Pentest 5

Pentest 5

Exploits and Attacks

Once you have conducted your initial survey of a target, including mapping out a full list of targets and probing them to identify potential vulnerabilities and weaknesses, the next step is to **analyze that data to identify which targets you will prioritize, what exploits you will attempt, and how you will access systems and devices that you have compromised.** After you have successfully compromised systems, post-exploit activities become important. Knowing how to retain access and conceal your activities and how to leverage the access you have gained to pivot to other systems that may not have been accessible before are all critical to your success.

Choosing Targets

In Chapter 5 you learned how to analyze a vulnerability report, **including reviewing the severity of issues and CVSS scores and looking for missing patches and other issues. A vulnerability scan report is one of a number of components you may include in your target selection process.** In addition, you may consider the primary goals of the penetration test you are conducting; the rules of engagement of the test; any additional data you have already gathered, such as account information or application details; and your own skill set. In most cases, you will target the most vulnerable systems for initial exploits to gain a foothold that may provide further access. In Figure 6.1, you can see an OpenVAS vulnerability scan of a sample highly vulnerable Windows system. This scan result shows 19 critical vulnerabilities as well as other vulnerabilities rated as Medium. In fact, using a normal OpenVAS scan, the system returns a total of 19 High, 61 Medium, and 7 Low issues. If a system like this showed up in a scan, it would be a tempting first target!

OpenVAS/Greenbone vulnerability reportExploits and Attacks

185

Metasploitable: A Handy Pen-Testing Practice System

The system shown in Figure 6.1, which we will use throughout this chapter to demonstrate the exploit process for a Windows server, is a Metasploitable 3 Windows 2008 virtual machine. Metasploitable is an intentionally vulnerable system for penetration test practice. The current version of Metasploitable, version 3, is designed to automatically build Windows Server 2008 and Ubuntu Linux 14.04 virtual machines, but it can be fragile. If you're up to the possible challenge, you can find setup and build instructions at https://github.com/rapid7/metasploitable3.

If you're just getting started with penetration testing, and don't have the time or experience that can be required to work through a sometimes challenging build process, the older version, Metasploitable 2, allows for a direct download of a VMWare or Virtualbox virtual machine (VM) from https://sourceforge.net/projects/metasploitable/

files/Metasploitable2/, which can help you get up to speed more quickly. While Metasploitable 2 is dated, it is useful for basic skills practice. We will make use of it in some examples as well.

In either case, remember to avoid exposing the vulnerable systems you will practice with to an untrusted network, because they are very, very vulnerable.

Identifying the Right Exploit

The system in Figure 6.1 has a very large number of potentially vulnerable services listed. While finding such a large number of vulnerable services exposed on a single system is rare, it isn't uncommon to find many vulnerabilities of varying severity spread across systems in an environment. That makes selecting the right exploit important to make sure that you focus on attacks.

Included in the list are seven vulnerabilities that OpenVAS rates as 9.0 or higher severity, which means that reviewing each of these is likely worthwhile—in fact, almost all of the high-rated vulnerabilities may be worth reviewing. We will focus on the ManageEngine Desktop Central 9 FileUploadServlet connection ID vulnerability shown

While the image is small, you can see it is has a severity of 10 and a quality of detection of 99 percent. Not only does this mean that the vulnerability is severe, but OpenVAS assures us that the vulnerability is correctly detected and is not a false positive. That pairing makes this a very attractive potential target.

While there are other vulnerabilities rated 10, you should also look at lower-rated vulnerabilities that may provide information or allow you to take further actions.

The Metasploitable 2 distribution provides a vulnerable Linux system, which includes a very old version of phpinfo. A scan of the system shows that this vulnerability is rated 7.5,186

Exploit and Pivot

with a quality of detection of 80 percent shown in Figure 6.3. This isn't quite as tempting as the ManageEngine vulnerability, but many vulnerabilities you encounter are more likely to be rated lower because organizations often have programs that patch most high-severity issues.

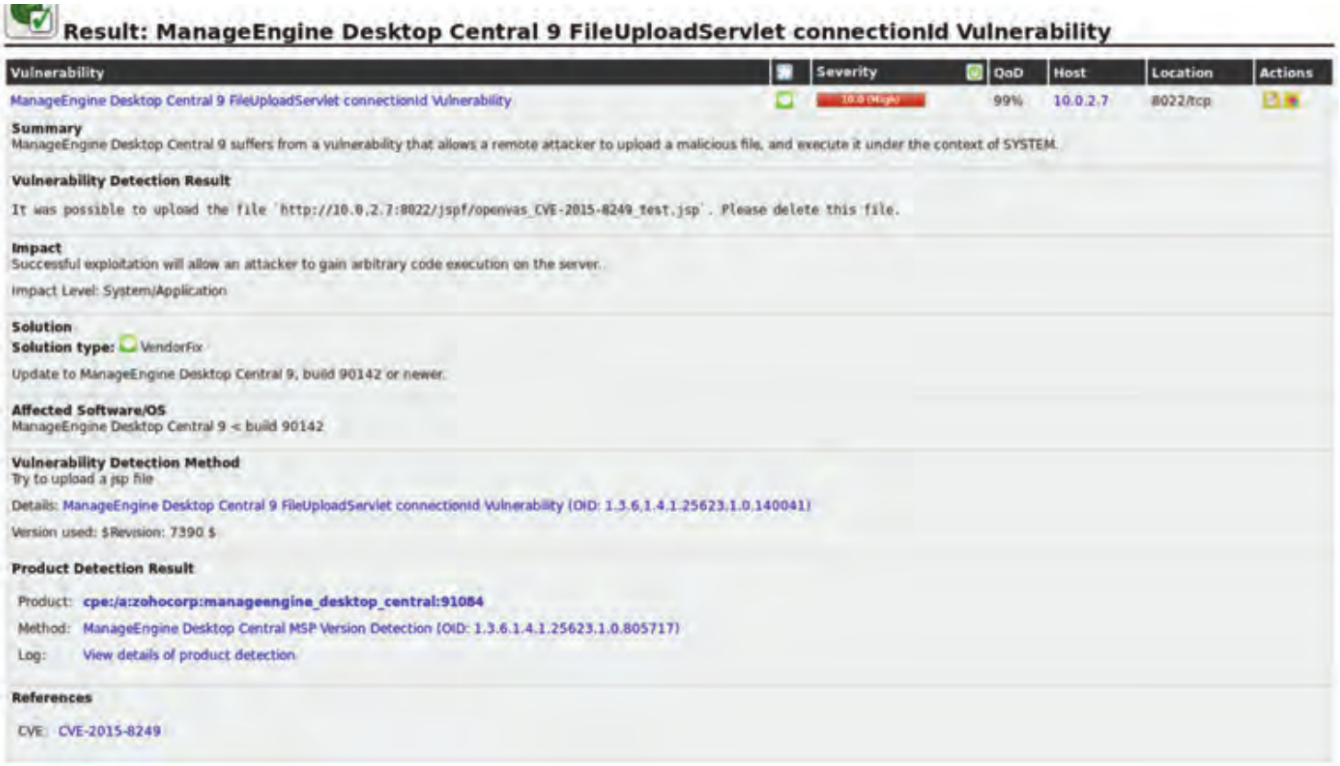
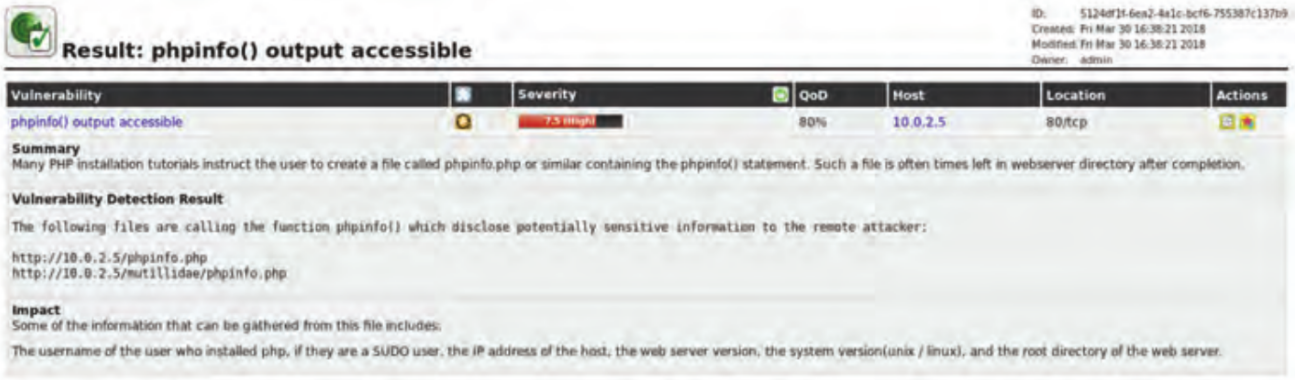


FIGURE 6.3 `phpinfo()` output accessible



The output for `phpinfo()` tells us that this is an information exposure vulnerability rather than a directly exploitable vulnerability. You shouldn't ignore information exposure vulnerabilities, even if they have a lower rating. They're often a great way to gain additional useful information about how a system or application is configured and may provide the details you need to perform further exploits. In fact, this is incredibly easy to test. Figure 6.4 shows the results of visiting the `phpinfo.php` page described in the finding. You should always take advantage of easy information gathering if you can!

.2Distributed Ruby vulnerability

phpinfo() output accessible

The output for phpinfo() tells us that this is an information exposure vulnerability rather than a directly exploitable vulnerability. You shouldn't ignore information exposure vulnerabilities, even if they have a lower rating. They're often a great way to gain addi-

tional useful information about how a system or application is configured and may provide the details you need to perform further exploits. In fact, this is incredibly easy to test. Figure 6.4 shows the results of visiting the phpinfo.php page described in the finding. You should always take advantage of easy information gathering if you can!Exploits and Attacks

phpinfo.php output

Once you have identified the vulnerabilities that you want to target, you can dig into exploits for them. Not every vulnerability has exploit code released, and even when exploit code is released, it can vary in quality and availability.

*Your first thought after reading through Figure 6.4 may have been "Nobody would run an eight-year-old version of PHP!" Unfortunately for system administrators and security professionals, and luckily for penetration testers, many embedded systems and prebuilt software packages include older versions of packages like PHP, .NET, Java, Tomcat, Flash, and other components. Once installed, many remain in place for years without being patched, providing a target for pen-testers and attackers. In fact, during the writing of this book, one of the authors was involved in remediation of an organization that was still actively using Windows 98 syst*ems to control critical equipment on a public, Internet-facing network.188

■ **Exploit and Pivot**

Exploit Resources

Exploits are available in a variety of places, ranging from personal sites to central collections. In addition to these, an increasing number of major vulnerabilities and exploits have their own publicity sites; examples include the Meltdown and Spectre bugs from 2017 (https://meltdownattack.com/). Many exploits are hosted on sites like GitHub, with direct code download available as part of the initial vulnerability disclosure from the individual or organization who discovered it. Exploit quality varies: some exploits require specific configurations or circumstances to work properly, while others simply work with minimal effort. As a penetration tester, you will need to learn how to assess the quality of exploits that you intend to use, and you will need to plan for some of the exploits you attempt to fail.

Downloading exploits can be dangerous, since it can be very challenging to verify that they have not had malware embedded in them by malicious actors. While some sites will provide an MD5 or SHA1 hash of the exploit fi les, others will simply provide a download or point to a code repository.

Of course, anti-malware tools often identify exploit code as malicious because it is used for attacks or includes tools that are commonly associated with malware or malicious activity!

Fortunately, there are a number of large central sites that specialize in making exploits and vulnerabilities searchable.

The Exploit Database

The Exploit Database (www.exploit-db.com) is one of the largest public exploit databases. It includes exploits, shellcode, and a variety of security papers as well as the Google Hacking Database, a collection of useful search techniques (often known as "Google dorks") for penetration testers and security professionals.
If you want to take the Exploit Database with you, you can! SearchSploit is a command line search tool that works with a local copy of the Exploit Database. Kali Linux already includes Exploit-DB by default. To use SearchSploit in your own Linux system, all you need to do is install it using git:
git clone https://github.com/offensive-security/exploit-database.git/opt/exploit-database
For more details, and instructions for other operating systems, visit https://www.exploit-db.com/searchsploit/#what.

The Rapid7 Vulnerability and Exploit Database

For Metasploit users, the Rapid7 Vulnerability and Exploit Database (https://www.rapid7.com/db) is a very useful tool, thanks to its integration with Metasploit exploits forExploits and Attacks

both the Metasploit framework and Metasploit Pro. If you intend to use Metasploit to drive your penetration test, the ability to search directly for exploits based on vulnerabilities you have found during a scan can speed up your planning and exploit process.

The National Vulnerability Database

NIST maintains the National Vulnerability database at http://nvd.nist.gov. The NVD is an excellent vulnerability resource, but it does not focus on the availability of exploits as much as the other resources mentioned so far. While exploits may be listed in the references section, they are not the focus of the NVD.

VULDB

Another option for vulnerability searches is http://vuldb.com, a large crowd-sourced vulnerability database. Unlike the other databases, VulDB includes an estimated exploit price and price rankings. This additional data can help penetration testers understand where market focus is and can be a leading indicator of what exploits may become available in the near future.

Building a Vulnerable Machine

In this *chapter, we will be using both Metasploitable 2 and Metasploitable 3, Rapid7's freely available vulnerable virtual machines. Instructions to build your own Metasploitable virtual machine for Virtualbox or VMware can be found at https://github.com/rapid7/metasploitable3; however, the build process can be challenging. The authors of this book found the instructions at https://andrusk.com/building-metasploitable-3-on-ubuntudebian/ useful for building in Ubuntu Linux and recommend the manual instructions for Windows to improve your chances of success. Once you have a working version of Metasploitable, you can see the full list of vulnerable services, along with credentials and other information, at https://github.com/rapid7/metasploitable3/wiki/Vulnerabilities, which you can practice against.*

If you find Metasploitable 3 challenging to set up, you can substitute Metasploitable 2

from https://sourceforge.net/projects/metasploitable/files/Metasploitable2/; however, instructions in this chapter are based on Metasploitable 3.

While deliberately vulnerable machines are useful, you can also simply download and install an older version of a common operating system. Unpatched versions of Windows (XP, 7, 2008 Server) and older Linux distributions make great targets too!

Developing Exploits

When a vulnerability is discovered and reported, the announcement often includes details of how and why the issue occurs. Based on this information, exploit developers can then probe the software, service, or tool that the vulnerability impacts. Once a developer has verified their ability to replicate the issue, they can then test it to see what can be done if the bug is exploited. Exploit developers look for ways to gain access to a service or administrative account, ways to modify memory to execute arbitrary code, and a variety of other ways to break security boundaries and isolation.

Once an exploit developer has identified both a way to exploit the vulnerability and what they can do with it, the next step is typically to make the exploit repeatable and reliable. This can be difficult, as some flaws may not consistently work or may require specific settings or circumstances to work properly. A highly reliable exploit is obviously more valuable than one that only works a small percentage of the time.

The good news for the PenTest+ exam is that you shouldn't need to develop an exploit from scratch. Instead, you need to know what is needed to develop an exploit, along with the basics of how you might modify an exploit to meet the needs of a penetration test you are conducting.

If you want to read more about how to write exploits, Corelan has a complete exploit writing tutorial in its Exploit Writing Tutorials section at https://www.corelan.be/index.php/articles/. FuzzySecurity covers the Windows side of things very well at http://www.fuzzysecurity.com/tutorials.html.

Exploit Proof-of-Concept Development

Proof-of-concept exploits are designed to validate that an exploit can be successful, and are often not built to be reliable or consistently repeatable. In fact, they just need to show that there is a flaw. Unlike the exploits we have looked at elsewhere in this chapter, a proof-of-concept exploit typically won't have the ability to deliver a useful payload and will instead focus on providing an easily visible indication of success. If you want to learn more about a real-world example of how to build a simple proof-of-concept exploit, https://www.anitian.com/blog/a-study-in-exploit-development-part-1-setup-and-proof-of-concept/ includes a complete walk-through that shows how Rick Osgood identified, built, and tested a proof-of-concept exploit.

While the PenTest+ exam describes exploit writing and modification, the ability to design and build exploits from the ground up is a very specialized skill set and requires lots of practice and experience—and it is beyond the realistic scope of an exam like the PenTest+. If you want to learn more, organizations like SANS also offer in-depth courses on the subject, including **Advanced Exploit Development for Penetration Testers (https://www.sans**

.org/event/cyber-defense-initiative-2018/course/advanced-exploit-development-penetration-testers). That's a 700-level class from SANS, which makes it one of their most advanced courses. As you might have guessed, exploit development isn't for the faint of heart!Exploitation Toolkits

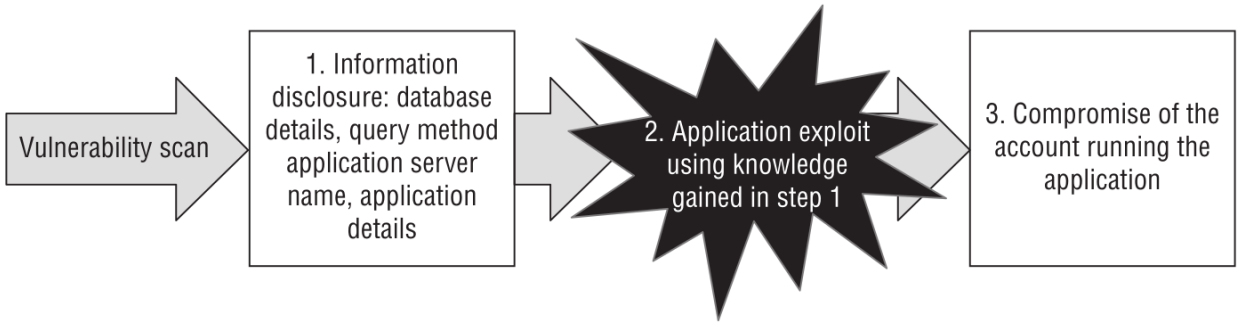
Exploit Modification

Exploit and payload modification is sometimes needed when an exploit either requires configuration or changes for the environment that you are targeting, or if the exploit doesn't fit the specific vulnerability you are targeting. Proof-of-concept exploits and early exploit releases are common examples of exploits that a penetration tester may need or want to modify. Fortunately, exploits like those used in the Metasploit Framework, which we will discuss in a few pages, are created in a common format, allowing easier modification.

Exploit Chaining

Exploit chaining requires you to use a series of exploits to gain information, privileges, or access. A frequent path through an exploit chain is shown in Figure 6.5. In this example, a penetration tester leverages an information disclosure vulnerability that discloses information about a backend database, the application server, and the application. The penetration tester then uses that information to attack the application, gaining control of the account that the application runs under. In most cases, the next step in the chain would be privilege escalation to gain additional access if possible. Other exploit chains may chain specific vulnerabilities together like an injection attack to get access to a memory stack vulnerability to create a successful exploit.

FIGURE 6.5 Exploit chaining



Exploitation Toolkits

Penetration testers need to deal with large numbers of targets in a way that allows them to use both exploits and exploit payloads effectively to compromise systems and retain access to them. Exploit toolkits play a bit role in that for many testers. Effective exploit toolkits combine prebuilt exploit modules, the ability to add and customize exploits in a common format, and a wide range of tools that make you a more effective penetration

Exploit and Pivot

Metasploit

One of the most powerful exploit tools in a modern penetration tester's arsenal is Metasploit. For most penetration testers, Metasploit is the default exploit tool in their arsenal, and it has become the most common development framework for exploits, with Metasploit plug-ins being released shortly after many major vulnerability announcements. If you're using Kali Linux, Metasploit is already built in. If you are using another Linux distribution and need to install Metasploit, or you need to install it on a target system, you can download it from https://information.rapid7.com/metasploit-framework.html.

There are two major versions of Metasploit available today: the Metasploit framework, a free, open-source version of Metasploit; and Metasploit Pro, a commercial version with enhanced features. Additional versions include Metasploit Community, a free web user interface for the Metasploit framework; Metasploit Express; and Armitage, a graphical interface for Metasploit that focuses on team collaboration for exploitation and penetration testing. We will focus on the freely available Metasploit framework in this book.

Metasploit includes tools and features that allow for more than just exploitation. In fact, Metasploit capabilities include discovery (Nmap and other tools), exploitation, payload delivery, and tools to help avoid detection.

Metasploit Basics

Metasploit has a multitude of features, but its basic use is relatively simple. At a high level, there are four main activities you need to know how to do:

■✓Start the consolemsfco

■✓Select an exploit

■✓Select a payload

■✓Run the exploit

We will explore this process over the next few pages, but you should bear in mind that Metasploit is complex enough to fi ll an entire book with its capabilities and uses. While we'll cover one scenario, you should practice with other exploits based on the vulnerability scans you have run previously. Make sure you focus on selecting a vulnerability, fi nding an exploit, and then exploiting it on a vulnerable target machine.

Metasploit is a very powerful tool, and learning everything Metasploit has to offer could fill a book all by itself. We'll cover the basics of using Metasploit, but if you want to learn more, Offensive Security has a great Metasploit Unleashed guide available at https://www.offensive-security.com/metasploit-unleashed/. If you want to dig deeper with Metasploit, we highly recommend Metasploit Unleashed .Exploitation Toolkits

Starting Metasploit

Starting Metasploit is simple—just enter the command msfconsole and wait for the msf> prompt to appear, as shown in Figure 6.6.

The Metasploit console

Metasploit has quite a few different initial load screens, so the image you

see in Figure 6.6 may not match the screen that you'll see. Don't worry—and if you want to skip the ASCII art, just use the msfconsole -q option for quiet mode.

shows the start screen, including the number of exploits and payloads that are loaded. If you've recently visited the Exploit-DB site, you'll notice that there are far fewer exploits included in Metasploit than exist on the ExploitsDB site. Exploits for Metasploit have to be built in the Metasploit framework, and they need to be usable in ways that match Metasploit's capabilities. As a result, fewer exploits are built for Metasploit, but they are more generally useful.

Once you have Metasploit started, you can review the commands available to you by typing a question mark and hitting Enter.

Selecting Your Exploit

In most cases, the next step toward a successful exploit is to search for your exploit. Earlier in this chapter we looked at OpenVAS output for a Metasploitable 3 system including a ManageEngine fi le upload vulnerability. Now you can use that vulnerability data to guide your exploit selection.

If you'd like to see the full list of exploits that are loaded, you can use the show exploits command shown in Figure 6.7. The output can be a bit overwhelming, since we have over 1,600 exploits loaded, but understanding how Metasploit lists and ranks exploits is helpful.194

Exploit and Pivot

Running show exploits in Metasploit

As you can see, each exploit has a name, which includes a hierarchical naming convention. The first exploit on the list is aix/local/lib-stat_path—this means it is an exploit for AIX, it is a local exploit, and it exploits the libstat path privilege escalation bug found on some AIX systems.

Next, you'll see the disclosure date, the rank, and a description of the exploit. The ranking is important! It describes how likely the exploit is to be successful and what impact it may have on the target system, as shown in Table 6.1.

Metasploit exploit quality ratings

RankDescription

ExcellentThe exploit will never crash the service.

GreatThe exploit has a default target and will either autodetect the target or perform a version check and use an application-specific return address.

GoodThe exploit has a default target and is the common case for the software.

NormalThe exploit is reliable but requires a specific version that can't be reliably autodetected.

AverageThe exploit is unreliable or difficult to exploit.

LowThe exploit is very difficult or unlikely to successfully result in an exploit (less than a 50 percent chance) for common platforms.

ManualThe exploit is unstable, difficult to exploit, or may result in a denial of service,

OR the exploit requires specific manual configuration by the user.Exploitation Toolkits
In general, this means that most penetration testers will focus on exploits that are ranked as normal or higher and that using exploits ranked Good, Great, or Excellent is preferable. Fortunately, Metasploit has the ability to filter exploits based on their built-in ranking. If you want to search only for exploits that are rated Great, you can search for them using the search -r great command or set a filter to only allow exploits of that level to be run by entering setg MinimumRank great.
Searching for Exploits
You can search for exploits inside Metasploit itself using the Search command. This command includes a number of keywords that make searches much easier, shown in Table 6.2.

Metasploit search terms
KeywordDescription
appClient or server attack
authorSearch by module author
bidSearch by Bugtraq ID
cveSearch by CVE ID
edbSearch by Exploit-DB ID
nameSearch by descriptive name
platformSearch by platform (Windows, Linux, Unix, Android, etc.)
refModules with a specific ref
typeSearch by type: exploit, auxiliary, or post
Searching for exploit in Metasploit can sometimes take some work. The OpenVAS listing for the ManageEngine vulnerability shows a CVE number of CVE-2015-8249, which is a good place to start, but if you type search type:exploit cve:cve-2015-8249, you won't find anything. In fact, not every exploit is fully documented in Metasploit with CVE, BID, EDB, and other details in place. Fortunately, other options exist. A bit of searching reveals that the exploit was created by sinn3r, so entering search type:exploit author:sinn3r will show us the results we want, including exploit/windows/http/manageengine_connectionid_write, the exploit we need.
In addition to the built-in command-line search, Rapid7 also makes a web-based exploit database search available at https://www.rapid7.com/db/modules/. Finding the196

M**anageEngine exploit there is simply a matter of entering ManageEngine and selecting Metasploit Modules from the drop-down search list.**
Now that you have identified the exploit you want to use, telling Metasploit to use it is simple. At the msf> prompt, type use exploit/windows/http/manageengine_connectionid-write as shown in Figure 6.8.

Selecting an exploit
Tab completion works in Metasploit, so take advantage of it to make selection of modules easier.
Selecting a Payload

A payload in Metasploit is one of three types of exploit modules: a single, a stager, or a stage. Singles are self-contained payloads, which will often do something simple like add a user or run a command, and are the simplest payloads to deploy. Stagers set up a network connection between the target and a host. Stagers use stages, which are payloads that they download to pull in bigger, more complex tools.

In addition to the three types of exploit modules, there are eight types of payloads:

bbbb

Inline payloads are single payloads, and include the exploit and payload in a single package.

Staged payloads work well for memory-restricted environments and load the rest of the payload after landing.

Meterpreter is a powerful payload that works via DLL injection on Windows systems and remains memory resident.

PassiveX uses ActiveX via Internet Explorer and is becoming largely deprecated, although occasional systems may still be vulnerable to it.

NoNX payloads are designed to counter modern memory protection like Data Execution Prevention (DEP) or Windows No Execute, or NX.

bvnvbnbv

■✓==**IPv6 payloads are designed for IPv6 networks.**==

■==✓**Reflective DLL injection modules also target Windows systems and run in memory only.**==

The default payload for this package is the Metasploit Meterpreter, so all we need to do is run the exploit to get Meterpreter in place.

To see the full list of Metasploit payloads, you can use the show payloads command at the msf> prompt before selecting an exploit to display screen after screen of payloads designed for Windows, Unix/Linux, and other operating systems.Exploitation Toolkits

Module Options

Many Metasploit modules have options that can be set. For our module to work properly, we need to check the options and set them (Figure 6.9).

Figure 6.9

Setting module options

This module includes an rhost setting, which is our remote target host. In some cases, you may need to set the rport setting, particularly if your target is running the vulnerable service on a nonstandard port. Finally, some modules may need a target ID set. In this case, since it is a Windows-specific exploit, the exploit module in use only sets a single target ID for Windows rather than offering options.

Exploitation

With an exploit and payload selected, you can attempt the exploit using the exploit command, as shown in Figure 6.10. Note that this exploit uses Meterpreter as its default payload and that we now have a powerful exploit package to use—and that Meterpreter cleaned up after itself by removing the Meterpreter upload. Since Meterpreter runs in memory,

there will not be evidence of the exploit in the target service directory! You can read more about Meterpreter at https://www.offensive-security.com/metasploit-unleashed/meterpreter-basics/.

Successful exploit

Exploit and Pivot

Once connected, Meterpreter offers the ability to attempt to escalate privileges with the getsystem command. If that fails, shell access is available by simply typing shell, which drops you to a shell in the directory that the exploited service runs in, C:\ManageEngine\DesktopCentral_Server\Bin, allowing you to take further actions from there.

PowerSploit

PowerSploit is a set of Windows PowerShell scripts that are designed to provide capabilities including antivirus bypass, code execution, exfi ltration, persistence, reverse engineering, and reconnaissance. Much like Metasploit, PowerSploit is a very powerful, flexible tool. Like many of the tools penetration testers use, PowerSploit will be picked up by Windows Defender and other anti-malware tools as soon as you download it. Turn off your AV if you want to avoid this—and remember to keep the system you use secure!

Fortunately for our purposes, Kali Linux also includes PowerSploit in the Applications ➢ Post Exploitation menu. This will drop you to a terminal window in /usr/share/PowerSploit. From there, you can run a simple Python web server to expose PowerSploit tools to Windows systems by running python -m SimpleHTTPServer, and then use an existing Meterpreter session on the remote Windows system to use PowerSploit tools.

If you have administrative access to a remote Windows workstation or server, PowerSploit can provide the toolkit you need to maintain persistence and to perform further reconnaissance. *One of the most popular tools to use with PowerSploit is the implementation of Mimikatz functionality that it includes as part of the Invoke-Mimikatz PowerShell script. This script injects Mimikatz into memory and then allows you to dump credentials without having Mimikatz on disk, where it could be discovered by antivirus that is monitoring disk activity. Once you have this functionality in memory, you can use typical Mimikatz functions like LSASS credential dumping, private certificate acquisition, and even acquisition of debug credentials. We will take a closer look at Mimikatz later in this chapter.*

The PenTest+ exam objectives also specifically call out Empire, a PowerShell- and Python-based post-exploitation tool. Empire uses encrypted communications and allows PowerShell agents to run without powershell.exe, and it has quite a few modules designed to help with post-exploitation activities on Windows systems. You can read more about Empire at http://www.powershellempire.com/ and on the Empire wiki at https://github.com/EmpireProject/Empire/wiki/Quickstart. Since we already cover PowerSploit in this chapter, we won't dig further into Empire—but you should be aware that it is another tool with similar functionality and a Metasploit-like interface.Exploit Specifics

Exploit Specifics

The PenTest+ exam objectives specifically mention a number of exploits that you should be prepared to encounter on the exam. These are discussed in the following sections.

RPC/DCOM

Historically, RPC/DCOM (Remote Procedure Call/Distributed Component Object Model) exploits were a common way to attack Windows NT, 2000, XP, and 2003 Server systems, and even modern attack tools often have RPC/DCOM exploits available. More modern exploits tend to focus on other elements, such as the .NET interoperability layers for DCOM. While occasionally RPC/DCOM vulnerabilities continue to appear, and exploits are often written for them, **RPC/DCOM exploits are far less common today.**

PsExec

The Sysinternals Windows toolkit includes PsExec, a tool designed to allow administrators to run programs on remote systems via SMB on port 445. That makes it an incredibly useful tool if it is available to you during a penetration test, as you can execute arbitrary commands, up to and including running an interactive shell. Unfortunately for modern attackers, this has been abused so much over time that most anti-malware tools will flag **PsExec the moment it lands on a system.**

A number of Metasploit exploit modules also reference PsExec, which isn't actually the Microsoft Sysinternals tool. Instead, the Metasploit PsExec exploit embeds a payload into a service executable, connects to the ADMIN$ share, uses the Service Control Manager to start the service, loads the code into memory and runs it, and then connects back to the Metasploit machine and cleans up after itself! For an in-depth look at this and related techniques, visit https://toshellandback.com/2017/02/11/psexec/.

PS Remoting/WinRM

Modern Windows systems running Windows 7 or later use Windows Remote Management (WinRM) to support remote PowerShell command execution. For a penetration tester, being able to run PowerShell commands on remote systems is very handy, but this feature has to be turned on first. Fortunately, it is simple. Remote PowerShell command execution can be turned on using the enable-PSRemoting -force command while running PowerShell as an administrator.

If the systems aren't part of the same domain, you will still have to set up trust between them using the TrustedHosts setting:

Set-Item wsman:\localhost\client\trustedhosts [ipaddress or hostname]200

Exploit and Pivot

Once you have done this, you have to restart WinRM, and then you can run remote PowerShell commands at will. For a penetration tester, this can make further exploits and retaining access even easier, as long as it remains undetected.

WMI

Windows Management Instrumentation (WMI) allows for remote management and data gathering installed on all Windows systems, making it an attractive target for penetration testers and attackers. WMI provides access to a huge variety of information, ranging from Windows Defender information to SNMP to Application Inventory listings. WMI can

allow remote execution of commands, fi le transfers, and data gathering from fi les and the Registry, among many other capabilities. Multiple PowerShell tools have been written to exploit WMI, including WMImplant and WmiSploit.
WMImplant has a number of useful functions for lateral movement, including information gathering using basic_info and checks to see if there is a logged-in user via vacant_system, as shown in Figure 6.11.

WMImplant WMI tools
The best way to learn more about WMI tools like these is to install them on a test host like the Metasploitable 3 virtual machine and then use them to gather information about the host and other systems.
Scheduled Tasks and cron Jobs
Using scheduled tasks to perform actions on a compromised Windows host is a tried-and-true method of retaining access. The same is true of cron jobs on Linux and Unix hosts, and this means that defenders will often monitor these locations for changes or check them early in an incident response process. That doesn't mean that the technique isn't useful—it merely means that it may be detected more easily than a more subtle method; but unlike memory resident exploits, both scheduled tasks and cron jobs can survive reboots.Exploit Specifics

To schedule a task via the command line for Windows, you can use a command like this, which starts the calculator once a day at 8:00 a.m.:
SchTasks /create /SC Daily /TN "Calculator" /TR "C:\Windows\System32\calc.exe" /ST 08:00
*The same technique works with Linux or Unix systems using cron, although cron keeps multiple directories in /etc/ on most systems, including /etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly, and /etc/cron.monthly. Scripts placed into these directories will be executed as you would expect based on the name of the directory, and the scripts can include a specific number of minutes after the hour, the 24-hour military time, the day of the month, the month, the day of the week, and the command to run. Thus 0 30 1 * * /home/hackeduser/hackscript.sh would run the fi rst day of every month at 12:30 a.m. and would execute hackscript.sh in the /home/hackeduser directory. Of course, if you're trying to retain access to a system, you'll want to be a lot more subtle with fi lenames and locations!*
One of the most common uses of this type of scheduled task is to retain access to systems via a remotely initiated "call home" script. This prevents defenders from seeing a constant inbound or outbound connection and can be used to simply pick up fi les or commands from a system that you control on a regular basis.
The PenTest+ test outline doesn't mention NFS (Network File System) shares, but NFS exploits are worth remembering while conducting a penetration test. Servers often use NFS mounts for shared filesystems or to access central storage, and improperly configured or secured NFS shares can provide useful information or access. If you find TCP ports 111 and

2049 open, you may have discovered an NFS server.

SMB

Server Message Block (SMB) is a fi le-sharing protocol with multiple common implementations. Historically, Windows implemented it as CIFS (Common Internet File System), with modern systems using SMB 2 or SMB3, while Linux uses Samba. In each case, the underlying protocol is the same, with slight differences in implementation and capabilities. Since SMB provides name resolution, fi le services, authentication, authorization, and print services, it is an attractive target for penetration testers who want access to remote systems that provide SMB services.

If you discover SMB services, the variety of implementations makes identifying the host operating system and the SMB implementation important when attempting exploits. Gathering information from open shares and services doesn't require that knowledge. Kali Linux includes SMB Scanner, and Metasploit has SMB scanning capabilities built in that can do everything from brute-force logins to enumerating SMB services.202

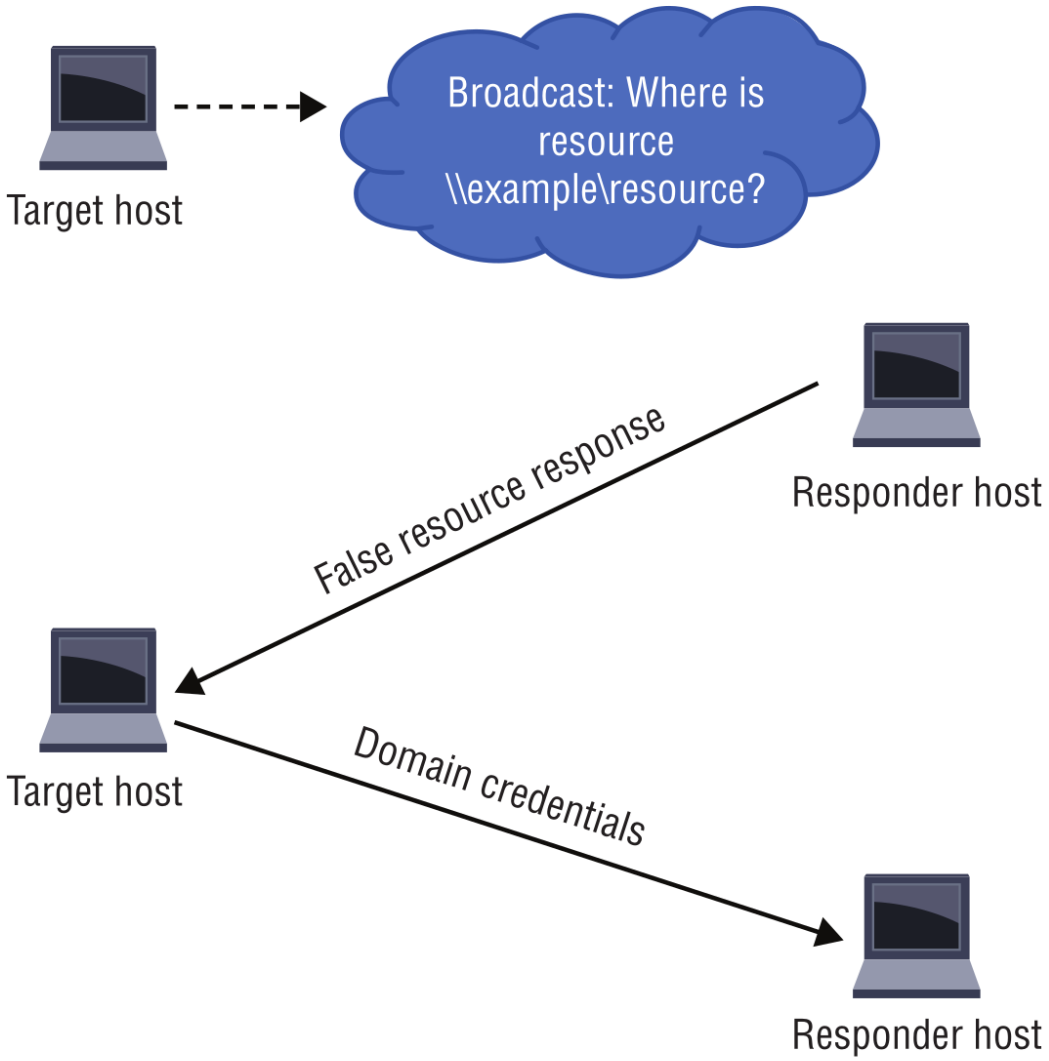
\

Credentials for SMB can be acquired by tools like Responder, which reply to queries for resources as shown in Figure 6.12. This exploits the trust in a service response to tell the client that the responder host is a legitimate service provider, causing it to send its hashed credentials, which the owner of the Responder host can then use to authenticate to legitimate servers.

Responder capture flow

Broadcast: Where is
resource
\\example\resource?

Target host



Similar tools exist in Metasploit, which means that in many cases you can use a single tool to provide many of the functions you might otherwise need multiple specialized tools to accomplish.

Once you have hashed credentials in hand, you can replay them to servers, in plaintext, Kerberos, or NTLM modes, with tools like Impacket.

Core's Impacket toolset provides many functions besides simple SMB hash playback. In fact, it includes tools that create WMi persistence, dump secrets from remote machines with clients, handle MS-SQL authentication, and replicate PsExec services.

DOM-Based XSS

DOM-based XSS is similar to reflected XSS, except that in DOM-based XSS, the user input never leaves the user's browser. In DOM-based XSS, the application takes in user input, processes it on the victim's browser, and then returns it to the user.

The Document Object Model (DOM) is a model that browsers use to render

a web page. The DOM represents a web page's structure; it defines the basic properties and behavior of each HTML element, and helps scripts access and modify the contents of the page. DOM-based XSS targets a web page's DOM directly: it attacks the client's local copy of the web page instead of going through the server. Attackers are able to attack the DOM when
Cross-Site Scripting
117a page takes user-supplied data and dynamically alters the DOM based on that input. JavaScript libraries like jQuery are prone to DOM-based XSS since they dynamically alter DOM elements.

Reflected XSS

Reflected XSS vulnerabilities happen when user input is returned to the user without being stored in a database. The application takes in user input, processes it server-side, and immediately returns it to the user.

The first example I showed, with the email form, involved a reflected XSS attack. These issues often happen when the server relies on user input to construct pages that display search results or error messages. For example, let's say a site has a search functionality.

Windows Remote Desktop (RDP) exploits are rare but powerful. The 2017 release of the EsteemAudit remote access exploit only worked on Windows 2003 and XP instead of modern Windows operating systems. Thus, most penetration testers focus on existing accountsExploit Specifics

rather than the service itself as their target. Captured credentials and an accessible RDP (TCP/UDP port 3389) service provide a useful path into a Windows system, particularly Windows servers, which often use RDP as a remote administration access method.

Apple Remote Desktop

Remote access tools like RDP and ARD, Apple's Remote Desktop tool, provide a great way to get GUI access to a remote system, but when they are vulnerable, they can create an easy route in for attackers. Penetration testers use **?":
**in two major ways. The first is via
known vulnerable versions that can be exploited for access. Examples include the version built into MacOS 10 High Sierra, which included a remote root exploit via Screen Sharing or Remote Management modes for ARD. Unfortunately for penetration testers, most modern Macs are set to update automatically, making the vulnerability less likely to be available for many Macs, despite the existence of a Metasploit module that makes using the vulnerability easy.

ARD is also useful as a remote access method for compromised MacOS systems and may present a way for a penetration tester to log into a Mac remotely using captured credentials if the service is running and exposed in a way that you can get to it.

VNC

Virtual Network Computing (VNC) is another common remote desktop tool. There are quite a few variants of VNC, including versions for Windows, MacOS, and Linux. Like

RDP and ARD, VNC provides a handy graphical remote access capability, but it may also have vulnerabilities that can be exploited, and it offers a way for an attacker to use captured credentials or to attempt to brute-force a remote system. Metasploit also includes VNC payloads, making VNC one of the easier means of gaining a remote GUI when delivering a Metasploit payload.

X-Server Forwarding

X11, or X-Windows, often simply called X, is the graphical windowing system used for many Linux and Unix systems. X sessions can be forwarded over a network connection, passing along an entire desktop or a single application. In most modern use, this is done via an SSH tunnel, but X sessions that are not secure can be captured and exploited through session hijacking or capture.

Telnet

T**elnet is an unencrypted service that provides remote shell access. Because the service is unencrypted, Telnet connections can be sniffed to capture credentials if they are in use. Simply finding Telnet accessible on a remote system does not mean that there is a vulnerability, but it does mean that you can target any logins if you can find an intermediate host that can capture network traffic bound for the Telnet server.204**

SSH

SSH (Secure Shell) provides remote shell access via an encrypted connection. Exploiting it normally relies on one of two methods. The first looks for a vulnerable version of the SSH server. If the SSH server service is vulnerable, various issues can occur, including credential exposure or even remote access. Replacing the SSH server service with a Trojaned or modified version to capture credentials or provide silent access is also possible if you are able to gain sufficient access to a system.

Another common SSH attack method is through the acquisition of SSH keys and their associated passphrases from compromised hosts or other exposures. S*SH keys are often* *shared inside organizations, and once they are shared they often remain static without a regular change process.* ***This means that capturing an SSH key, particularly one that is embedded into scripts or otherwise part of an organization's infrastructure, can result in long-term access to the system or systems using that key. Since SSH keys that are shared sometimes have blank passphrases, or the passphrases are distributed with the shared key, even that layer of security is often compromised.***

Going Back in Time: rsh and rlogin

The PenTest+ exam objectives include both rsh (remote shell) and rlogin (remote login); however, very few modern environments are likely to have either of these legacy services enabled. In fact, almost every security baseline released in the past decade includes specific guidance to turn off services like these. Current systems use SSH for remote login, service calls, and other remote usage.

If you do encounter rsh, rlogin, rexec, or any of the other remote services, there's a good chance you've encountered a poorly maintained or legacy system—and thus a good target.

Leveraging Exploits

Once they have successfully used an exploit and have access to a system, penetration testers

will typically investigate their options for lateral movement and post-exploit attacks. Post-exploit attacks may be aimed at information gathering, privilege escalation, or even lateral movement on the same host to gain a broader perspective or to attempt to test security boundaries that exist for the account or service that was originally exploited.

Common Post-Exploit Attacks

There are many ways to conduct post-exploit attacks that can provide further access or information. Understanding the basics of each of these techniques and when it is typically used can help you better deploy exploits.Leveraging Exploits

You may run across a cracking and attack tool called Cain and Abel while reading older security materials and briefings. The tool itself was popular for earlier versions of Windows up to Windows XP, but it is no longer useful for modern Windows systems, including Vista, 7, 8, and 10.

Password attacks come in many forms, ranging from attacks against an authentication system or login page to attacks that are focused on captured credential stores and password fi les. While acquiring a password without having to crack it is always preferable, sometimes the only way into a system is through a more direct password attack. Two of the most common attacks that don't rely on credential theft or social engineering are brute-forcing and the use of rainbow tables on password stores.

Common methods of acquiring passwords from a compromised machine include these:

■✓

■✓

■✓

■✓

■✓

pwdump and related utilities that acquire Windows passwords from the Windows Security Account Manager, or SAM.

Information about user accounts on Linux or Unix systems can be obtained from /etc/passwd and the hashed values of the passwords from /etc/shadow.

cachedump and creddump utilities focus on retrieving stored domain hashes, passwords, or other cached information from caches or the Windows Registry.

SQL queries against system views or database administrative tables can provide information about users, rights, and passwords depending on the database and schema in use.

Sniffing passwords on the wire is less frequently useful in modern networks because encryption is used for many, if not most, authentication systems. It remains a worthwhile tool to try if it's accessible, since sniffing traffic can help pen-testers map networks and applications, and some credentials are still passed in plaintext at times!

mimikatz

Mimikatz is one of the premiere Windows post-exploitation tools. Because of its broad utility and popularity, it is available in a variety of forms, including as a Meterpreter script, as a stand-alone tool, and in modified forms in various PowerShell tools like **Empire and PowerSploit. Mimikatz can retrieve cleartext passwords and NTLM hashes, conduct Golden Ticket attacks that make invalid Windows Kerberos sessions valid, and perform**

other functions that can make post-exploitation Windows hacking a penetration tester's dream. The Offensive Security Metasploit Unleashed documentation includes a good ==introduction to the embedded version of Mimikatz at https://www.offensive-security== .com/metasploit-unleashed/mimikatz/.

Credential brute-forcing relies on automated tools to test username and password pairs until it is successful. There are quite a few tools that penetration testers frequently use for206 Exploit and Pivot this, including THC-Hydra, John the Ripper, and Brutus. In addition, Metasploit includes a brute-force capability as part of its toolkit.

How you track and manage passwords is important for larger penetration tests where you may gather hundreds or thousands of passwords. Matching user accounts to passwords and hosts is also important, as credential reuse for further attacks is a powerful technique when targeting organizations.

==*Using a tool li*==ke John the Ripper can be quite simple. Figure 6.13 shows John in use against an MD5-hashed password fi le from the 2012 Crack Me If You Can competition using the RockYou word list, which is built into Kali Linux.

John the Ripper

Building a custom word list is a common technique when targeting a specific organization and can make documents and other data gathered during the information-gathering stage more useful. Remember to pay attention to cracked and captured passwords to identify patterns, commonly reused passwords, and other information that may improve your password-cracking capabilities.

If you want to try cracking a password file, the 2012 Crack Me If You Can files mentioned above can be found at https://contest-2012 .korelogic.com/. Instructions on how to use John the Ripper can be found at http://www.openwall.com/john/.

Dictionary attacks rely on a prebuilt dictionary of words like the RockYou dictionary mentioned earlier. In many cases, penetration testers will add additional organization-specific dictionary entries to a dictionary fi le for their penetration test based on knowledge they have about the organization. If you know common words or phrases that are likely to be meaningful to staff at the target organization, such as a motto, popular figure or term

,Leveraging Exploits

or even simply a bad habit of staff of the organization's help desk when they reset passwords, those can be very useful for this type of attack. If you don't have that type of information, there is good news: many users who are allowed to set their own passwords use poor passwords, even with complexity rules, and as long as you're not fighting multifactor authentication, there's a good chance you'll be able to crack at least some passwords easily using a dictionary-based attack!

Rainbow tables provide a powerful way to attack hashed passwords by performing a

lookup rather than trying to use brute force. A rainbow table is a pre-computed listing of every possible password for a given set of password requirements, which has then been hashed based on a known hashing algorithm like MD5. *While hashes can't be reversed, this bypasses the problem by computing all possible hashes and then simply using a speedy lookup capability to fi nd the hash and the password that was hashed to create it! Of course, if your target follows password hashing best practices and uses salts and purpose-built password hashing algorithms, it is possible to make rainbow tables much harder to use*, if not impossible. Fortunately for penetration testers, that's not as common as it should be!

If you're not familiar with the concept of password hashing, you'll want to read up on it, as well as password hashing and storage best practices. Despite years of best practice documentation like the OWASP Password Storage Cheat Sheet (https://www.owasp.org/index.php/Password_Storage_Cheat_Sheet==) and training for IT practitioners, organizations== continue to use un-salted MD5 hashes for password storage, leading to massive breaches!

Cross compiling code is used when a target platform is running on a different architecture than the host that you can build an exploit on. During a penetration test, you may gain administrative access to an x86 architecture system and then need to deploy an exploit to an Android device running on an ARM64 platform. If you can't sneak the compiled binary for the exploit through your target's security, you may be able to transfer the source code—or even replicate it on the compromised remote system.

The term cross compiling may make you think of "portable code" that would run on multiple platforms. Actual cross compiling like gcc can compile to multiple architectures, but the binaries will only work on the target architecture.

Privilege Escalation

Privilege escalation attacks come in many forms, but they are frequently categorized into two major types: **vertical and horizontal escalation.** Vertical escalation attacks focus on attackers gaining higher privileges. It is important to remember that while going directly to administrative or root credentials is tempting, a roundabout attack that slowly gains greater access can have the same effect and may bypass controls that would stop an attack attempting to gain root access.

Horizontal escalation attacks move sideways to other accounts or services that have the same level of privileges. Gaining access to other accounts is often aimed at accessing the data or specific rights that the account has rather than targeting advanced privileges.

In addition to the targeting of the exploit, the exploit method used for privilege escalation is a useful distinction between escalation exploits. Common exploit targets include these:

■✓

■✓

■✓

■✓

Kernel exploits, which are one of the most commonly used local exploit methods for vertical escalation. *Many require local accounts and thus are less likely to be patched*

immediately by defenders who may focus on patching remote exploits and other critical vulnerabilities.

Application and service exploits may target accounts that the service runs as or under, or they may target business logic or controls in the application or service itself.

Database privilege escalation attacks may leverage SQL injection or other database software flaws to use elevated privilege or to query data from the database.

Design and configuration issues can also allow privilege escalation, making it worth a penetration tester's time to validate which controls are applied to accounts and if accounts have rights or privileges that they wouldn't be expected to have.

Many of the same techniques used by advanced persistent threat actors are useful for penetration testers, and vice versa. If your persistence techniques aren't monitored for and detected by your client's systems, your findings should include information that can help them design around this potential issue.

Social Engineering

Technical exploitation methods can be highly effective, but humans remain the most vulnerable part of any environment. That means penetration testers need to be ready to include social engineering in their test plan if it is allowed by the rules of engagement and included in the scope of work. The use of deception-based techniques that leverage human weaknesses can provide access that bypasses technical security layers that cannot otherwise be overcome.

Social engineering attacks against an organization may take a multitude of forms:

■✓

■✓

■✓

■✓

Phone, email, social media, and SMS phishing for credentials or access

On-site attacks like impersonation of vendors, staff, or other trusted individuals or organizations

Acquisition of information via dumpster diving

Distribution of USB thumb drives or other devices containing Trojans or other attack softwarePersistence and Evasion

Social engineering techniques can significantly improve the personnel-related information provided in a penetration test report, and penetration testers need to be aware of the potential advantages that the test brings. A social engineering test can provide information about employee behavior, policy compliance and enforcement, and security awareness in addition to the information and access that it may provide through an organization's security boundaries. Such tests can also be very challenging to do well, and they require a distinct skill set beyond technical penetration-testing capabilities.

Scenario Part 2

Now that you have gained access to the vulnerable system you identified and exploited at the start of this chapter, you next need to ensure that you can retain access and avoid

detection.

Answer the following questions and practice the techniques you identify against the Metasploitable 3 virtual machine; then log in as an administrator or using the vagrant user and verify that you do not see obvious signs of exploit in the service directory or elsewhere.

■■How can you create a persistent service?

■■What commands would you use to create the persistent service?

■■What Metasploit payload best supports this?

■■

■■

How can you best protect against detection by an antivirus tool like Windows Defender?

What other evasion and cleanup techniques would you use to help avoid detection?

Persistence and Evasion

The ability to compromise a host is important, but the ability to retain access to the system to continue to gather data and to conduct further attacks is even more critical to most penetration attacks. That means persistence is a critical part of a penetration tester's efforts.

Scheduled Jobs and Scheduled Tasks

One of the simplest ways to maintain access to a system is via a scheduled job or task using the techniques we reviewed earlier in this chapter. An advantage of a scheduled action is that it can allow recurring callbacks to a remote system rather than requiring a detectable service to be run. T*his is the same reason many botnets rely on outbound SSL-protected calls to remote web servers for their command and control. Using a secure protocol for the remote connection and ensuring that the system or systems to which the compromised host connects are not easily associated with the penetration tester's activities can help conceal the compromise.*

Inetd Modification

The Inetd super daemon and its relatives (Xinetd, Rlinetd) run a variety of services on Linux systems. Adding additional services to Inetd can allow you to maintain a persistent connection via a service that you control, and subtle Inetd changes like changing the binary that provides a service may be missed by defenders.

If the system you are attacking can easily be re-exploited, you don't have to worry much about persistence—just repeat the attack that got you access last time!

Daemons and Services

Installing a fake service or inserting malicious code into an existing service in memory via a tool like Meterpreter can allow ongoing access to a system. Installing a daemon or service will provide longer access than code injected into memory, which won't survive reboots, but injected code is typically harder to detect.

Back Doors and Trojans

Back doors and Trojans can also be used to provide persistence. While purpose-built back doors can be a powerful tool, they're also more likely to be detected by anti-malware tools. An alternate method of creating a back door is to replace an existing service with a vulner-

able version. Once a vulnerable version is in place, you can simply exploit it, often without the system owner noticing the change in the executable or version.
Remember that Trojans are defined as malware that is disguised as legitimate software. A back door is defined as a means of bypassing security controls and/or authentication.
A fi nal method that attackers can use is direct code modification for web applications, scripts, and other tools where the code is accessible on the system. Removing input validation from web applications, adding vulnerable code or remote access tools,Pivoting

or making similar changes can provide penetration testers with ongoing access or alternate access methods.
New Users
Creation of a new user account is a tried-and-true method for retaining access to a system.
In well-managed and monitored environments, adding an account is likely to be caught and result in an alarm, but in many environments creation of a local user account on a system may allow ongoing access to the system, device, or application.
Metasploit's Meterpreter *makes this very easy on a compromised Windows system if you have an account with administrative privileges. Simply executing net user [newusername] [password] /add and net localgroup administrators [newusername] /add will result in the creation of user accounts. Metasploit also includes payloads that are designed to add a UID 0 user (root level access) to Linux, but this type of action is also simple to do from the command line once you have a privileged account or sudo rights.* C**oncealing new user creation can be difficult, but carefully selecting the new user account's name to match the names of existing or common services or other users who have local accounts can help conceal both the use of the account and any actions the account takes.**
Security incident responders who are responding to a breach will commonly check for new user accounts by reviewing the Windows SAM or the Linux password file. Some pen-testers (and attackers) may attempt to conceal their presence by modifying these files to make evidence like the creation order or date of the new account less obvious. ############
Pivoting
Once you have obtained a foothold by compromising a system and ensuring that you will have continued access, you can leverage that system to obtain a new perspective on the target network or systems. Using a compromised system can provide a new path into a network or help you identify new targets that were not visible from the original scan viewpoint.

FIGURE 6.14 Pivoting

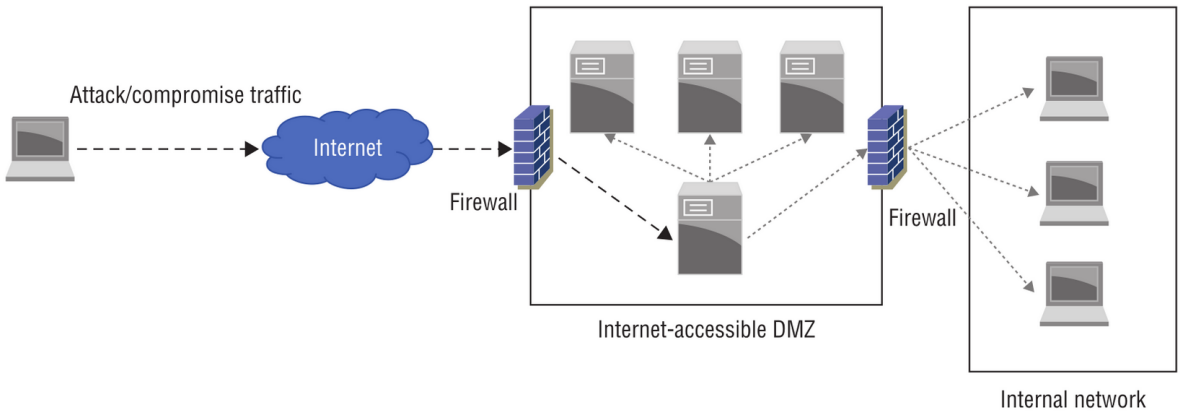


Figure 6.14 shows an attacker pivoting once they have breached a vulnerable system inside an Internet-accessible DMZ. The attacker may have discovered a vulnerable web service or another front-facing, exploitable vulnerability. Once they have compromised a server in the DMZ, they can scan systems that were not previously visible through the multiple layers of fi rewalls that the example organization has put into place. Note that in this case, both additional DMZ servers and workstations in the internal work are accessible. The same techniques discussed in prior chapters of this book would then be leveraged to conduct information gathering and pre-exploit activities.212

Pivoting can also occur on a single system when attackers pivot from one account or service to another. This change in approach or view is a critical part of a penetration tester's process, since very few organizations have all of their services and systems exposed to the outside world or to a single place that attackers can access. Understanding the organization's network and systems design, including internal and external defenses and services, can allow for more effective pivoting.

Covering Your Tracks

An important post-exploit task is cleaning up the tools, logs, and other traces that the exploit process may have left on the target machine. This can be very simple or quite complex, depending on the techniques that were used, the configuration and capabilities of the target system, and the tools that were needed to complete the attack.

One of the first steps you should consider when covering your tracks is how to make the tools, daemons, or Trojans that you will use for long-term access appear to be innocuous. **Some tools like Meterpreter do this by inserting themselves into existing processes, using names similar to common harmless processes or otherwise working to blend in with the normal behaviors and files found on the system.**

It can be difficult, if not impossible, to conceal all of the tools required to compromise and retain access to a system. In cases where it is possible that your tools may be discovered, encryption and encoding tools like packers, polymorphic tools that change code so that it cannot be easily detected as the same as other versions of the same attack tools, and similar techniques can help slow down defenders. The same techniques used by advanced persistent threats and major malware packages to avoid detection and prevent analysis can be useful to penetration testers because their goal is similar.

In addition to hiding the tools and other artifacts required to retain access, cleanup is

important. Penetration testers need to know where the files that their attacks and actions created will be and should ensure that those files have been removed. You also need to track the log files that may contain evidence of your actions. While it may be tempting to wipe the log files, empty log files are far more suspicious than modified log files in

cases. If the target organization uses a remote logging facility, you may not be able to effectively remove all log-based evidence, and the difference between local and remote logs can indicate compromise if staff at the target notice the changes. This means that most practitioners fi rst choose to modify logs or clean them if possible, and then use log wiping only if they don't have another option.

Concealing communications between the target system and a penetration tester's workstation, or between multiple compromised systems, is also a key part of covering your tracks. The same techniques used by advanced malware are useful here, too. A combination of encrypted communications, use of common protocols, and ensuring that outbound communication travels to otherwise innocuous hosts can help to prevent detection. A direct RDP session in from the penetration tester's workstation after performing a series of port and vulnerability scans is much more likely to be detected by a reasonably competent security team!

In a penetration test conducted against an organization with a strong security team, you may need to use more advanced techniques. While they're beyond the scope of the PenTest+ exam and this book, anti-analysis and anti-forensic tools like packers and other encoders, as well as other techniques and applications, may be useful. Advanced Penetration Testing: Hacking the World's Most Secured Networks by Will Allsopp (Wiley, 2017) is a good book to start with if you want to learn more.

Summary

Once a penetration tester has gathered vulnerability information about a target, the next step is to map those vulnerabilities to potential exploits. Vulnerability and exploit databases both allow penetration testers to match the vulnerabilities that they discover to exploits, while tools like Metasploit provide ratings for prebuilt exploit packages that allow testers to select the exploits that are most likely to succeed.

In addition to exploits, techniques like exploit chaining, which uses multiple steps to complete an exploit, are important for penetration testers to both understand and be able to use. Developing custom exploits can be challenging, but modifying or configuring exploits to fit the targets that you are facing can make the difference between a successful attack and a failed exploitation attempt.

In addition to technical exploitation techniques, penetration testers need to be aware of social engineering techniques like phishing, dumpster diving, in-person impersonation, and other deception methods. Targeting human weaknesses can bypass technical and administrative security controls that penetration testers may not otherwise be able to circumvent. A helpful staff member may provide you with the foothold you need!

Once you have successfully exploited one or more systems and have gained a toehold inside an organization, post-exploitation activities begin. A fi rst step is to attempt lateral

movement to other systems and devices that may only be accessible from inside the organization. Penetration testers should also consider additional information gathering and

vulnerability assessment from their new vantage point, since most systems and networks focus more on outside attackers than those inside of security boundaries due to the need for internal users to perform their jobs.

Post-exploitation activities also include cleanup, concealment, and retaining access for longer-term penetration testing activities. ***You should make sure you know how to hide the evidence of your actions by cleaning up log files, removing the files created by your tools, and ensuring that other artifacts are not easily discoverable by defenders. Techniques like encryption, secure communications, and building scripted callbacks are all important to concealing and retaining long-term access.***

Exam Essentials