We first tried to see what else we could do with sqlmap and saw that it was possible to spawn a shell with the –os-shell flag and when trying it with the command below we managed to get a shell and could access the underlying linux machine:

```
sqlmap -u http://10.162.2.83/php/recherche_old.php --forms --os-shell
--cookie="PHPSESSID=n7hkf6r0kp9916ad2u73q9t0c2" --batch
```

When inside with sqlmap os-shell it also setup a file stager during exploitation where you could upload files through a php file uploader. For this we tried to upload some shells and some files but most didn't work. The we tried to upload LinEnum.sh and ran a report on the system with:

```
chmod +x LinEnum.sh (needs execute permissions after uploading it)
./LinEnum.sh -r report -e ./ -t
```

When scrolling through this we saw that we as www-data had read and write permission to an hourly cron job called cuitieur_cleaning which removed some files in the website php folder. We saw in the /etc/crontab that the cron.hourly ran as root periodically so what we did was the command:

```
echo 'ls /root  > /var/www/html/php/cronjob_clean_read.txt' >>
/etc/cron.hourly/cuiteur-cleaning
```

What this does is it tries to list files in the root folder, and since the cronjob runs as root it will have root permissions and therefore is able to read in the root folder.
So then some time later we had the text file cronjob_clean_read.txt in php folder and saw that it had printed the contents of the root folder which contained the flag:
flag{9f1f16671871f0020f7d4887d88d655591150d5f60ff9a}.jpg

After this we can also do other root stuff through the cronjob, like e.g viewing the /etc/shadow file and get the password for the user with the name eh.