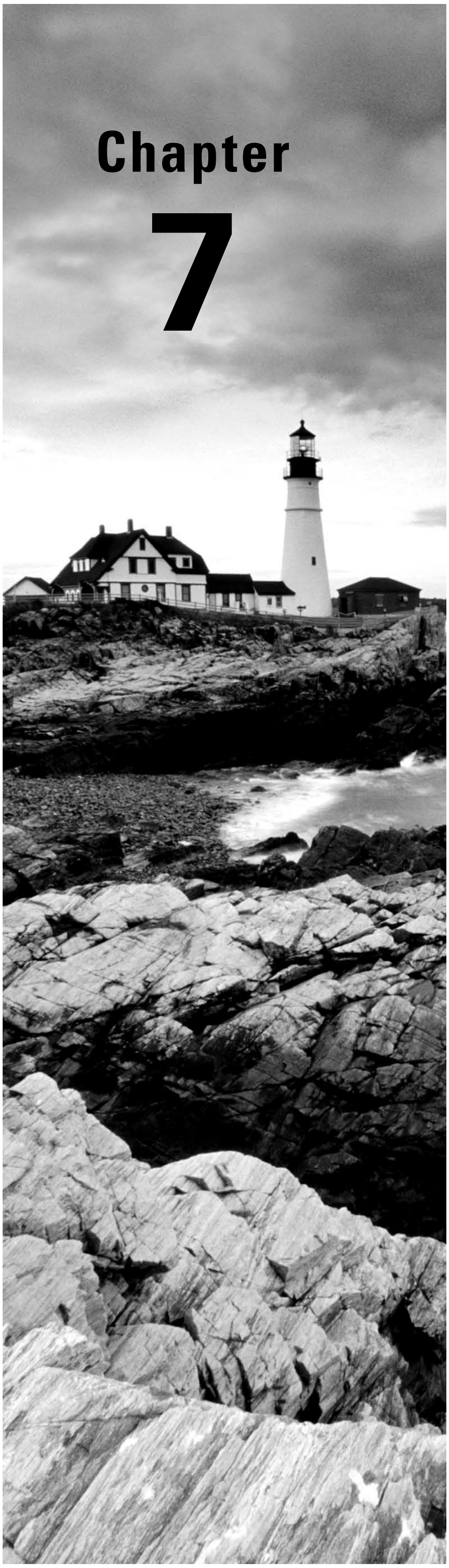

Chapter

7

# Gaining Access to a System

## CEH EXAM OBJECTIVES COVERED IN THIS CHAPTER:

✓ **III. Security**

- O. Vulnerabilities

✓ **IV. Tools/Systems/Programs**

- O. Operating Environments
- Q. Log Analysis Tools
- S. Exploitation Tools

Using the information gathered so far, you can now transition into the next phase: gaining access to a system. All the information you've gathered up to this point has been focused toward this goal. In this chapter, you will see how you can use information from previous interactions to "kick down the door" of a system and carry out your goal.

After enumeration, scanning, and footprinting, you can now start your attack on the system. If you look at the information you obtained in past phases, such as usernames, groups, passwords, permissions, and other system details, you can see that you are attempting to paint a picture of the victim that is as complete as is possible. The more information you gather, the better, and the easier it is for you to locate the points that lend themselves to attack or are most vulnerable.

NOTE Always remember as a pen tester to keep good notes about your activities and the information you gather. This is important for numerous reasons: You will want to present the information to your client, keep it among your legal records, and, in this chapter, use it to help you put together the best possible attack and assessment.

# Up to This Point

Let's take a brief look back at the previous phases to see what types of information you have and how it carries forward to this point.

## Footprinting

Footprinting is the first step in this process and simply involves gathering as much information as you possibly can about a target. You are looking for information pertaining to the whole organization, including technology, people, policies, facilities, network information, and anything else that may seem useful. Footprinting helps you understand the organization, create a profile that you can use for later stages of your attack, and plan a defensive strategy.

Information you gather during this phase may include the following:

- IP address ranges
- Namespaces
- Employee information
- Phone numbers

- Facility information
- Job information

Footprinting shows you the amount of information that is left lying on the table by most organizations. During your exploration, you learned that you can acquire a significant amount of data from myriad sources, both common and uncommon.

## Scanning

When you moved on from footprinting, you transitioned into the scanning phase. Scanning is focused on gathering information from a network with the intention of locating active hosts. You identify hosts for the purpose of attack and in order to make security assessments as needed. You can find information about target systems over the Internet by using public IP addresses. In addition to addresses, you also try to gather information about services running on each host.

During this phase, you use techniques such as these:

- Pings
- Ping sweeps
- Port scans
- Tracert

Some of the processes you use unmask or uncover varying levels of detail about services. You can also use inverse-scanning techniques that allow you to determine which IP addresses from the ranges you uncovered during footprinting do not have a corresponding live host behind them.

## Enumeration

The last phase before you attempt to gain access to a system is enumeration. Enumeration, as you have observed, is the systematic probing of a target with the goal of obtaining user lists, routing tables, and protocols from the system. This phase represents a significant shift in the process: it is your first step from being on the outside looking in, to being on the inside of the system and gathering data. Information about shares, users, groups, applications, protocols, and banners can prove useful in getting to know your target. This information is now carried forward into the attack phase.

The attacker seeks to locate items such as user and group data that let them remain under the radar longer. Enumeration involves making many more active connections with the system than during previous phases; once you reach this phase, the possibility of detection is much higher, because many systems are configured to log any and all attempts to gain information. Some of the data you locate may already have been made public by the target, but you may also uncover hidden share information, among other items.

The information gathered during this phase typically includes, but is not limited to, the following:

- Usernames
- Group information

- Passwords
- Hidden shares
- Device information
- Network layout
- Protocol information
- Server data
- Service information

## System Hacking

Once you have completed the first three phases, you can move into the system-hacking phase. At this point, the process becomes much more complex: You can't complete the system-hacking phase in a single pass. It involves using a methodical approach that includes cracking passwords, escalating privileges, executing applications, hiding files, covering tracks, concealing evidence, and then pushing into a more involved attack.

Let's look at the first step in system hacking: password cracking.

### Password Cracking

In the enumeration phase, you collected a wealth of information, including usernames. These usernames are important now because they give you something on which to focus your attack more closely. You use password cracking to obtain the credentials of a given account with the intention of using the account to gain authorized access to the system under the guise of an authentic user.

NOTE

In a nutshell, password cracking is the process of recovering passwords from transmitted or stored data. In this way, an attacker may seek to recover and use a misplaced or forgotten password. System administrators may use password cracking to audit and test a system for holes in order to strengthen the system, and attackers may use password cracking to gain authorized access.

Typically, the hacking process starts with assaults against passwords. Passwords may be cracked or audited using manual or automated techniques designed to reveal credentials.

To fully grasp why password cracking is so often used first during an attack and is commonly successful, let's look at the nature of passwords. A password is designed to be something an individual can remember easily but at the same time not something that can be easily guessed or broken. This is where the problem lies: Human beings tend to choose passwords that are easy to remember, which can make them easy to guess. Although choosing passwords that are easier to remember is not a bad thing, it can be a liability if individuals choose passwords that are too simple to recall or guess.

Here are some examples of passwords that lend themselves to cracking:

- Passwords that use only numbers
- Passwords that use only letters
- Passwords that are all upper- or lowercase
- Passwords that use proper names
- Passwords that use dictionary words
- Short passwords (fewer than eight characters)

Generally speaking, the rules for creating a strong password are a good line of defense against the attacks we will explore. Many companies already employ these rules in the form of password requirements or complexity requirements, but let's examine them in the interest of being complete.

Typically, when a company is writing policy or performing training they will have a document, guidance, or statement that says to avoid the following:

- Passwords that contain letters, special characters, and numbers: stud@52
- Passwords that contain only numbers: 23698217
- Passwords that contain only special characters: &*#@!(%)
- Passwords that contain letters and numbers: meetl23
- Passwords that contain only letters: POTHMYDE
- Passwords that contain only letters and special characters: rex@&ba
- Passwords that contain only special characters and numbers: 123@$4

Users that select passwords that contain patterns that adhere to any of the points on this list are less vulnerable to most of the attacks we will discuss for recovering passwords.

NOTE Remember that just because a password adheres to the conventions discussed here does not mean it is bulletproof with regard to attacks. Adherence to these guidelines makes it less vulnerable, but not impervious. One of the points you will learn both as an attacker and a defender is that there is no 100-percent solution to security, only ways to reduce your vulnerability.

## Password Cracking Techniques

Popular culture would have us believe that cracking a password is as simple as running some software and tapping a few buttons. The reality is that special techniques are used to recover passwords. For the most part, you can break these techniques into five categories, which you will explore in depth later in this chapter; but let's take a high-level look at them now:

**Dictionary Attacks** An attack of this type takes the form of a password-cracking application that has a dictionary file loaded into it. The dictionary file is a text file that contains a list of known words up to and including the entire dictionary. The application uses this list to test different words in an attempt to recover the password. Systems that use passphrases typically are not vulnerable to this type of attack.

**Brute-force Attacks** In this type of attack, every possible combination of characters is attempted until the correct one is uncovered. According to RSA Labs, "Exhaustive key-search, or brute-force search, is the basic technique for trying every possible key in turn until the correct key is identified."

**Hybrid Attack** This form of password attack builds on the dictionary attack, but with additional steps as part of the process. In most cases, this means passwords that are tried during a dictionary attack are modified with the addition and substitution of special characters and numbers, such as *P@ssw0rd* instead of *Password*.

**Syllable Attack** This type of attack is a combination of a brute-force and a dictionary attack. It is useful when the password a user has chosen is not a standard word or phrase.

**Rule-based Attack** This could be considered an advanced attack. It assumes that the user has created a password using information the attacker has some knowledge of ahead of time, such as phrases and digits the user may have a tendency to use.

In addition to these techniques, there are four types of attacks. Each offers a different, effective way of obtaining a password from a target:

**Passive Online Attacks** Attacks in this category are carried out simply by sitting back and listening—in this case, via technology, in the form of sniffing tools such as Wireshark, man-in-the-middle attacks, or replay attacks.

**Active Online Attacks** The attacks in this category are more aggressive than passive attacks because the process requires deeper engagement with the targets. Attackers using this approach are targeting a victim with the intention of breaking a password. In cases of weak or poor passwords, active attacks are very effective. Forms of this attack include password guessing, Trojan/spyware/key loggers, hash injection, and phishing.

**Offline Attacks** This type of attack is designed to prey on the weaknesses not of passwords, but of the way they are stored. Because passwords must be stored in some format, an attacker seeks to obtain them where they are stored by exploiting poor security or weaknesses inherent in a system. If these credentials happen to be stored in a plaintext or unencrypted format, the attacker will go after this file and gain the credentials. Forms of this attack include precomputed hashes, distributed network attacks, and rainbow attacks.

**Nontechnical Attacks** Also known as non-electronic attacks, these move the process offline into the real world. A characteristic of this attack is that it does not require any technical knowledge and instead relies on theft, deception, and other means. Forms of this attack include shoulder surfing, social engineering, and dumpster diving.

Let's look at each of these forms and its accompanying attacks so you can better understand them.

## Passive Online Attacks

A passive online attack, as you've learned, is one in which the attacker tends to be not engaged or less engaged than they would be during other kinds of attacks. The effectiveness of this attack tends to rely not only on how weak the password system is, but also on how reliably the password-collection mechanism is executed.

### Packet Sniffing

You learned about the technique of sniffing traffic and now it's time to apply this approach to an attack. Typically, a sniffer is not the preferred tool to use in an attack, due to the way it works and how it processes information. If you use a sniffer without any extra steps, you are limited to a single common collision domain. In other words, you can only sniff hosts that are not connected by a switch or bridge in the selected network segment.

NOTE It is possible to sniff outside of a given common collision domain, even if a switch is in the way, if you use an approach that is designed to attack and overcome the switch or bridge. However, such methods are aggressive and active and therefore generate a lot of traffic that makes detection that much easier for the defender.

Generally, a sniffing attack is most effective if it is performed on a network that employs a hub between the attacker and victim, or if the two parties are on the same segment of the collision domain. Many of the tools you will encounter or use will be most effective in the context of a network that employs a hub.

NOTE When you sniff for passwords, typically you are on the lookout for passwords from Telnet, FTP, SMTP, rlogin, and other vulnerable protocols. Once you've gathered the credentials, you can use them to gain access to systems or services.

### Man-in-the-middle

During this type of attack, two parties are communicating with one another and a third party inserts itself into the conversation and attempts to alter or eavesdrop on the communications. In order to be fully successful, the attacker must be able to sniff traffic from both parties at the same time.

Man-in-the-middle attacks commonly target vulnerable protocols and wireless technologies. Protocols such as Telnet and FTP are particularly vulnerable to this type of attack. However, such attacks are tricky to carry out and can result in invalidated traffic.

### Replay Attack

In a replay attack, packets are captured using a packet sniffer. After the relevant information is captured and extracted, the packets can be placed back on the network. The intention is to inject the captured information—such as a password—back onto the network and direct it toward a resource such as a server, with the goal of gaining access. Once replayed, the valid credentials provide access to a system, potentially giving an attacker the ability to change information or obtain confidential data.

## Active Online Attacks

The next attack type is the active online attack. These attacks use a more aggressive form of penetration that is designed to recover passwords.

### Password Guessing

Password guessing is a very crude but effective type of attack. An attacker seeks to recover a password by using words from the dictionary or by brute force. This process is usually carried out using a software application designed to attempt hundreds or thousands of words each second. The application tries all variations, including case changes, substitutions, digit replacement, and reverse case.

To refine this approach, an attacker may look for information about a victim, with the intention of discovering favorite pastimes or family names.

NOTE Password complexity goes a long way toward thwarting many of these types of attacks, because it makes the process of discovering a password slower and much more difficult.

### Trojans, Spyware, and Keyloggers

Malware is discussed in depth elsewhere in this book, but here we should mention its potential role during an attack. Malware such as Trojans, spyware, and keyloggers can prove very useful during an attack by allowing the attacker to gather information of all types, including passwords.

One form is keyboard sniffing or keylogging, which intercepts a password as the user enters it. This attack can be carried out when users are the victims of keylogging software or if they regularly log on to systems remotely without using protection.

### Hash Injection

This type of attack relies on the knowledge of hashing that you acquired during our investigation on cryptography and a few tricks. The attack relies on you completing the following four steps:

1. Compromise a vulnerable workstation or desktop.
2. When connected, attempt to extract the hashes from the system for high-value users, such as domain or enterprise admins.
3. Use the extracted hash to log on to a server such as a domain controller.
4. If the system serves as a domain controller or similar, attempt to extract hashes from the system with the intention of exploiting other accounts.

Real World Scenario

### Password Hashing

Passwords are not stored in cleartext on a system in most cases due to their extremely sensitive nature. Because storing passwords in the clear can be considered risky, you can use security measures such as password hashes.

As you learned in the Chapter 3, "Cryptography," hashing is a form of one-way encryption that is used to verify integrity. Passwords are commonly stored in a hashed format so the password is not in cleartext. When a password provided by the user needs to be verified, it is hashed on the client side and then transmitted to the server, where the stored hash and the transmitted hash are compared. If they match, the user is authenticated; if not, the user is not authenticated.

## Offline Attacks

Offline attacks represent yet another form of attack that is very effective and difficult to detect in many cases. Such attacks rely on the attacking party being able to learn how passwords are stored and then using this information to carry out an attack.

EXERCISE 7.1

### Extracting Hashes from a System

Now that you have seen how hashes can be extracted, let's use pwdump to perform this process:

1. Open the command prompt.
2. Type **`pwdump7.exe`** to display the hashes on a system.
3. Type **`pwdump7 > C:\hash.txt.`**
4. Press Enter.
5. Using Notepad, browse to the C drive and open the `hash.txt` file to view the hashes.

### Precomputed Hashes or Rainbow Tables

Precomputed hashes are used in an attack type known as a rainbow table. Rainbow tables compute every possible combination of characters prior to capturing a password. Once all the passwords have been generated, the attacker can capture the password hash from the network and compare it with the hashes that have already been generated.

With all the hashes generated ahead of time, it becomes a simple matter to compare the captured hash to the ones generated, typically revealing the password in a few moments.

Of course, there's no getting something for nothing, and rainbow tables are no exception. The downside of rainbow tables is that they take time. It takes a substantial period of time, sometimes days, to compute all the hash combinations ahead of time. Another downside is that you can't crack passwords of unlimited length, because generating passwords of increasing length takes more time.

## Generating Rainbow Tables

You can generate rainbow tables many ways. One of the utilities you can use to perform this task is winrtgen, a GUI-based generator. Supported hashing formats in this utility include all of the following:

- Cisco PIX
- FastLM
- HalfLMChall
- LM
- LMCHALL
- MD2
- MD4
- MD5
- MSCACHE
- MySQL323
- MySQLSHAl
- NTLM
- NTLMCHALL
- ORACLE
- RIPEMD-160
- SHA1
- SHA-2 (256), SHA-2 (384), SHA-2 (512)

### EXERCISE 7.2

### Creating Rainbow Tables

Let's create a rainbow table to see what the process entails. Keep in mind that this process can take a while once started.

To perform this exercise, you will need to download the winrtgen application. To use winrtgen, follow these steps:

1. Start the `winrtgen.exe` tool.
2. Once winrtgen starts, click the Add Table button.
3. In the Rainbow Table Properties window, do the following:
   a. Select NTLM from the Hash drop-down list.
   b. Set Minimum Length to **4** and Maximum Length to **9**, with a Chain Count of **4000000**.
   c. Select Loweralpha from the Charset drop-down list.
4. Click OK to create the rainbow table.

Note that the creation of the rainbow table file will take a significant amount of time, depending on the speed of your computer and the settings you choose.

Exercise 7.1 and Exercise 7.2 perform two vital steps of the process: Exercise 7.1 extracts hashes of passwords from a targeted system, and Exercise 7.2 creates a rainbow table of potential matches (hopefully there is a match, if you used the right settings). Now that you have performed these two steps, you must recover the password (Exercise 7.3).

## EXERCISE 7.3

### Working with Rainbow Crack

Once you have created the rainbow table, you can use it to recover a password using the information from pwdump and winrtgen.

1. Double-click `rcrack_gui.exe`.
2. Click File, and then click Add Hash. The Add Hash window opens.
3. If you performed the pwdump hands on, you can now open the text file it created and copy and paste the hashes.
4. Click OK.
5. Click Rainbow Table from the menu bar, and click Search Rainbow Table. If you performed the winrtgen hands on, you can use that rainbow table here.
6. Click Open.

Rainbow tables are an effective method of revealing passwords, but the effectiveness of the method can be diminished through salting. Salting is used in Linux, Unix, and BSD, but it is not used in some of the older Windows authentication mechanisms such as LM and NTLM.

Salting a hash is a means of adding entropy or randomness in order to make sequences or patterns more difficult to detect. Rainbow tables perform a form of cryptanalysis. Salting tries to thwart this analysis by adding randomness (sometimes known as inducing entropy). Although you still may be able to break the system, it will be tougher to do.

## Distributed Network Attacks

One of the modern approaches to cracking passwords is a Distributed Network Attack (DNA). It takes advantage of unused processing power from multiple computers in an attempt to perform an action: in this case, cracking a password.

To make this attack work, you install a manager on a chosen system, which is used to manage multiple clients. The manager is responsible for dividing up and assigning work to the various systems involved in processing the data. On the client side, the software receives the assigned work unit, processes it, and returns the results to the manager.

The benefit of this type of attack is the raw computing power available. This attack combines small amounts of computing power from individual systems into a vast amount of computing power. Each computer's processing power is akin to a single drop of water: individually they are small, but together they become much more. Drops form larger bodies of water, and small pieces of processing power come together to form a huge pool of processing power.

### Real World Scenario

### Seeking Out New Life

One of the first well-known implementations of distributed computing is the SETI@home project. The Search for Extraterrestrial Intelligence (SETI) is a project that analyzes signals received from space to look for signs of life off Earth. The following is a description of the project from the `SETI@home` site.

Most of the SETI programs in existence today, including those at UC Berkeley, build large computers that analyze data in real time. None of these computers look very deeply at the data for weak signals, nor do they look for a large class of signal types, because they are limited by the amount of computer power available for data analysis. To tease out the weakest signals, a great amount of computer power is necessary. It would take a monstrous supercomputer to get the job done. SETI could never afford to build or buy that computing power. Rather than a huge computer to do the job, they could use a smaller computer and take longer to do it. But then there would be lots of data piling up. What if they used *lots* of small computers, all working simultaneously on different parts of the analysis? Where can the SETI team possibly find the thousands of computers they need to analyze the data continuously streaming in?

The UC Berkeley SETI team has discovered thousands of computers that may be available for use. Most of them sit around most of the time with toasters flying across their screens, accomplishing absolutely nothing and wasting electricity to boot. This is where SETI@home (and you!) come into the picture. The SETI@home project hopes to convince you to let them borrow your computer when you aren't using it, to help them "... search out new life and new civilizations." You do this by installing a screen saver that gets a chunk of data from SETI over the Internet, analyzes that data, and then reports the results. When you need your computer, the screen saver instantly gets out of the way and only continues its analysis when you are finished with your work.

## Other Options for Obtaining Passwords

There are still other ways to obtain passwords.

### Default Passwords

One of the biggest potential vulnerabilities is also one of the easiest to resolve: default passwords. Default passwords are set by the manufacturer when the device or system is built. They are documented and provided to the final consumer of the product and are intended to be changed. However, not all users or businesses get around to taking this step, and hence they leave themselves vulnerable. The reality is that with a bit of scanning and investigation, an attacking party can make some educated guesses about what equipment or systems you may be running. If they can determine that you have not changed the defaults, they can look up your default password at any of the following sites:

- `http://cirt.net`
- `http://default-password.info`
- `www.defaultpassword.us`
- `www.passwordsdatabase.com`
- `https://w3dt.net`
- `www.virus.org`
- `http://open-sez.me`
- `http://securityoverride.org`
- `www.routerpasswords.com`
- `www.fortypoundhead.com`

## Guessing

Although it is decidedly old school, guessing passwords manually can potentially yield results, especially in environments where good password practices are not followed. Simply put, an attacker may target a system by doing the following:

1. Locate a valid user.
2. Determine a list of potential passwords.
3. Rank possible passwords from least to most likely.
4. Try passwords until access is gained or the options are exhausted.

This process can be automated through the use of scripts created by the attacker, but it still qualifies as a manual attack.

### USB Password Theft

In contrast to manual methods, there are some automated mechanisms for obtaining passwords, such as via USB drives. This method entails embedding a password-stealing application on a USB drive and then physically plugging the drive into a target system. Because many users store their passwords for applications and online sites on their local machine, the passwords may be easily extracted (see Exercise 7.4).

**EXERCISE 7.4**

### PSPV

In order to carry out this attack you can use the following generic steps:

1. Obtain a password-hacking utility such as `pspv.exe`.
2. Copy the utility to a USB drive.
3. Create a Notepad file called **`launch.bat`** containing the following lines:

```
[autorun]
en = launch.bat
Start pspv.exe /s passwords.txt
```

4. Save `launch.bat` to the USB drive.

At this point, you can insert the USB drive into a target computer. When you do, `pspv.exe` will run, extract passwords, and place them in the `passwords.txt` file, which you can open in Notepad.

It is worth noting that this attack can be thwarted quite easily by disabling autoplay of USB devices, which is on by default in Windows.

NOTE The `pspv.exe` tool is a protected-storage password viewer that displays stored passwords on a Windows system if they are contained in Internet Explorer and other applications.

### Using Password Cracking

Using any of the methods discussed here with any type of password-cracking software may sound easy, but there is one item to consider: which password to crack? Going back to the enumeration phase, we discussed that usernames can be extracted from the system using a number of software packages or methods. Using these software tools, the attacker can uncover usernames and then target a specific account with their password-cracking tool of choice.

So, which password to crack? Accounts such as the administrator account are targets of opportunity, but so are lower-level accounts such as guest that may not be as heavily defended nor even considered during security planning.

## Authentication on Microsoft Platforms

Now that you know the different mechanisms through which you can obtain credentials, as well as how you can target them, let's look at some authentication mechanisms. We will focus on mechanisms on the Microsoft platform: SAM, NTLM, LM, and Kerberos.

### Security Accounts Manager (SAM)

Inside the Windows operating system is a database that stores security principals (accounts or any entity that can be authenticated). In the Microsoft world, these principals can be stored locally in a database known as the Security Accounts Manager (SAM). Credentials, passwords, and other account information are stored in this database; the passwords are stored in a hashed format. When the system is running, Windows keeps a file lock on the SAM to prevent it from being accessed by other applications or processes. When the system is running, however, a copy of the SAM database also resides in memory and can be accessed, given the right tools.

NOTE The system will only give up exclusive access of the SAM when powered off or when the system has a Blue Screen of Death failure.

In order to improve security, Microsoft added some features designed to preserve the integrity of the information stored in the database. For example, a feature known as the SYSKEY was added starting in Windows NT 4.0 to improve the existing security of the SAM. The SYSKEY is nothing more than a fancy name for an encryption key that is used to partially encrypt the SAM and protect the information stored within. By default, this feature is enabled on all systems later than NT 4.0; although it can be disabled, it is strongly recommended that you do not do so. With the SYSKEY in place, credentials are safe against many offline attacks.

### How Passwords Are Stored within the SAM

In Windows XP and later platforms, passwords are stored in a hashed format using the LM/NTLM hashing mechanisms. The hashes are stored in `c:\windows\system32\SAM`.

An account in the SAM looks like this:

```
Link:1010:624AAC413795CDC14E835F1CD90F4C76:6F585FF8FF6280B59CCE252FDB50
0EB8:::
```

The bold part before the colon is the LM hash, and the bold part after the colon represents the NTLM hash—both for a given password on a standard user account. Password crackers such as Ophcrack and L0phtcrack display and attempt to decipher these hashes, as do applications such as pwdump.

> NOTE
>
> Versions of Windows after XP no longer store the LM hash by default. They store a blank or a dummy value that has no direct correlation to any user's actual password, so extracting this value and using a brute-force attack to decipher it is pointless. This dummy value is also used when the password exceeds 14 characters, which is longer than the LM hash mechanism can support.

In Windows, as in other systems, password hashing may be strengthened by using a process known as salting. This technique is designed to add an additional layer of randomness to a hash during the generation process. With salt added to a hash offline and precomputed, attacks become much more difficult to execute successfully.

## NTLM Authentication

NT LAN Manager (NTLM) is a protocol exclusive (proprietary) to Microsoft products. NTLM versions 1 and 2 are still very widely used in environments and applications where other protocols such as Kerberos are not available, but Microsoft recommends that its use be avoided or phased out.

NTLM comes in two versions: NTLMv1 and NTLMv2. NTLMv1 has been in use for many years and still has some support in newer products, but it has largely been replaced in applications and environments with at least NTLMv2 if not other mechanisms. NTLMv2 is an improved version of the NTLM protocol. It boasts better security than version 1, but it is still seen as relatively insecure and as such should be avoided as well.

> NOTE
>
> You may hear of another mechanism layered on top of NTLM known as Security Support Provider (SSP). This protocol is combined with NTLM to provide an additional layer of protection on top of the existing authentication process.

Overall, the process of authentication with the NTLM protocol uses the following steps:

1. The client enters their username and password into the login prompt or dialog.
2. Windows runs the password through a hashing algorithm to generate a hash for the specific password.
3. The client transmits the username and hash to a domain controller.
4. The domain controller generates a 16-byte random character string known as a *nonce* and transmits it back to the client.
5. The client encrypts the nonce with the hash of the user password and sends it back to the domain controller.
6. The domain controller retrieves the hash from its SAM and uses it to encrypt the nonce it sent to the client.

At this point, if the hashes match, the login request is accepted. If not, the request is denied.

## Kerberos

On the Microsoft platform, version 5 of the Kerberos authentication protocol has been in use since Windows 2000. The protocol offers a robust authentication framework through the use of strong cryptographic mechanisms such as secret key cryptography. It provides mutual authentication of client and server.

The Kerberos protocol makes use of the following groups of components:

- Key distribution center (KDC)
- Authentication server (AS)
- Ticket-granting server (TGS)

The process of using Kerberos works much like the following:

1. You want to access another system, such as a server or client. Because Kerberos is in use in this environment, a “ticket” is required.
2. To obtain this ticket, you are first authenticated against the AS, which creates a session key based on your password together with a value that represents the service you wish to connect to. This request serves as your ticket-granting ticket (TGT).
3. Your TGT is presented to a TGS, which generates a ticket that allows you to access the service.
4. Based on the situation, the service either accepts or rejects the ticket. In this case, assume that you are authorized and gain access.

The TGT is valid for only a finite period of time before it has to be regenerated. This acts as a safeguard against it being compromised.

## Privilege Escalation

When you obtain a password and gain access to an account, there is still more work to do: privilege escalation. The reality is that the account you’re compromising may end up being a lower-privileged and less-defended one. If this is the case, you must perform privilege escalation prior to carrying out the next phase. The goal should be to gain a level where fewer restrictions exist on the account and you have greater access to the system.

Every operating system ships with a number of user accounts and groups already present. In Windows, preconfigured users include the administrator and guest accounts. Because it is easy for an attacker to find information about the accounts that are included with an operating system, you should take care to ensure that such accounts are secured properly, even if they will never be used. An attacker who knows that these accounts exist on a system is more than likely to try to obtain their passwords.

There are two defined types of privilege escalation, each of which approaches the problem of obtaining greater privileges from a different angle:

**Horizontal Privilege Escalation** An attacker attempts to take over the rights and privileges of another user who has the same privileges as the current account.

**Vertical Privilege Escalation** The attacker gains access to an account and then tries to elevate the privileges of the account. It is also possible to carry out a vertical escalation by compromising an account and then trying to gain access to a higher-privileged account.

One way to escalate privileges is to identify an account that has the desired access and then change the password. Several tools that offer this ability, including the following:

- Active@ Password Changer
- Trinity Rescue Kit
- ERD Commander
- Windows Recovery Environment (WinRE)
- Password Resetter

Let's look at one of these applications a little closer: Trinity Rescue Kit (TRK). According to the developers of TRK:

Trinity Rescue Kit (TRK) is a Linux distribution that is specifically designed to be run from a CD or flash drive. TRK was designed to recover and repair both Windows and Linux systems that were otherwise unbootable or unrecoverable. While TRK was designed for benevolent purposes, it can easily be used to escalate privileges by resetting passwords of accounts that you would not otherwise have access to. TRK can be used to change a password by booting the target system off of a CD or flash drive and entering the TRK environment. Once in the environment, a simple sequence of commands can be executed to reset the password of an account.

The following steps change the password of the administrator account on a Windows system using the TRK:

1. At the command line, enter the following command: **`winpass -u Administrator`**.
2. The `winpass` command displays a message similar to the following:

```
Searching and mounting all file system on local machine
Windows NT/2K/XP installation(s) found in:
1: /hda1/Windows
Make your choice or 'q' to quit [1]:
```

3. Type **1**, or the number of the location of the `Windows` folder if more than one install exists.
4. Press Enter.
5. Enter the new password, or accept TRK's suggestion to set the password to a blank.
6. You see this message: "Do you really wish to change it?" Enter **Y**, and press Enter.
7. Type **init 0** to shut down the TRK Linux system.
8. Reboot.

# Executing Applications

Once you gain access to a system and obtain sufficient privileges, it's time to compromise the system and carry out the attack. Which applications are executed at this point is up to the attacker, but they can either be custom-built applications or off-the-shelf software.

NOTE

In some circles, once an attacker has gained access to a system and is executing applications on it, they are said to *own* the system.

An attacker executes different applications on a system with specific goals in mind:

**Backdoors** Applications of this type are designed to compromise the system in such a way as to allow later access to take place. An attacker can use these backdoors later to attack the system. Backdoors can come in the form of rootkits, Trojans, and similar types. They can even include software in the form of remote access Trojans (RATs).

**Crackers** Any software that fits into this category is characterized by the ability to crack code or obtain passwords.

**Keyloggers** Keyloggers are hardware or software devices used to gain information entered via the keyboard.

**Malware** This is any type of software designed to capture information, alter, or compromise the system.

## Planting a Backdoor

There are many ways to plant a backdoor on a system, but let's look at one provided via the PsTools suite. This suite includes a mixed bag of utilities designed to ease system administration. Among these tools is PsExec, which is designed to run commands interactively or noninteractively on a remote system. Initially, the tool may seem similar to Telnet or remote desktop, but it does not require installation on the local or remote system in order to work. To work, PsExec need only be copied to a folder on the local system and run with the appropriate switches.

Let's take a look at some of the commands you can use with PsExec:

- The following command launches an interactive command prompt on a system named `\\zelda`: `psexec \\zelda cmd`.
- This command executes `ipconfig` on the remote system with the `/all` switch, and displays the resulting output locally: `psexec \\zelda ipconfig /all`.
- This command copies the program `rootkit.exe` to the remote system and executes it interactively: `psexec \\zelda -c rootkit.exe`.
- This command copies the program `rootkit.exe` to the remote system and executes it interactively using the administrator account on the remote system: `psexec \\zelda -u administrator -c rootkit.exe`.

As these commands illustrate, it is possible for an attacker to run an application on a remote system quite easily. The next step is for the attacker to decide what to do or what to run on the remote system. Some of the common choices are Trojans, rootkits, and backdoors.

Other utilities that may prove helpful in attaching to a system remotely are the following:

**PDQ Deploy** This utility is designed to assist with the deployment of software to a single system or to multiple systems across a network. The utility is designed to integrate with Active Directory as well as other software packages.

**RemoteExec** This utility is designed to work much like PsExec, but it also makes it easy to restart, reboot, and manipulate folders on the system.

**DameWare** This is a set of utilities used to remotely administer and control a system. Much like the other utilities on this list, it is readily available and may not be detected by antivirus utilities. DameWare also has the benefit of working across platforms such as Windows, OS X, and Linux.

## Covering Your Tracks

Once you have penetrated a system and installed software or run some scripts, the next step is cleaning up after yourself or covering your tracks. The purpose of this phase is to prevent your attack from being easily discovered by using various techniques to hide the red flags and other signs. During this phase, you seek to eliminate error messages, log files, and other items that may have been altered during the attack process.

### Disabling Auditing

One of the best ways to prevent yourself from being discovered is to leave no tracks at all. And one of the best ways to do that is to prevent any tracks from being created or at least minimize the amount of evidence. When you're trying not to leave tracks, a good starting point is altering the way events are logged on the targeted system.

Disabling auditing on a system prevents certain events from appearing and therefore slows detection efforts. Remember that auditing is designed to allow for the detection and tracking of selected events on a system. Once auditing is disabled, you have effectively deprived the defender of a great source of information and forced them to seek other methods of detection.

In the Windows environment, you can disable auditing with the `auditpol` command included. Using the `NULL` session technique you saw during your enumeration activities, you can attach to a system remotely and run the command as follows:

```
auditpol \\<ip address of target> /clear
```

You can also perform what amounts to the surgical removal of entries in the Windows Security Log, using tools such as the following:

- Dumpel
- Elsave
- WinZapper
- CCleaner
- Wipe
- MRU-Blaster
- Tracks Eraser Pro
- Clear My History

## Data Hiding

There are other ways to hide evidence of an attack, including hiding the files placed on the system such as EXE files, scripts, and other data. Operating systems such as Windows provide many methods you can use to hide files, including file attributes and alternate data streams.

File attributes are a feature of operating systems that allow files to be marked as having certain properties, including read-only and hidden. Files can be flagged as hidden, which is a convenient way to hide data and prevent detection through simple means such as directory listings or browsing in Windows Explorer. Hiding files this way does not provide complete protection, however, because more advanced detective techniques can uncover files hidden in this manner.

## Alternate Data Streams (ADS)

A very effective method of hiding data on a Windows system is also one of the lesser-known ones: Alternate Data Streams (ADS). This feature is part of the NTFS file system and has been since the 1990s, but since its introduction it has received little recognition; this makes it both useful for an attacker who is knowledgeable and dangerous for a defender who knows little about it.

Originally, this feature was designed to ensure interoperability with the Macintosh Hierarchical File System (HFS), but it has since been used for other purposes. ADS provides the ability to fork or hide file data within existing files without altering the appearance or behavior of a file in any way. In fact, when you use ADS, you can hide a file from all traditional detection techniques as well as `dir` and Windows Explorer.

In practice, the use of ADS is a major security issue because it is nearly a perfect mechanism for hiding data. Once a piece of data is embedded and hidden using ADS, it can lie in wait until the attacker decides to run it later.

The process of creating an ADS is simple:

```
type triforce.exe > smoke.doc:triforce.exe
```

Executing this command hides the file `triforce.exe` behind the file `smoke.doc`. At this point, the file is streamed. The next step is to delete the original file that you just hid, `triforce.exe`.

As an attacker, retrieving the file is as simple as this:

```
start smoke.doc:triforce.exe
```

This command has the effect of opening the hidden file and executing it.

As a defender, this sounds like bad news, because files hidden this way are impossible to detect using most means. But by using some advanced methods, they can be detected. Some of the tools that can be used to do this include the following:

- SFind—A forensic tool for finding streamed files
- LNS—Used for finding ADS streamed files
- Tripwire—Used to detect changes in files; by nature can detect ADS

NOTE

ADS is available only on NTFS volumes, although the version of NTFS does not matter. This feature does not work on other file systems.

# Summary

This chapter covered the process of gaining access to a system. We started by looking at how to use the information gathered during the enumeration process as inputs into the system-hacking process. You gathered information in previous phases with little or no interaction or disturbance of the target, but in this phase you are finally actively penetrating the target and making an aggressive move. Information brought into this phase includes usernames, IP ranges, share names, and system information.

An attacker who wants to perform increasingly aggressive and powerful actions needs to gain greater access. This is done by attempting to obtain passwords through brute force, social engineering, guessing, or other means. Once an attacker has obtained or extracted a password for a valid user account from a system, they can then attempt to escalate their privileges either horizontally or vertically in order to perform tasks with fewer restrictions and greater power.

When an account with greater power has been compromised, the next step is to try to further breach the system. An attacker at this point can try more damaging and serious actions by running scripts or installing software on the system that can perform any sort of action. Common actions that an attacker may attempt to carry out include installing keyloggers, deploying malware, installing remote access Trojans, and creating backdoors for later access.

Finally, an attacker will attempt to cover their tracks in order to avoid having the attack detected and stopped. An attacker may attempt to stop auditing, clear event logs, or surgically remove evidence from log files. In extreme cases, an attacker may even choose to use features such as Alternate Data Streams to conceal evidence.

# Exam Essentials

**Understand the process of gaining access to a system.** Make sure you can identify the process of system hacking, how it is carried out against a system, and what the end results are for the attacker and the defender.

**Know the different types of password cracking.** Understand the differences between the types of password cracking and hacking techniques. Understand the difference between online and offline attacks as well as nontechnical attacks. Know how accounts are targeted based on information obtained from the enumeration phase.

**Understand the difference between horizontal and vertical privilege escalation.** Two methods are available for escalating privileges: horizontal and vertical escalation. Horizontal escalation involves compromising an account with similar privileges, and vertical escalation attempts to take over an account with higher privileges.

**Identify the methods of covering your tracks.** Understand why covering your tracks is so important. When an attack is carried out against a system, the attacker typically wants to maintain access as long as is possible. In order to maintain this access, they cover the tracks thoroughly to delay the detection of their attack as long as possible.

# Review Questions

1. Enumeration is useful to system hacking because it provides ______________
   - **A.** Passwords
   - **B.** IP ranges
   - **C.** Configuration
   - **D.** Usernames

2. What can enumeration *not* discover?
   - **A.** Services
   - **B.** User accounts
   - **C.** Ports
   - **D.** Shares

3. ___________ involves gaining access to a system.
   - **A.** System hacking
   - **B.** Privilege escalation
   - **C.** Enumeration
   - **D.** Backdoor

4. ___________ is the process of exploiting services on a system.
   - **A.** System hacking
   - **B.** Privilege escalation
   - **C.** Enumeration
   - **D.** Backdoor

5. How is a brute-force attack performed?
   - **A.** By trying all possible combinations of characters
   - **B.** By trying dictionary words
   - **C.** By capturing hashes
   - **D.** By comparing hashes

6. A _________ is an offline attack.
   - **A.** Cracking attack
   - **B.** Rainbow attack
   - **C.** Birthday attack
   - **D.** Hashing attack

7. An attacker can use a(n) ______________ to return to a system.
   - **A.** Backdoor
   - **B.** Cracker
   - **C.** Account
   - **D.** Service

8. A __________ is used to store a password.
   - **A.** NULL session
   - **B.** Hash
   - **C.** Rainbow table
   - **D.** Rootkit

9. A __________ is a file used to store passwords.
   - **A.** Network
   - **B.** SAM
   - **C.** Database
   - **D.** NetBIOS

10. A __________ is a hash used to store passwords.
    - **A.** LM
    - **B.** SSL
    - **C.** SAM
    - **D.** LMv2

11. ____________ is used to partially encrypt the SAM.
    - **A.** SYSKEY
    - **B.** SAM
    - **C.** NTLM
    - **D.** LM

12. Which system should be used instead of LM or NTLM?
    - **A.** NTLMv2
    - **B.** SSL
    - **C.** Kerberos
    - **D.** LM

13. NTLM provides what benefit versus LM?
    - A. Performance
    - B. Security
    - C. Mutual authentication
    - D. SSL

14. ADS requires what to be present?
    - A. SAM
    - B. Domain
    - C. NTFS
    - D. FAT

15. What utility may be used to stop auditing or logging of events?
    - A. ADS
    - B. LM
    - C. NTFS
    - D. Auditpol

16. On newer Windows systems, what hashing mechanism is disabled?
    - A. Kerberos
    - B. LM
    - C. NTLM
    - D. NTLMv2

17. Which is a utility used to reset passwords?
    - A. TRK
    - B. ERC
    - C. WinRT
    - D. IRD

18. A good defense against password guessing is _____________.
    - A. Complex passwords
    - B. Password policy
    - C. Fingerprints
    - D. Use of NTLM

**19.** If a domain controller is not present, what can be used instead?

**A.** Kerberos

**B.** LM

**C.** NTLMv1

**D.** NTLMv2

**20.** Alternate Data Streams are supported in which file systems?

**A.** FAT16

**B.** FAT32

**C.** NTFS

**D.** CDFS