Chapter 6

Exploit and Pivot

Outlines

- Exploits and Attacks
- The Exploit Database
- The Rapid7 Vulnerability and Exploit Database
- The National Vulnerability Database
- VULDB
- Developing Exploits
- Exploit Proof-of-Concept Development
- Exploit Modification
- Exploit Chaining
- Metasploit
- PowerSploit
- RPC/DCOM
- PsExec
- PS Remoting/WinRM
- WMI
- Scheduled Tasks and cron Jobs
- SMB
- RDP
- Apple Remote Desktop
- VNC
- X-Server Forwarding
- SSH
- Common Post-Exploit Attacks
- Privilege Escalation
- Social Engineering
- Scheduled Jobs and Scheduled Tasks
- Inetd Modification
- Daemons and Services
- Back Doors and Trojans
- New Users
- Pivoting
- Covering Your Tracks

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.

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.

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 for 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.

Developing Exploits

When 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.

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-setupand proof-of-concept/ includes a complete walk-through that shows how Rick Osgood identified, built, and tested a proof-of-concept exploit.

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 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.

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.

PowerSploit

PowerSploit is a set of Windows PowerShell scripts that are designed to provide capabilities including antivirus bypass, code execution, exfiltration, persistence, reverse engineering, and reconnaissance. Much like Metasploit, PowerSploit is a very powerful, flexible tool.

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.

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.

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.

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.

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.

RDP

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

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. SSH 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.

Common Post-Exploit Attacks

Password attack

- **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!

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.

- 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.

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.

- 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 software

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. This 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.

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 vulnerable 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.

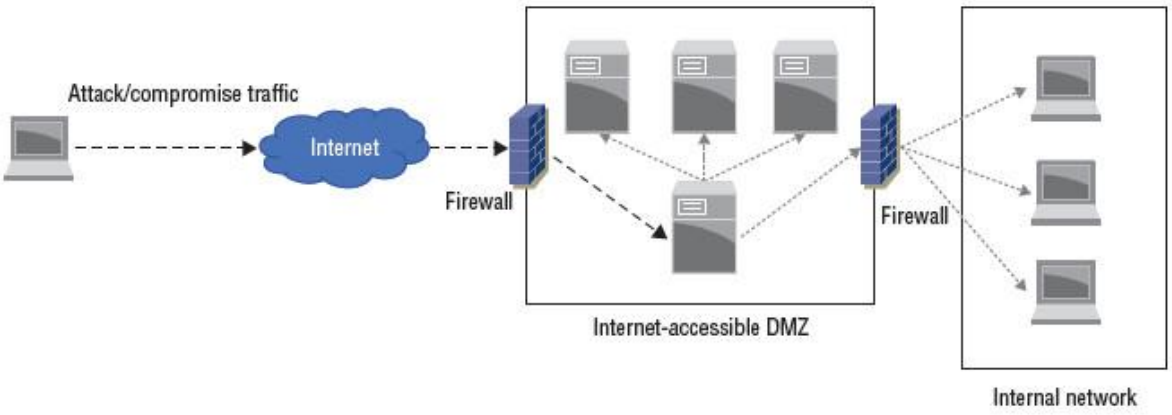
New Users

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.

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.

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 firewalls 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.



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.

Questions

1. Alice discovers a rating that her vulnerability scanner lists as 9.3 out of 10 on its severity scale. The service that is identified runs on TCP 445. What type of exploit is Alice most likely to use on this service?
   A. SQL injection
   B. SMB exploit
   C. CGI exploit
   D. MIB exploit

2. Which of the entries should Charles prioritize from this list if he wants to gain access to the system?
   A. the Ruby on Rails vulnerability
   B. the OpenSSH vulnerability
   C. the MySQL vulnerability
   D. None of these; he should find another target.

If Charles wants to build a list of additional system user accounts, which of the vulnerabilities is most likely to deliver that information?

A. the Ruby on Rails vulnerability

B. the OpenSSH vulnerability

C. the MySQL vulnerability

D. Both the OpenSSH and MySQL vulnerabilities

If Charles selects the Ruby on Rails vulnerability, which of the following methods cannot be used to search for an existing Metasploit vulnerability?

A. CVE

B. BID

C. MSF

D. EDB

Matt wants to pivot from a Linux host to other hosts in the network but is unable to install additional tools beyond those found on a typical Linux server. How can he leverage the system he is on to allow vulnerability scans of those remote hosts if they are firewalled against inbound connections and protected from direct access from his penetration testing workstation?

A. SSH tunneling

B. Netcat port forwarding

C. Enable IPv6

D. Modify browser plug-ins

After gaining access to a Windows system, Fred uses the following command: SchTasks /create /SC Weekly /TN "Antivirus" /TR C:\Users\SSmith\av.exe" /ST 09:00 What has he accomplished?

A. He has set up a weekly antivirus scan.

B. He has set up a job called “weekly.”

C. He has scheduled his own executable to run weekly.

D. Nothing; this command will only run on Linux.

After gaining access to a Linux system through a vulnerable service, Cassandra wants to list all of the user accounts on the system and their home directories. Which of the following locations will provide this list?

A. /etc/shadow

B. /etc/passwd

C. /var/usr

D. /home

A few days after exploiting a target with the Metasploit Meterpreter payload, Robert loses access to the remote host. A vulnerability scan shows that the vulnerability that he used to exploit the system originally is still open. What has most likely happened?

A. A malware scan discovered Meterpreter and removed it.

B. The system was patched.

C. The system was rebooted.

D. Meterpreter crashed.

Angela wants to run John the Ripper against a hashed password file she has acquired from a compromise. What information does she need to know to successfully crack the file?

A. A sample word list

B. The hash used

C. The number of passwords

D. None of the above

Chris cross compiles code for his exploit and then deploys it. Why would he cross-compile code?

A. To make it run on multiple platforms

B. To add additional libraries

C. To run it on a different architecture

D. To allow him to inspect the source code

Lauren has acquired a list of valid user accounts but does not have passwords for them. If she has not found any vulnerabilities but believes that the organization she is targeting has poor password practices, what type of attack can she use to try to gain access to a target system where those usernames are likely valid?

A. Rainbow tables

B. Dictionary attacks

C. Thesaurus attacks

D. Meterpreter

What built-in Windows server administration tool can allow command-line PowerShell access from other systems?

A. VNC

B. PowerSSHell

C. PSRemote

D. RDP

John wants to retain access to a Linux system. Which of the following is not a common method of maintaining persistence on Linux servers?

A. Scheduled tasks

B. Cron jobs

C. Trojaned services

D. Modified daemons

Tim has selected his Metasploit exploit and set his payload as cmd/unix/generic. After attempting the exploit, he receives the following output. What went wrong?

```
msf exploit(unix/misc/distcc_exec) > exploit

[-] Exploit failed: The following options failed to validate: RHOST.
[*] Exploit completed, but no session was created.
```

A. The remote host is firewalled.

B. The remote host is not online.

C. The host is not routable.

D. The remote host was not set.

Cameron runs the following command via an administrative shell on a Windows system he has compromised. What has he accomplished?

$command = 'cmd /c powershell.exe -c Set-WSManQuickConfig

-Force;Set-Item WSMan:\localhost\Service\Auth\Basic -Value $True;Set-Item

WSMan:\localhost\Service\AllowUnencrypted

-Value $True;Register-PSSessionConfiguration -Name Microsoft.PowerShell

-Force'

A. He has enabled PowerShell for local users.

B. He has set up PSRemoting.

C. He has disabled remote command-line access.

D. He has set up WSMan.

Mike discovers a number of information exposure vulnerabilities while preparing for the exploit phase of a penetration test. If he has not been able to identify user or service information beyond vulnerability details, what priority should he place on exploiting them?

A. High priority; exploit early.

B. Medium priority; exploit after other system and service exploits have been attempted.

C. Low priority; only exploit if time permits.

D. Do not exploit; information exposure exploits are not worth conducting.

Part of Annie's penetration testing scope of work and rules of engagement allows her physical access to the facility she is testing. If she cannot find a remotely exploitable service, which of the following social engineering methods is most likely to result in remote access?

A. Dumpster diving

B. Phishing

C. A thumb drive drop

D. Impersonation on a help desk call

Jacob wants to capture user hashes on a Windows network. Which tool could he select to gather these from broadcast messages?

A. Metasploit

B. Responder

C. Impacket

D. Wireshark

Cynthia wants to find a Metasploit framework exploit that will not crash the remote service she is targeting. What ranking must the exploit she chooses meet or exceed to ensure this?

A. Excellent

B. Great

C. Good

D. Normal

Alex wants to use rainbow tables against a password file she has captured. How do rainbow tables crack passwords?

A. Un-hashing the passwords

B. Comparing hashes to identify known values

C. Decrypting the passwords

D. Brute-force testing of hashes