashes, Windows RDP passwords, Cisco IOS and PIX hashes, VNC passwords, RADIUS hashes, and lots more. (*Hashes* are cryptographic representations of passwords.)
- **Elcomsoft Distributed Password Recovery** (`www.elcomsoft.com/edpr.html`) cracks Windows, Microsoft Office, PGP, Adobe, iTunes, and numerous other passwords in a distributed fashion using up to 10,000 networked computers at one time. Plus, this tool uses the same graphics processing unit (GPU) video acceleration as the Elcomsoft Wireless Auditor tool, which allows for cracking speeds up to 50 times faster. (I talk about the Elcomsoft Wireless Auditor tool in Chapter 9.)
- **Elcomsoft System Recovery** (`www.elcomsoft.com/esr.html`) cracks or resets Windows user passwords, sets administrative rights, and resets password expirations all from a bootable CD.

- **John the Ripper** (www.openwall.com/john) cracks hashed Linux/UNIX and Windows passwords.
- **ophcrack** (http://ophcrack.sourceforge.net) cracks Windows user passwords using rainbow tables from a bootable CD. *Rainbow tables* are pre-calculated password hashes that can help speed up the cracking process. See the nearby sidebar "A case study in Windows password vulnerabilities with Dr. Philippe Oechslin" for more information.
- **Proactive Password Auditor** (www.elcomsoft.com/ppa.html) runs brute-force, dictionary, and rainbow cracks against extracted LM and NTLM password hashes.
- **Proactive System Password Recovery** (www.elcomsoft.com/pspr.html) recovers practically any locally stored Windows password, such as logon passwords, WEP/WPA passphrases, SYSKEY passwords, and RAS/dialup/VPN passwords.
- **pwdump3** (www.openwall.com/passwords/microsoft-windows-nt-2000-xp-2003-vista-7#pwdump) extracts Windows password hashes from the SAM (Security Accounts Manager) database.
- **RainbowCrack** (http://project-rainbowcrack.com) cracks LanManager (LM) and MD5 hashes very quickly by using rainbow tables.
- **THC-Hydra** (www.thc.org/thc-hydra) cracks logons for HTTP, FTP, IMAP, SMTP, VNC and many more.

TIP

Some of these tools require physical access to the systems you're testing. You might be wondering what value that adds to password cracking. If a hacker can obtain physical access to your systems and password files, you have more than just basic information security problems to worry about, right? True, but this kind of access is entirely possible! What about a summer intern, a disgruntled employee, or an outside auditor with malicious intent? The mere risk of an unencrypted laptop being lost or stolen and falling into the hands of someone with ill intent should be reason enough.

To understand how the preceding password-cracking programs generally work, you first need to understand how passwords are encrypted. Passwords are typically encrypted when they're stored on a computer, using an encryption or one-way hash algorithm, such as DES or MD5. Hashed passwords are then represented as fixed-length encrypted strings that always represent the same passwords with exactly the same strings. These hashes are irreversible for all practical purposes, so, in theory, passwords can never be decrypted. Furthermore, certain passwords, such as those in Linux, have a random value called a *salt* added to them to create a degree of randomness. This prevents the same password used by two people from having the same hash value.

Password-cracking utilities take a set of known passwords and run them through a password-hashing algorithm. The resulting encrypted hashes are then compared at lightning speed to the password hashes extracted from the original password database. When a match is found between the newly generated hash and the hash in the original database, the password has been cracked. It's that simple.

Other password-cracking programs simply attempt to log on using a predefined set of user IDs and passwords. This is how many dictionary-based cracking tools work, such as Brutus (`www.hoobie.net/brutus`) and SQLPing3 (`www.sqlsecurity.com/downloads`). I cover cracking web application and database passwords in Chapters 14 and 15.

Passwords that are subjected to cracking tools eventually lose. You have access to the same tools as the bad guys. These tools can be used for both legitimate security assessments and malicious attacks. You want to find password weaknesses before the bad guys do, and in this section, I show you some of my favorite methods for assessing Windows and Linux/UNIX passwords.

WARNING!

When trying to crack passwords, the associated user accounts might be locked out, which could interrupt your users. Be careful if intruder lockout is enabled in your operating systems, databases, or applications. If lockout is enabled, you might lock out some or all computer/network accounts, resulting in a denial of service situation for your users.

Password storage locations vary by operating system:

- Windows usually stores passwords in these locations:
    - Security Accounts Manager (SAM) database (`c:\winnt\system32\config`) or (`c:\windows\system32\config`)
    - Active Directory database file that's stored locally or spread across domain controllers (`ntds.dit`)

  Windows may also store passwords in a backup of the SAM file in the `c:\winnt\repair` or `c:\windows\repair` directory.

  WARNING!

  Some Windows applications store passwords in the Registry or as plain-text files on the hard drive! A simple registry or file-system search for "password" may uncover just what you're looking for.
- Linux and other UNIX variants typically store passwords in these files:
    - `/etc/passwd` (readable by everyone)
    - `/etc/shadow` (accessible by the system and the root account only)
    - `/etc/security/passwd` (accessible by the system and the root account only)
    - `/.secure/etc/passwd` (accessible by the system and the root account only)

Dictionary attacks

Dictionary attacks quickly compare a set of known dictionary-type words — including many common passwords — against a password database. This database is a text file with hundreds if not thousands of dictionary words typically listed in alphabetical order. For instance, suppose that you have a dictionary file that you downloaded from one of the sites in the following list. The English dictionary file at the Purdue site contains one word per line starting with *10th, 1st* . . . all the way to *zygote.*

Many password-cracking utilities can use a separate dictionary that you create or download from the Internet. Here are some popular sites that house dictionary files and other miscellaneous word lists:

- `ftp://ftp.cerias.purdue.edu/pub/dict`
- `www.outpost9.com/files/WordLists.html`

Don't forget to use other language files as well, such as Spanish and Klingon.

REMEMBER

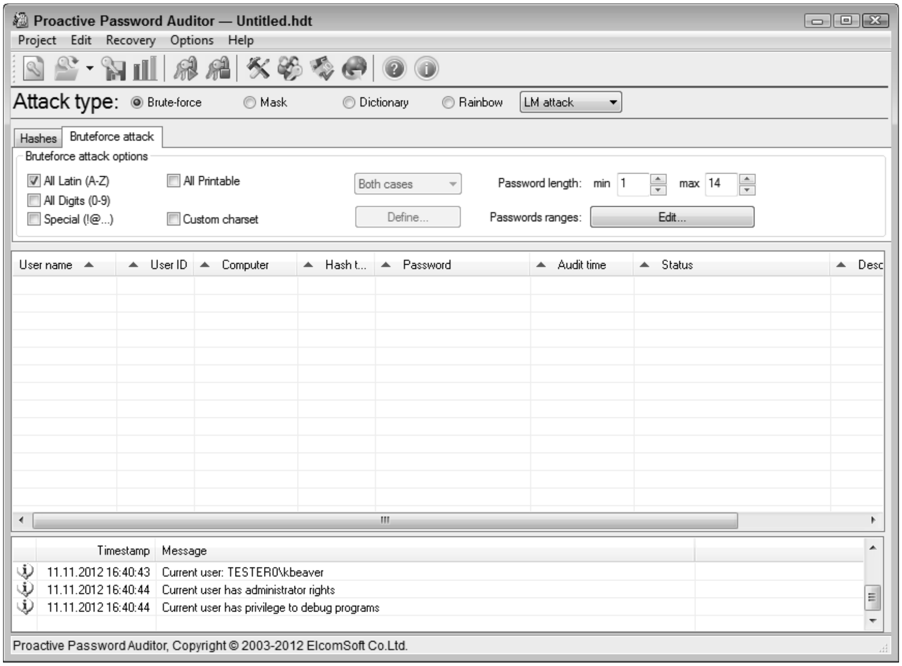
Dictionary attacks are only as good as the dictionary files you supply to your password-cracking program. You can easily spend days, even weeks, trying to crack passwords with a dictionary attack. If you don't set a time limit or similar expectation going in, you'll likely find that dictionary cracking is often a mere exercise in futility. Most dictionary attacks are good for *weak* (easily guessed) passwords. However, some special dictionaries have common misspellings or alternative spellings of words, such as `pa$$w0rd` (password) and `5ecur1ty` (security). Additionally, special dictionaries can contain non-English words and thematic words from religions, politics, or *Star Trek.*

Brute-force attacks

Brute-force attacks can crack practically any password, given sufficient time. Brute-force attacks try every combination of numbers, letters, and special characters until the password is discovered. Many password-cracking utilities let you specify such testing criteria as the character sets, password length to try, and known characters (for a "mask" attack). Sample Proactive Password Auditor brute-force password-cracking options are shown in Figure 7-1.

WARNING!

A brute-force test can take quite a while, depending on the number of accounts, their associated password complexities, and the speed of the computer that's running the cracking software. As powerful as brute-force testing can be, it literally can take forever to exhaust all possible password combinations, which in reality is not practical in every situation.



Attack type: ● Brute-force ○ Mask ○ Dictionary ○ Rainbow LM attack

Bruteforce attack options

☑ All Latin (A-Z) ☐ All Printable Both cases Password length: min 1 max 14
☐ All Digits (0-9)
☐ Special (!@...) ☐ Custom charset Passwords ranges:

Timestamp	Message
11.11.2012 16:40:43	Current user: TESTER0\kbeaver
11.11.2012 16:40:44	Current user has administrator rights
11.11.2012 16:40:44	Current user has privilege to debug programs

Proactive Password Auditor, Copyright © 2003-2012 ElcomSoft Co.Ltd.

Figure 7-1: Brute-force password-cracking options in Proactive Password Auditor.

Smart hackers attempt logins slowly or at random times so the failed login attempts aren't as predictable or obvious in the system log files. Some malicious users might even call the IT help desk to attempt a reset of the account they just locked out. This social engineering technique could be a major issue, especially if the organization has no (or minimal) mechanisms in place to verify that locked-out users are who they say they are.

Can an expiring password deter a hacker's attack and render password-cracking software useless? Yes. After the password is changed, the cracking must start again if the hacker wants to test all the possible combinations. This is one reason why it's a good idea to change passwords periodically. Shortening the change interval can reduce the risk of passwords being cracked but can also be politically unfavorable in your business. You have to strike a balance between security and convenience/usability. Refer to the United States Department of Defense's Password Management Guideline document (`www.itl.nist.gov/fipspubs/app-e.htm`) for more information on this topic.

Exhaustive password-cracking attempts usually aren't necessary. Most passwords are fairly weak. Even minimum password requirements, such as a password length, can help you in your testing. You might be able to discover security policy information by using other tools or via your web browser. (See Part IV for tools and techniques for testing the security of operating systems. See Chapter 14 for information on testing websites/applications.) If you find this password policy information, you can configure your cracking programs with more well-defined cracking parameters, which often generate faster results.

Rainbow attacks

A rainbow password attack uses rainbow cracking (see the earlier sidebar, "A case study in Windows password vulnerabilities with Dr. Philippe Oechslin") to crack various password hashes for LM, NTLM, Cisco PIX, and MD5 much more quickly and with extremely high success rates (near 100 percent). Password-cracking speed is increased in a rainbow attack because the hashes are precalculated and thus don't have to be generated individually on the fly as they are with dictionary and brute-force cracking methods.

Unlike dictionary and brute-force attacks, rainbow attacks cannot be used to crack password hashes of unlimited length. The current maximum length for Microsoft LM hashes is 14 characters, and the maximum is up to 16 characters (dictionary-based) for Windows Vista and 7 hashes (also known as NT hashes). The rainbow tables are available for purchase and download via the ophcrack site at `http://ophcrack.sourceforge.net`. There's a length limitation because it takes *significant* time to generate these rainbow tables. Given enough time, a sufficient number of tables will be created. Of course, by then, computers and applications likely have different authentication mechanisms and hashing standards — including a new set of vulnerabilities — to contend with. Job security for ethical hacking never ceases to grow.

If you have a good set of rainbow tables, such as those offered via the ophcrack site and Project RainbowCrack (`http://project-rainbowcrack.com`), you can crack passwords in seconds, minutes, or hours versus the days, weeks, or even years required by dictionary and brute-force methods.

Cracking Windows passwords with pwdump3 and John the Ripper

The following steps use two of my favorite utilities to test the security of current passwords on Windows systems:

- pwdump3 (to extract password hashes from the Windows SAM database)
- John the Ripper (to crack the hashes of Windows and Linux/UNIX passwords)

The following test requires administrative access to either your Windows standalone workstation or the server:

1. **Create a new directory called `passwords` from the root of your Windows C: drive.**
2. **Download and install a decompression tool if you don't already have one.**

WinZip (`www.winzip.com`) is a good commercial tool I use and 7-Zip (`www.7-zip.org`) is a free decompression tool. Windows XP, Windows Vista, and Windows 7 also include built-in Zip file handling.

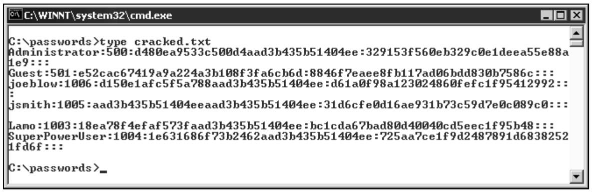
3. **Download, extract, and install the following software into the `passwords` directory you created, if you don't already have it on your system:**

   - *pwdump3:* Download the file from `www.openwall.com/passwords/microsoft-windows-nt-2000-xp-2003-vista-7#pwdump`
   - *John the Ripper:* Download the file from `www.openwall.com/john`

4. **Enter the following command to run pwdump3 and redirect its output to a file called `cracked.txt`:**

   ```
   c:\passwords\pwdump3 > cracked.txt
   ```

   This file captures the Windows SAM password hashes that are cracked with John the Ripper. Figure 7-2 shows the contents of the `cracked.txt` file that contains the local Windows SAM database password hashes.



C:\WINNT\system32\cmd.exe

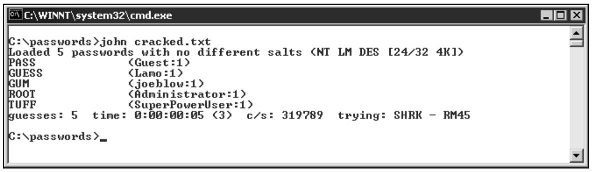
```
C:\passwords>type cracked.txt
Administrator:500:d480ea9533c500d4aad3b435b51404ee:329153f560eb329c0e1deea55e88a
1e9:::
Guest:501:e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaee8fb117ad06bdd830b7586c:::
joeblow:1006:d150e1afc5f5a788aad3b435b51404ee:d61a0f98a123024860fefc1f95412992::
:
jsmith:1005:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::

Lamo:1003:18ea78f4efaf573faad3b435b51404ee:bc1cda67bad80d40040cd5eec1f95b48:::
SuperPowerUser:1004:1e631686f73b2462aad3b435b51404ee:725aa7ce1f9d2487891d6838252
1fd6f:::

C:\passwords>_
```

Figure 7-2: Output from pwdump3.

5. **Enter the following command to run John the Ripper against the Windows SAM password hashes to display the cracked passwords:**

   ```
   c:\passwords\john cracked.txt
   ```

   This process — shown in Figure 7-3 — can take seconds or days, depending on the number of users and the complexity of their associated passwords. My Windows example took only five seconds to crack five weak passwords.



C:\WINNT\system32\cmd.exe

```
C:\passwords>john cracked.txt
Loaded 5 passwords with no different salts (NT LM DES [24/32 4K])
PASS                 (Guest:1)
GUESS                (Lamo:1)
GUM                  (joeblow:1)
ROOT                 (Administrator:1)
TUFF                 (SuperPowerUser:1)
guesses: 5  time: 0:00:00:05 (3)  c/s: 319789  trying: SHRK - RM45

C:\passwords>_
```

Figure 7-3: Cracked password file hashes using John the Ripper.

Cracking UNIX/Linux passwords with John the Ripper

John the Ripper can also crack UNIX/Linux passwords. You need root access to your system and to the password (`/etc/passwd`) and shadow password (`/etc/shadow`) files. Perform the following steps for cracking UNIX/Linux passwords:

1. **Download the UNIX source files from** `www.openwall.com/john`**.**
2. **Extract the program by entering the following command:**

   ```
   [root@localhost kbeaver]#tar -zxf john-1.7.9.tar.gz
   ```

   or whatever the current filename is.

   TIP: You can also crack UNIX or Linux passwords on a Windows system by using the Windows/DOS version of John the Ripper.
3. **Change to the** `/src` **directory that was created when you extracted the program and enter the following command:**

   ```
   make generic
   ```

4. **Change to the** `/run` **directory and enter the following command to use the unshadow program to combine the** `passwd` **and** `shadow` **files and copy them to the file** `cracked.txt`**:**

   ```
   ./unshadow /etc/passwd /etc/shadow > cracked.txt
   ```

   WARNING! The unshadow process won't work with all UNIX variants.
5. **Enter the following command to start the cracking process:**

   ```
   ./john cracked.txt
   ```

   When John the Ripper is complete (and this could take some time), the output is similar to the results of the preceding Windows process. (Refer to Figure 7-3.)

After completing the preceding Windows or UNIX steps, you can either force users to change passwords that don't meet specific password policy requirements, you can create a new password policy, or you can use the information to update your security awareness program. Just do something.

WARNING! Be careful handling the results of your password cracking. You create an accountability issue because more than one person now knows the passwords. Always treat the password information of others as strictly confidential. If you end up storing them on your test system, make sure it's extra secure. If it's a laptop, encrypting the hard drive is the best defense.

Passwords by the numbers

One hundred twenty-eight different ASCII characters are used in typical computer passwords. (Technically, only 126 characters are used because you can't use the NULL and the carriage return characters.) A truly random eight-character password that uses 126 different characters can have 63,527,879,748,485,376 different combinations. Taking that a step further, if it were possible (and it is in Linux and UNIX) to use all 256 ASCII characters (254, without NULL and carriage return characters) in a password, 17,324,859,965,700,833,536 different combinations would be available. This is approximately 2.7 billion times more combinations than there are people on earth!

A text file containing all the possible passwords would require millions of terabytes of storage space. Even if you include only the more realistic combination of 95 or so ASCII letters, numbers, and standard punctuation characters, such a file would still fill thousands of terabytes of storage space. These storage requirements force dictionary and brute-force password-cracking programs to form the password combinations on the fly, instead of reading all possible combinations from a text file. That's why rainbow attacks are more effective at cracking passwords than dictionary and brute-force attacks.

Given the effectiveness of rainbow password attacks, it's realistic to think that eventually, anyone will be able to crack all possible password combinations, given the current technology and average lifespan. It probably won't happen; however, many thought in the 1980s that 640K of RAM and a 10MB hard drive in a PC were all that would ever be needed!

Cracking password-protected files

Do you wonder how vulnerable password-protected word-processing, spreadsheet, and Zip files are when users send them into the wild blue yonder? Wonder no more. Some great utilities can show how easily passwords are cracked.

Cracking files

Most password-protected files can be cracked in seconds or minutes. You can demonstrate this "wow factor" security vulnerability to users and management. Here's a hypothetical scenario that could occur in the real world:

1. Your CFO wants to send some confidential financial information in an Excel spreadsheet to a company board member.
2. She protects the spreadsheet by assigning it a password during the file-save process in Excel.
3. For good measure, she uses WinZip to compress the file and adds another password to make it *really* secure.
4. The CFO sends the spreadsheet as an e-mail attachment, assuming that the e-mail will reach its destination.

The financial advisor's network has content filtering, which monitors incoming e-mails for keywords and file attachments. Unfortunately, the financial advisory firm's network administrator is looking in the content-filtering system to see what's coming in.

5. This rogue network administrator finds the e-mail with the confidential attachment, saves the attachment, and realizes that it's password protected.
6. The network administrator remembers a great password-cracking tool available from Elcomsoft called Advanced Archive Password Recovery (`www.elcomsoft.com/archpr.html`) that can help him out so he proceeds to use it to crack the password.

Cracking password-protected files is as simple as that! Now all that the rogue network administrator must do is forward the confidential spreadsheet to his buddies or to the company's competitors.

TIP

If you carefully select the right options in Advanced Archive Password Recovery, you can drastically shorten your testing time. For example, if you know that a password is not over five characters long or is lowercase letters only, you can cut the cracking time in half.

I recommend performing these file-password-cracking tests on files that you capture with a content filtering or network analysis tool. This is a good way to determine whether your users are adhering to policy and using adequate passwords to protect sensitive information they're sending.

Countermeasures

The best defense against weak file password protection is to require your users to use a stronger form of file protection, such as PGP, or the AES encryption that's built in to WinZip, when necessary. Ideally, you don't want to rely on users to make decisions about what they should use to secure sensitive information, but it's better than nothing. Stress that a file encryption mechanism, such as a password-protected Zip file, is secure only if users keep their passwords confidential and never transmit or store them in unsecure cleartext (such as in a separate e-mail).

If you're concerned about unsecure transmissions through e-mail, consider using a content-filtering system or a data leak–prevention system to block all outbound e-mail attachments that aren't protected on your e-mail server.

Understanding other ways to crack passwords

Over the years, I've found other ways to crack (or capture) passwords technically and through social engineering.

Keystroke logging

One of the best techniques for capturing passwords is remote *keystroke logging* — the use of software or hardware to record keystrokes as they're typed into the computer.

WARNING! Be careful with keystroke logging. Even with good intentions, monitoring employees raises various legal issues if it's not done correctly. Discuss with your legal counsel what you'll be doing, ask for their guidance, and get approval from upper management.

Logging tools

With keystroke-logging tools, you can assess the log files of your application to see what passwords people are using:

- Keystroke-logging applications can be installed on the monitored computer. I recommend that you check out eBlaster and Spector Pro by SpectorSoft (`www.spectorsoft.com`). Another popular tool is Invisible KeyLogger Stealth, available at `www.amecisco.com/iks.htm`. Dozens of other such tools are available on the Internet.
- Hardware-based tools, such as KeyGhost (`www.keyghost.com`), fit between the keyboard and the computer or replace the keyboard altogether.

WARNING! A keystroke-logging tool installed on a shared computer can capture the passwords of every user who logs in.

Countermeasures

The best defense against the installation of keystroke-logging software on your systems is to use an anti-malware program or similar endpoint protection software that monitors the local host. It's not foolproof but can help. As for physical keyloggers, you'll need to visually inspect each system.

WARNING! The potential for hackers to install keystroke-logging software is another reason to ensure that your users aren't downloading and installing random shareware or opening attachments in unsolicited e-mails. Consider locking down your desktops by setting the appropriate user rights through local or group security policy in Windows. Alternatively, you could use a commercial lockdown program, such as Fortres 101 (`www.fortresgrand.com`) for Windows or Deep Freeze Enterprise (`www.faronics.com/products/deep-freeze/enterprise`) for Windows, Linux, and Mac OS X.

Weak password storage

Many legacy and standalone applications, such as e-mail, dial-up network connections, and accounting software, store passwords locally, making them vulnerable to password hacking. By performing a basic text search, I've found passwords stored in cleartext on the local hard drives of machines. You can automate the process even further by using a program called Identity Finder (www.identityfinder.com/us/Business). I cover these file and related storage vulnerabilities in Chapter 15.

Searching

You can try using your favorite text-searching utility — such as the Windows search function, findstr, or grep — to search for *password* or *passwd* on your computer's drives. You might be shocked to find what's on your systems. Some programs even write passwords to disk or leave them stored in memory.

Weak password storage is a criminal hacker's dream. Head it off if you can.

Countermeasures

The only reliable way to eliminate weak password storage is to use only applications that store passwords securely. This might not be practical, but it's your only guarantee that your passwords are secure. Another option is to instruct users not to store their passwords when prompted.

Before upgrading applications, contact your software vendor to see how they manage passwords, or search for a third-party solution.

Network analyzer

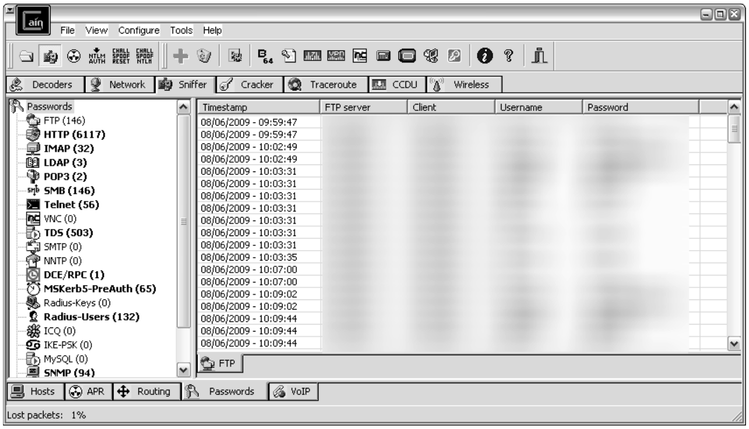
A network analyzer sniffs the packets traversing the network. This is what the bad guys do if they can gain control of a computer, tap into your wireless network, or gain physical network access to set up their network analyzer. If they gain physical access, they can look for a network jack on the wall and plug right in!

Testing

Figure 7-4 shows how crystal-clear passwords can be through the eyes of a network analyzer. This figure shows how Cain & Abel (www.oxid.it/cain.html) can glean thousands of passwords going across the network in a matter of a couple of hours. As you can see in the left pane, these cleartext password vulnerabilities can apply to FTP, web, telnet, and more. (The actual usernames and passwords are blurred out to protect them.)

If traffic is not tunneled through a VPN, SSH, SSL, or some other form of encrypted link, it's vulnerable to attack.

Cain & Abel is a password-cracking tool that also has network analysis capabilities. You can also use a regular network analyzer, such as the commercial products OmniPeek (www.wildpackets.com/products/omnipeek_network_analyzer) and CommView (www.tamos.com/products/commview) as well as the free open source program, Wireshark (www.wireshark.org). With a network analyzer, you can search for password traffic in various ways. For example, to capture POP3 password traffic, you can set up a filter and a trigger to search for the PASS command. When the network analyzer sees the PASS command in the packet, it captures that specific data.



- Passwords
  - FTP (146)
  - **HTTP (6117)**
  - **IMAP (32)**
  - **LDAP (3)**
  - **POP3 (2)**
  - **SMB (146)**
  - **Telnet (56)**
  - VNC (0)
  - **TDS (503)**
  - SMTP (0)
  - NNTP (0)
  - **DCE/RPC (1)**
  - **MSKerb5-PreAuth (65)**
  - Radius-Keys (0)
  - **Radius-Users (132)**
  - ICQ (0)
  - IKE-PSK (0)
  - MySQL (0)
  - **SNMP (94)**

Timestamp	FTP server	Client	Username	Password
08/06/2009 - 09:59:47	[redacted]	[redacted]	[redacted]	[redacted]
08/06/2009 - 09:59:47	[redacted]	[redacted]	[redacted]	[redacted]
08/06/2009 - 10:02:49	[redacted]	[redacted]	[redacted]	[redacted]
08/06/2009 - 10:02:49	[redacted]	[redacted]	[redacted]	[redacted]
08/06/2009 - 10:03:31	[redacted]	[redacted]	[redacted]	[redacted]
08/06/2009 - 10:03:31	[redacted]	[redacted]	[redacted]	[redacted]
08/06/2009 - 10:03:31	[redacted]	[redacted]	[redacted]	[redacted]
08/06/2009 - 10:03:31	[redacted]	[redacted]	[redacted]	[redacted]
08/06/2009 - 10:03:31	[redacted]	[redacted]	[redacted]	[redacted]
08/06/2009 - 10:03:31	[redacted]	[redacted]	[redacted]	[redacted]
08/06/2009 - 10:03:31	[redacted]	[redacted]	[redacted]	[redacted]
08/06/2009 - 10:03:35	[redacted]	[redacted]	[redacted]	[redacted]
08/06/2009 - 10:07:00	[redacted]	[redacted]	[redacted]	[redacted]
08/06/2009 - 10:07:00	[redacted]	[redacted]	[redacted]	[redacted]
08/06/2009 - 10:09:02	[redacted]	[redacted]	[redacted]	[redacted]
08/06/2009 - 10:09:02	[redacted]	[redacted]	[redacted]	[redacted]
08/06/2009 - 10:09:44	[redacted]	[redacted]	[redacted]	[redacted]
08/06/2009 - 10:09:44	[redacted]	[redacted]	[redacted]	[redacted]
08/06/2009 - 10:09:44	[redacted]	[redacted]	[redacted]	[redacted]

Figure 7-4: Using Cain & Abel to capture passwords going across the network.

Network analyzers require you to capture data on a hub segment of your network or via a monitor/mirror/span port on a switch. Otherwise, you can't see anyone else's data traversing the network — just yours. Check your switch's user guide for whether it has a monitor or mirror port and instructions on how to configure it. You can connect your network analyzer to a hub on the public side of your firewall. You'll capture only those packets that are entering or leaving your network — not internal traffic. I cover this type of network infrastructure hacking in detail in Chapter 8.

Countermeasures

Here are some good defenses against network analyzer attacks:

- **Use switches on your network, not hubs.** If you must use hubs on network segments, a program like sniffdet (`http://sniffdet.sourceforge.net`) for UNIX-based systems and PromiscDetect (`http://ntsecurity.nu/toolbox/promiscdetect`) for Windows can detect network cards in *promiscuous mode* (accepting all packets, whether destined for the local machine or not). A network card in promiscuous mode signifies that a network analyzer is running on the network.
- **Make sure that unsupervised areas, such as an unoccupied lobby or training room, don't have live network connections.**
- **Don't let anyone without a business need gain physical access to your switches or to the network connection on the public side of your firewall.** With physical access, a hacker can connect to a switch monitor port or tap into the unswitched network segment outside the firewall and capture packets.

WARNING!

Switches don't provide complete security because they're vulnerable to ARP poisoning attacks, which I cover in Chapter 8.

Weak BIOS passwords

Most computer BIOS (basic input/output system) settings allow power-on passwords and/or setup passwords to protect the computer's hardware settings that are stored in the CMOS chip. Here are some ways around these passwords:

- You can usually reset these passwords either by unplugging the CMOS battery or by changing a jumper on the motherboard.
- Password-cracking utilities for BIOS passwords are available on the Internet and from computer manufacturers.
- If gaining access to the hard drive is your ultimate goal, you can simply remove the hard drive from the computer and install it in another one and you're good to go. This is a great way to prove that BIOS/power-on passwords are *not* an effective countermeasure for lost or stolen laptops.

TIP

For a good list of default system passwords for various vendor equipment, check `www.cirt.net/passwords`.

There are tons of variables for hacking and hacking countermeasures depending on your hardware setup. If you plan to hack your own BIOS passwords, check for information in your user manual or refer to the BIOS password-hacking guide I wrote at `http://searchenterprisedesktop.techtarget.com/tutorial/BIOS-password-hacking`. If protecting the information on your hard drives is your ultimate goal, then full (sometimes referred to as *whole*) disk is the best way to go. I cover mobile-related password cracking in-depth in Chapter 10.

Weak passwords in limbo

Bad guys often exploit user accounts that have just been created or reset by a network administrator or help desk. New accounts might need to be created for new employees or even for your own ethical hacking purposes. Accounts might need to be reset if users forget their passwords or if the accounts have been locked out because of failed attempts.

Weaknesses

Here are some reasons why user accounts can be vulnerable:

- When user accounts are reset, they often are assigned an easily cracked password (such as the user's name or the word *password*). The time between resetting the user account and changing the password is a prime opportunity for a break-in.
- Many systems have either default accounts or unused accounts with weak passwords or no passwords at all. These are prime targets.

Countermeasures

The best defenses against attacks on passwords in limbo are solid help desk policies and procedures that prevent weak passwords from being available at *any* given time during the new account generation and password reset processes. Perhaps the best ways to overcome this vulnerability are as follows:

- Require users to be on the phone with the help desk, or have a help desk member perform the reset at the user's desk.
- Require that the user immediately log in and change the password.
- If you need the ultimate in security, implement stronger authentication methods, such as challenge/response questions, smart cards, or digital certificates.
- Automate password reset functionality via self-service tools on your network so users can manage most of their password problems without help from others.

I cover mobile-related password cracking in Chapter 10 and website/application password cracking in Chapter 14.

General Password-Cracking Countermeasures

A password for one system usually equals passwords for many other systems because many people use the same (or at least similar) passwords on every system they use. For this reason, you might want to consider instructing users to create different passwords for different systems, especially on the systems that protect information that's more sensitive. The only downside to this is that users have to keep multiple passwords and, therefore, might be tempted to write them down, which can negate any benefits.

TIP

Strong passwords are important, but you need to balance security and convenience:

- You can't expect users to memorize passwords that are insanely complex and must be changed every few weeks.
- You can't afford weak passwords or no passwords at all, so come up with a strong password policy and accompanying standard — preferably one that requires long and strong passphrases (combinations of words that are easily remembered yet next to impossible to crack) that have to be changed only once or twice a year.

Storing passwords

If you have to choose between weak passwords that your users can memorize and strong passwords that your users must write down, I recommend having readers write down passwords and store the information securely. Train users to store their written passwords in a secure place — not on keyboards or in easily cracked password-protected computer files (such as spreadsheets). Users should store a written password in either of these locations:

- A locked file cabinet or office safe
- Full (whole) disk encryption which can prevent an intruder from ever accessing the OS and passwords stored on the system. Just know it's not foolproof, as I outline in Chapter 10.
- A secure password management tool such as
    - LastPass (`http://lastpass.com`)
    - Password Safe, an open source software originally developed by Counterpane (`http://passwordsafe.sourceforge.net`)

No passwords on sticky notes! People joke about it, but it *still* happens a lot, and it's not good for business!

Creating password policies

As an ethical hacker, you should show users the importance of securing their passwords. Here are some tips on how to do that:

- **Demonstrate how to create secure passwords.** Refer to them as *passphrases* because people tend to take *passwords* literally and use only words, which can be less secure.
- **Show what can happen when weak passwords are used or passwords are shared.**
- **Diligently build user awareness of social engineering attacks.**

Enforce (or at least encourage the use of) a strong password-creation policy that includes the following criteria:

- **Use upper- and lowercase letters, special characters, and numbers.** Never use only numbers. Such passwords can be cracked quickly.
- **Misspell words or create acronyms from a quote or a sentence.** For example, *ASCII* is an acronym for *American Standard Code for Information Interchange* that can also be used as part of a password.

- **Use punctuation characters to separate words or acronyms.**
- **Change passwords every 6 to 12 months or immediately if they're suspected of being compromised.** Anything more frequent introduces an inconvenience that serves only to create more vulnerabilities.
- **Use different passwords for each system.** This is especially important for network infrastructure hosts, such as servers, firewalls, and routers. It's okay to use similar passwords — just make them slightly different for each type of system, such as *SummerInTheSouth-Win7* for Windows systems and Linux+*SummerInTheSouth* for Linux systems.
- **Use variable-length passwords.** This trick can throw off attackers because they won't know the required minimum or maximum length of passwords and must try all password length combinations.
- **Don't use common slang words or words that are in a dictionary.**
- **Don't rely completely on similar-looking characters, such as *3* instead of *E*, *5* instead of *S*, or *!* instead of *1*.** Password-cracking programs can check for this.
- **Don't reuse the same password within at least four to five password changes.**
- **Use password-protected screen savers.** Unlocked screens are a great way for systems to be compromised even if their hard drives are encrypted.
- **Don't share passwords.** To each his or her own!
- **Avoid storing user passwords in an unsecured central location, such as an unprotected spreadsheet on a hard drive.** This is an invitation for disaster. Use Password Safe or a similar program to store user passwords.

Taking other countermeasures

Here are some other password-hacking countermeasures that I recommend:

- **Enable security auditing to help monitor and track password attacks.**
- **Test your applications to make sure they aren't storing passwords indefinitely in memory or writing them to disk.** A good tool for this is WinHex (`www.winhex.com/winhex/index-m.html`). I've used this tool to search a computer's memory for *password, pass=, login,* and so on and have come up with some passwords that the developers thought were cleared from memory.

  TIP

  Some password-cracking Trojan-horse applications are transmitted through worms or simple e-mail attachments. Such malware can be lethal to your password-protection mechanisms if they're installed on your systems. The best defense is malware protection or whitelisting software, from Symantec, McAfee, or Bit9.

- **Keep your systems patched.** Passwords are reset or compromised during buffer overflows or other denial of service (DoS) conditions.
- **Know your user IDs.** If an account has never been used, delete or disable the account until it's needed. You can determine unused accounts by manual inspection or by using a tool such as DumpSec (`www.systemtools.com/somarsoft/?somarsoft.com`), a tool that can enumerate the Windows operating system and gather user IDs and other information.

As the security administrator in your organization, you can enable *account lockout* to prevent password-cracking attempts. Account lockout is the ability to lock user accounts for a certain time after a certain number of failed login attempts has occurred. Most operating systems (and some applications) have this capability. Don't set it too low (fewer than five failed logins), and don't set it too high to give a malicious user a greater chance of breaking in. Somewhere between 5 and 50 might work for you. I usually recommend a setting of around 10 or 15. Consider the following when configuring account lockout on your systems:

- To use account lockout to prevent any possibilities of a user DoS condition, require two different passwords, and don't set a lockout time for the first one if that feature is available in your operating system.
- If you permit autoreset of the account after a certain period — often referred to as *intruder lockout* — don't set a short time period. Thirty minutes often works well.

A failed login counter can increase password security and minimize the overall effects of account lockout if the account experiences an automated attack. A login counter can force a password change after a number of failed attempts. If the number of failed login attempts is high and occurred over a short period, the account has likely experienced an automated password attack.

Other password-protection countermeasures include

- **Stronger authentication methods.** Examples of these are challenge/response, smart cards, tokens, biometrics, or digital certificates.
- **Automated password reset.** This functionality lets users manage most of their password problems without getting others involved. Otherwise, this support issue becomes expensive, especially for larger organizations.
- **Password-protect the system BIOS.** This is especially important on servers and laptops that are susceptible to physical security threats and vulnerabilities.

Securing Operating Systems

You can implement various operating system security measures to ensure that passwords are protected.

Regularly perform these low-tech and high-tech password-cracking tests to make sure that your systems are as secure as possible — perhaps as part of a monthly, quarterly, or biannual audit.

Windows

The following countermeasures can help prevent password hacks on Windows systems:

- Some Windows passwords can be gleaned by simply reading the clear-text or crackable ciphertext from the Windows Registry. Secure your registries by doing the following:
  - Allow only administrator access.
  - Harden the operating system by using well-known hardening best practices, such as those from SANS (`www.sans.or`), NIST (`http://csrc.nist.gov`), the Center for Internet Security Benchmarks/Scoring Tools (`www.cisecurity.org`), and the ones outlined in *Network Security For Dummies* by Chey Cobb.
- Keep all SAM database backup copies secure.
- Disable the storage of LM hashes in Windows for passwords that are shorter than 15 characters.

  For example, you can create and set the NoLMHash registry key to a value of 1 under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa`.
- Use local or group security policies to help eliminate weak passwords on Windows systems before they're created.
- Disable null sessions in your Windows version.
- In Windows XP and later versions, enable the Do Not Allow Anonymous Enumeration of SAM Accounts and Shares option in the local security policy.

Chapter 11 covers Windows hacks you need to understand and test in more detail.

Linux and UNIX

The following countermeasures can help prevent password cracks on Linux and UNIX systems:

- Ensure that your system is using shadowed MD5 passwords.
- Help prevent the creation of weak passwords. You can use either the built-in operating system password filtering (such as cracklib in Linux) or a password-auditing program (such as npasswd or passwd+).
- Check your `/etc/passwd` file for duplicate root UID entries. Hackers can exploit such entries to gain backdoor access.

Chapter 12 explains the Linux hacks and how to test Linux systems for vulnerabilities.

Introduction

Welcome to *Hacking For Dummies,* 4th Edition. This book outlines — in plain English — computer hacker tricks and techniques that you can use to assess the security of your information systems, find the security vulnerabilities that matter, and fix the weaknesses before criminal hackers and malicious users take advantage of them. This hacking is the professional, aboveboard, and legal type of security testing — which I call *ethical hacking* throughout the book.

Computer and network security is a complex subject and an ever-moving target. You must stay on top of it to ensure that your information is protected from the bad guys. That's where the tools and techniques outlined in this book can help.

You can implement all the security technologies and other best practices possible, and your information systems might be secure — as far as you know. However, until you understand how malicious attackers think, apply that knowledge, and use the right tools to assess your systems from their point of view, you can't get a true sense of how secure your information really is.

Ethical hacking — which encompasses formal and methodical *penetration testing, white hat hacking,* and *vulnerability testing* — is necessary to find security flaws and to help validate that your information systems are truly secure on an ongoing basis. This book provides you with the knowledge to implement an ethical hacking program successfully, perform ethical hacking tests, and put the proper countermeasures in place to keep external hackers and malicious users in check.

Who Should Read This Book?

WARNING!

Disclaimer: If you choose to use the information in this book to hack or break into computer systems maliciously and without authorization, you're on your own. Neither I (the author) nor anyone else associated with this book shall be liable or responsible for any unethical or criminal choices that you might make and execute using the methodologies and tools that I describe. This book is intended solely for IT and information security professionals to test information security — either on your own systems or on a client's systems — in an authorized fashion.

Okay, now that that's out of the way, it's time for the good stuff! This book is for you if you're a network administrator, information security manager, security consultant, security auditor, compliance manager, or interested in finding out more about legally and ethically testing computer systems and IT operations to make things more secure.

As the ethical hacker performing well-intended information security assessments, you can detect and point out security holes that might otherwise be overlooked. If you're performing these tests on your systems, the information you uncover in your tests can help you win over management and prove that information security really is a business issue to be taken seriously. Likewise, if you're performing these tests for your clients, you can help find security holes that can be plugged before the bad guys have a chance to exploit them.

The information in this book helps you stay on top of the security game and enjoy the fame and glory of helping your organization and clients prevent bad things from happening to their information.

About This Book

Hacking For Dummies, 4th Edition, is a reference guide on hacking your systems to improve security and help minimize business risks. The ethical hacking techniques are based on written and unwritten rules of computer system penetration testing, vulnerability testing, and information security best practices. This book covers everything from establishing your hacking plan to testing your systems to plugging the holes and managing an ongoing ethical hacking program. Realistically, for many networks, operating systems, and applications, thousands of possible hacks exist. I cover the major ones on various platforms and systems. Whether you need to assess security vulnerabilities on a small home office network, a medium-sized corporate network, or across large enterprise systems, *Hacking For Dummies,* 4th Edition, provides the information you need.

How to Use This Book

This book includes the following features:

- Various technical and nontechnical hack attacks and their detailed methodologies
- Information security testing case studies from well-known information security experts
- Specific countermeasures to protect against hack attacks

Before you start hacking your systems, familiarize yourself with the information in Part I so you're prepared for the tasks at hand. The adage "if you fail to plan, you plan to fail" rings true for the ethical hacking process. You must get permission and have a solid game plan in place if you're going to be successful.

This material is not intended to be used for unethical or illegal hacking purposes to propel you from script kiddie to megahacker. Rather, it is designed to provide you with the knowledge you need to hack your own or your clients' systems — ethically and legally — to enhance the security of the information involved.

What You Don't Need to Read

Depending on your computer and network configurations, you may be able to skip chapters. For example, if you aren't running Linux or wireless networks, you can skip those chapters. Just be careful. You may think you're not running certain systems, but they could very well be on your network somewhere.

Foolish Assumptions

I make a few assumptions about you, the aspiring information security professional:

- You're familiar with basic computer-, network-, and information-security–related concepts and terms.
- You have a basic understanding of what hackers and malicious users do.
- You have access to a computer and a network on which to use these techniques.
- You have access to the Internet to obtain the various tools used in the ethical hacking process.
- You have permission to perform the hacking techniques described in this book.

How This Book Is Organized

This book is organized into seven modular parts, so you can jump around from one part to another as needed. Each chapter provides practical methodologies and practices you can use as part of your ethical hacking efforts, including checklists and references to specific tools you can use, as well as resources on the Internet.

Part I: Building the Foundation for Ethical Hacking

This part covers the fundamental aspects of ethical hacking. It starts with an overview of the value of ethical hacking and what you should and shouldn't do during the process. You get inside the malicious mindset and discover how to plan your ethical hacking efforts. This part covers the steps involved in the ethical hacking process, including how to choose the proper tools.

Part II: Putting Ethical Hacking in Motion

This part gets you rolling with the ethical hacking process. It covers several well-known and widely used hack attacks, including social engineering and cracking passwords, to get your feet wet. This part covers the human and physical elements of security, which tend to be the weakest links in any information security program. After you plunge into these topics, you'll know the tips and tricks required to perform common general hack attacks against your systems, as well as specific countermeasures to keep your information systems secure.

Part III: Hacking Network Hosts

Starting with the larger network in mind, this part covers methods to test your systems for various well-known network infrastructure vulnerabilities. From weaknesses in the TCP/IP protocol suite to wireless network insecurities, you find out how networks are compromised by using specific methods of flawed network communications, along with various countermeasures that you can implement to avoid becoming a victim. I then delve down into mobile devices and show how phones, tablets, and the like can be exploited. This part also includes case studies on some of the network hack attacks that are presented.

Part IV: Hacking Operating Systems

Practically all operating systems have well-known vulnerabilities that hackers often exploit. This part jumps into hacking the widely used operating systems: Windows and Linux. The hacking methods include scanning your operating systems for vulnerabilities and enumerating the specific hosts to gain detailed information. This part also includes information on exploiting well-known vulnerabilities in these operating systems, taking over operating systems remotely, and specific countermeasures that you can implement to make your operating systems more secure. This part includes case studies on operating system hack attacks.

Part V: Hacking Applications

Application security is gaining more visibility in the information security arena these days. An increasing number of attacks — which are often able to bypass firewalls, intrusion detection systems, and antivirus software — are aimed directly at various applications. This part discusses hacking specific business applications, including coverage of e-mail systems, Voice over Internet Protocol (VoIP), web applications, databases, and storage systems, along with practical countermeasures that you can put in place to make your systems more secure.

Part VI: Ethical Hacking Aftermath

After you perform your ethical hack attacks, what do you do with the information you gather? Shelve it? Show it off? How do you move forward? This part answers these questions and more. From developing reports for upper management to remediating the security flaws that you discover to establishing procedures for your ongoing ethical hacking efforts, this part brings the ethical hacking process full circle. This information not only ensures that your effort and time are well spent, but also is evidence that information security is an essential element for success in any business that depends on computers and information technology.

Part VII: The Part of Tens

This part contains tips to help ensure the success of your ethical hacking program. You find out how to get upper management to buy into your ethical hacking program so you can get going and start protecting your systems. This part also includes the top ten ethical hacking mistakes you absolutely must avoid.

This part also includes an Appendix that provides a one-stop reference listing of ethical hacking tools and resources. You can find all the links in the Appendix on the *Hacking For Dummies* online Cheat Sheet at `www.dummies.com/cheatsheet/hacking`.

Icons Used in This Book

REMEMBER

This icon points out information that is worth committing to memory.

WARNING!

This icon points out information that could have a negative impact on your ethical hacking efforts — so please read it!

TIP

This icon refers to advice that can help highlight or clarify an important point.

TECHNICAL STUFF

This icon points out technical information that is interesting but not vital to your understanding of the topic being discussed.

Where to Go from Here

The more you know about how external hackers and rogue insiders work and how your systems should be tested, the better you're able to secure your computer systems. This book provides the foundation that you need to develop and maintain a successful ethical hacking program in order to minimize business risks.

Keep in mind that the high-level concepts of ethical hacking won't change as often as the specific information security vulnerabilities you protect against. Ethical hacking will always remain both an art and a science in a field that's ever-changing. You must keep up with the latest hardware and software technologies, along with the various vulnerabilities that come about month after month and year after year. When I do have important updates to this book, you can find them at `www.dummies.com/go/hackingfdupdates`.

You won't find a single *best* way to hack your systems, so tweak this information to your heart's content. Happy (ethical) hacking!

Contents at a Glance

Table of Contents