**Soccer walkthrough**

# **Index**

[Index 1](#_Toc194428388)

[List of pictures 1](#_Toc194428389)

[Disclaimer 2](#_Toc194428390)

[Reconnaissance 2](#_Toc194428391)

[Initial foothold 3](#_Toc194428392)

[User flag 4](#_Toc194428393)

[Privilege escalation 7](#_Toc194428394)

[Personal comments 8](#_Toc194428395)

[References 8](#_Toc194428396)

# **List of pictures**

[Figure 1 - nMap scan results (part 1) 2](#_Toc194428372)

[Figure 2 - nMap scan results (part 2) 3](#_Toc194428373)

[Figure 3 - New path found 3](#_Toc194428374)

[Figure 4 - File manager service "Tiny" 4](#_Toc194428375)

[Figure 5 - Access to the Tiny service 4](#_Toc194428376)

[Figure 6 - Service user shell 4](#_Toc194428377)

[Figure 7 - New virtual host found 5](#_Toc194428378)

[Figure 8 - End point invocked via web socket 5](#_Toc194428379)

[Figure 9 - SQLMap execution 6](#_Toc194428380)

[Figure 10 - SQLMap results 6](#_Toc194428381)

[Figure 11 - Database identified 6](#_Toc194428382)

[Figure 12 - Table identified 7](#_Toc194428383)

[Figure 13 - Credentials found 7](#_Toc194428384)

[Figure 14 - Doas configuration 8](#_Toc194428385)

[Figure 15 - Privilege escalation 8](#_Toc194428386)

[Figure 16 - Root flag 8](#_Toc194428387)

# **Disclaimer**

I do this box to learn things and challenge myself. I’m not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who are willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Just to say: I am not an English native person, so sorry if I did some grammatical and syntax mistakes.

# **Reconnaissance**

The results of an initial nMap scan are the following:



Figure 1 - nMap scan results (part 1)

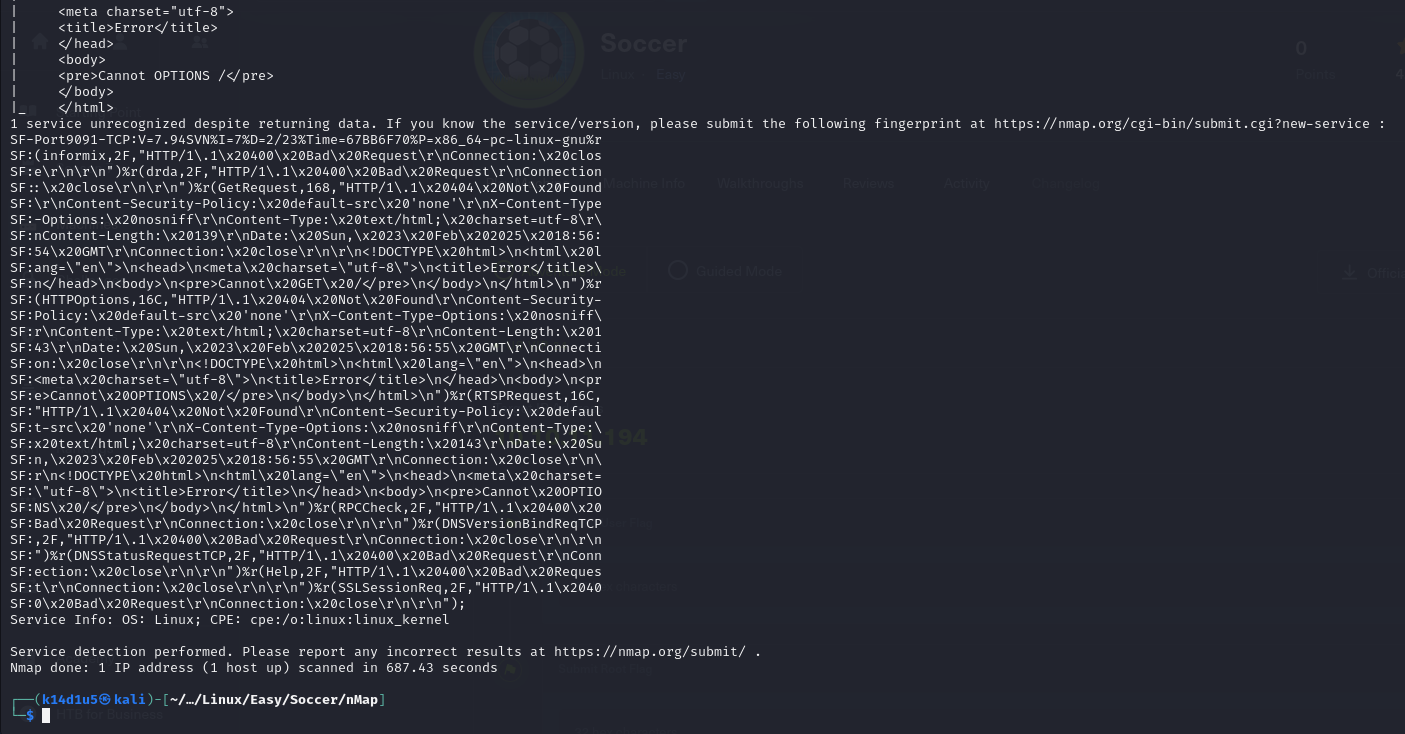


Figure 2 - nMap scan results (part 2)

Open ports are 22, 80, 9091, so enabled service is SSH. Also, there is a web application running on port 80. Honestly, I was not sure about which service is running on port 9091. Lastly, nMap recognized Linux as operative system.

# **Initial foothold**

Since I recognized a web application running on port 80, I started to analyzed it. To do so, I needed to add the relative entry in the file. Ne of the test I did was about founding new web content on the web application. As usual, I run tool to accomplish this task and, luckily, I found a new path:

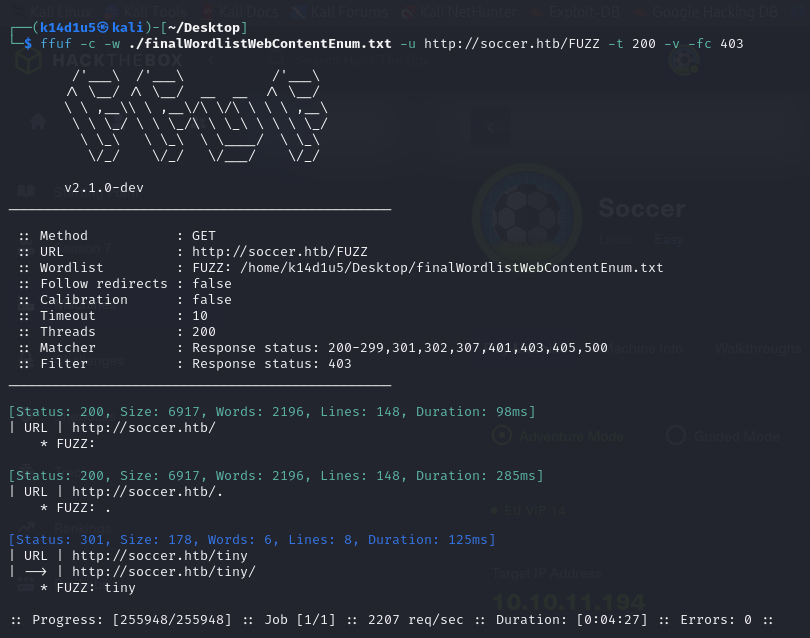


Figure 3 - New path found

At this point, I browsed to this path and I found out a file manager service, as shown in the following figure:

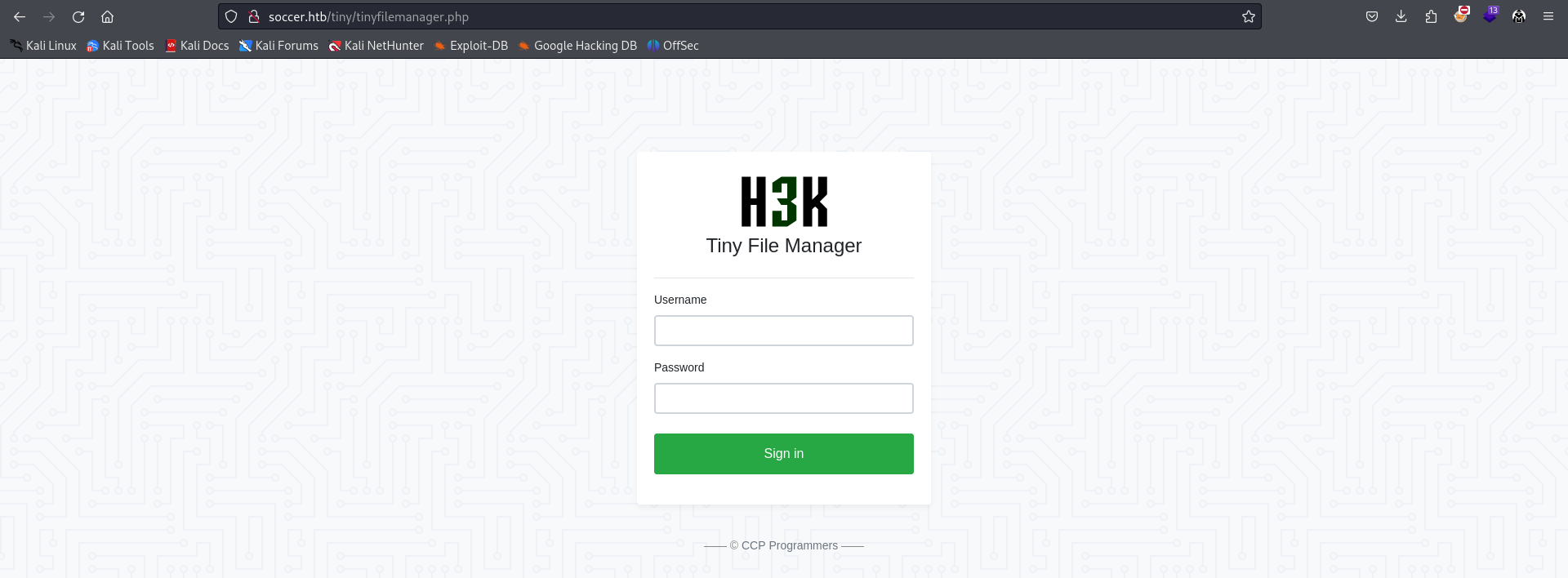


Figure 4 - File manager service "Tiny"

# **User flag**

I searched more information about this file manager service on the Internet. In this way I found out some default credentials. So, I tried it on the application and luckily them worked:

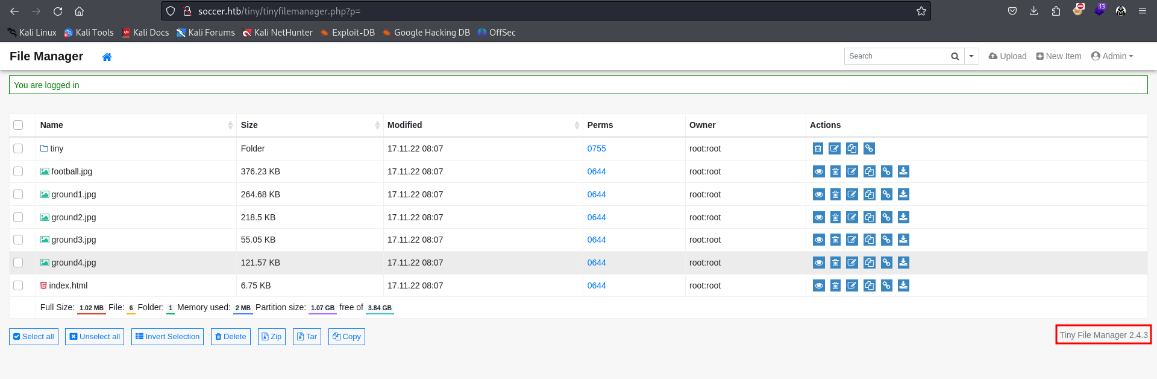


Figure 5 - Access to the Tiny service

After log in the application, I found out the version as well. I tried to upload some file and, after some tries, I understood that I can write only in the path. So, I uploaded a PHP web shell and, using it, I obtained a shell as user:



Figure 6 - Service user shell

Since I obtained this shell as a service user, I needed to do lateral movement to became a user. I navigated the file system. I analyzed nginx files and I found new virtual host as shown in the following figure:

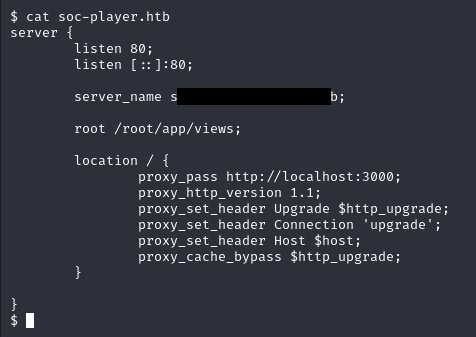


Figure 7 - New virtual host found

Again, I needed to add a new entry in the . In the previous screenshot I shown that this application is running in localhost on port 3000. To access to it, I needed to implement a port forwarding using Chisel tool. On this subdomain, I was able to create an account and log in. The account I created is valid until the log out. Analyzing this new web application, in particular the page source code, I found a specific endpoint invoked via web socket:

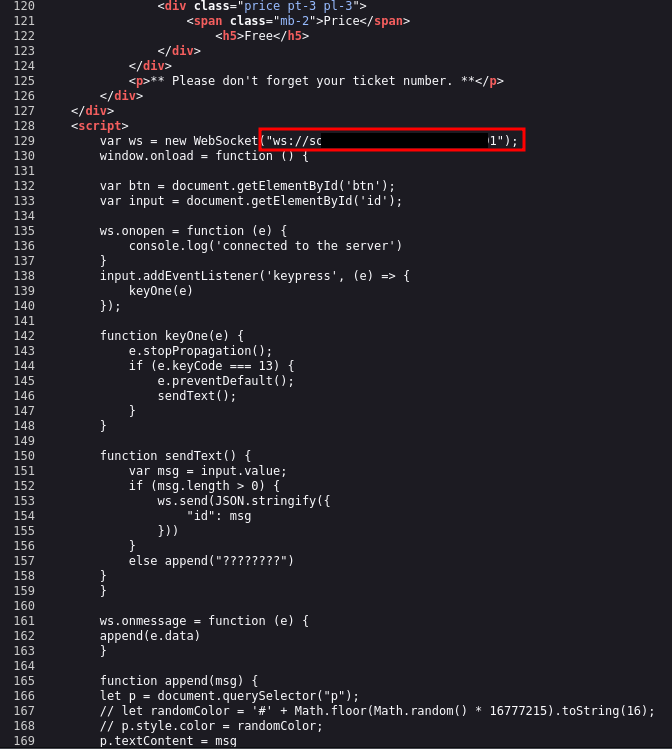


Figure 8 - End point invoked via web socket

I was able to analyze this endpoint using Burp Suite and it looked like it could be vulnerable to SQL Injection. Since I had this suspect, I tried to run SQLMap:

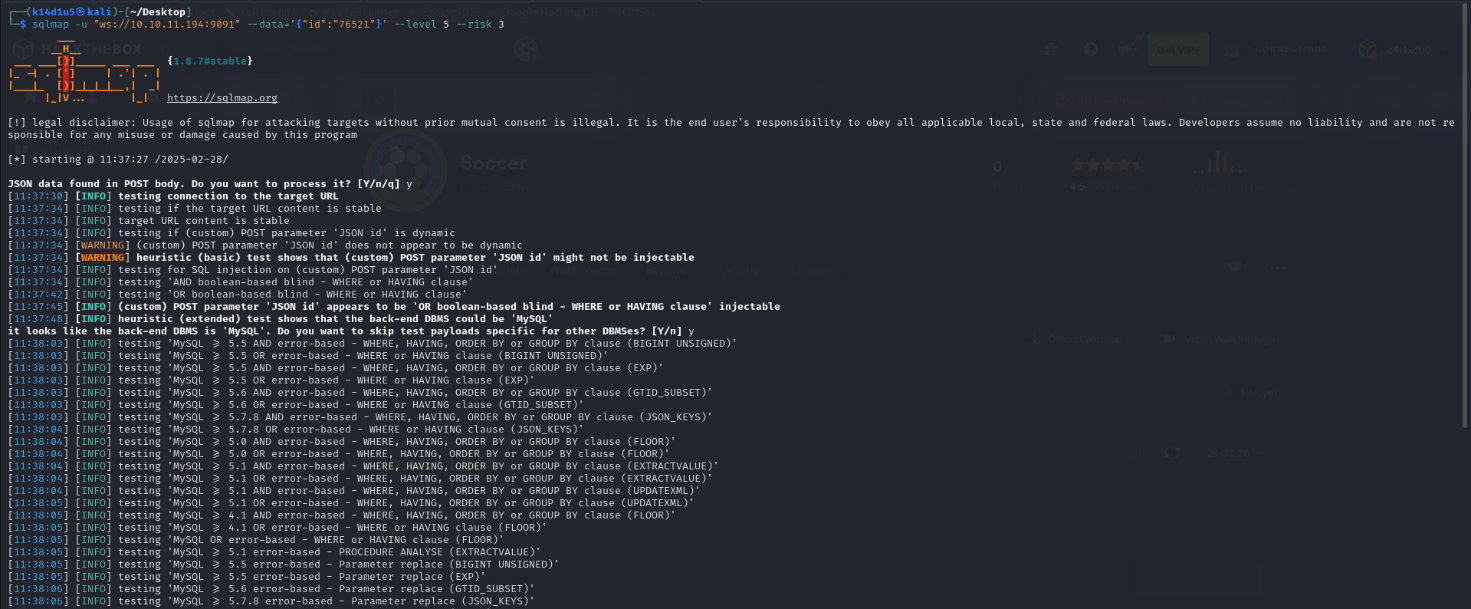


Figure 9 - SQLMap execution

As I suspected, this endpoint was vulnerable to SQL Injection:

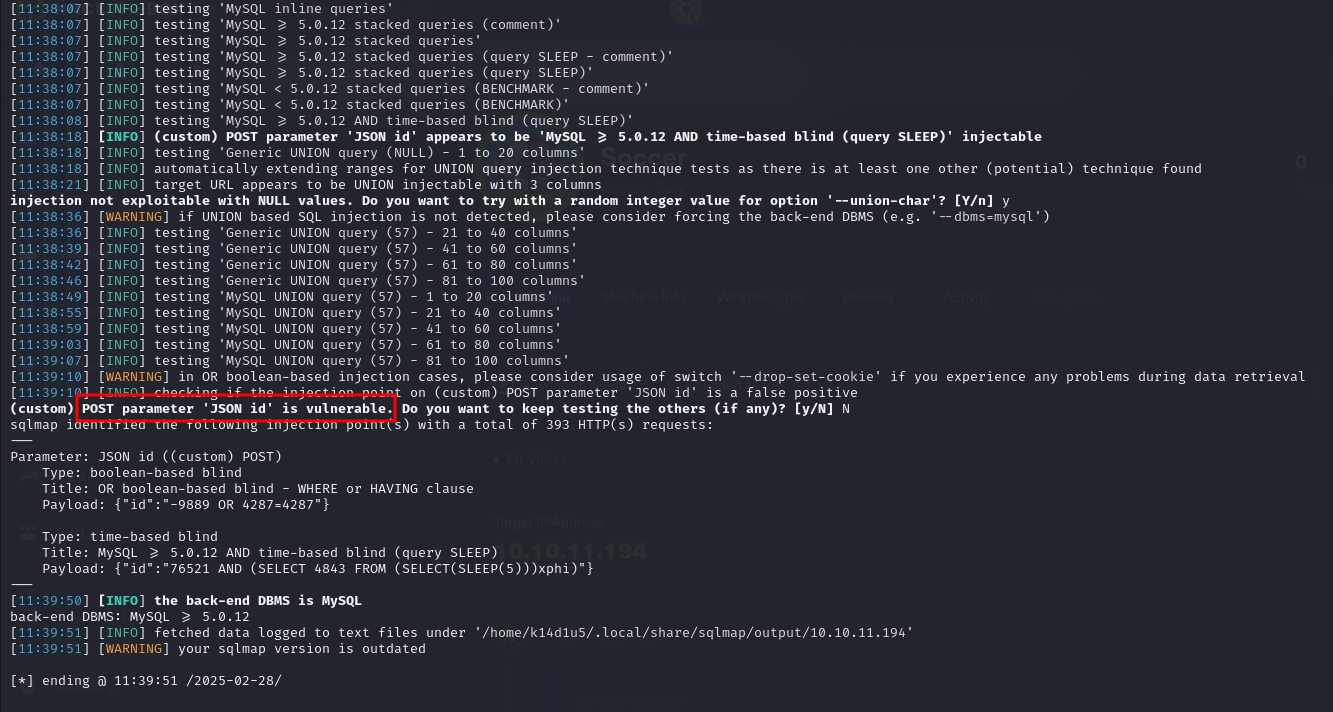


Figure 10 - SQLMap results

So, I was able to read data on the database, following the steps described in the nets figures:

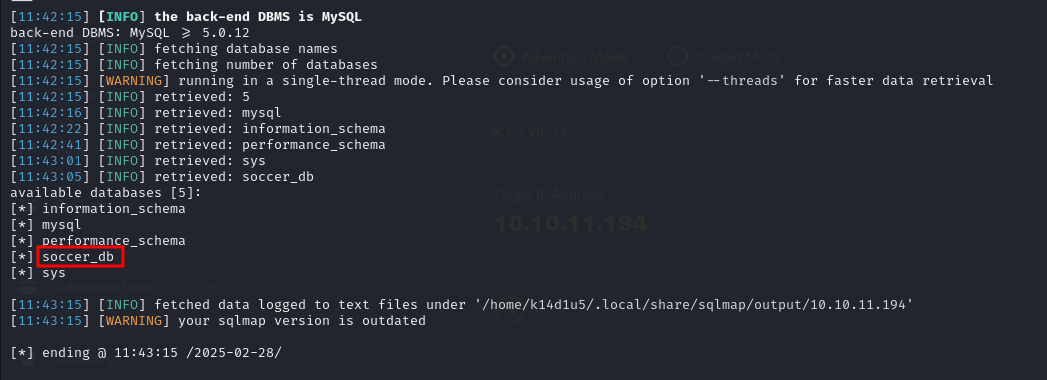


Figure - Database identified

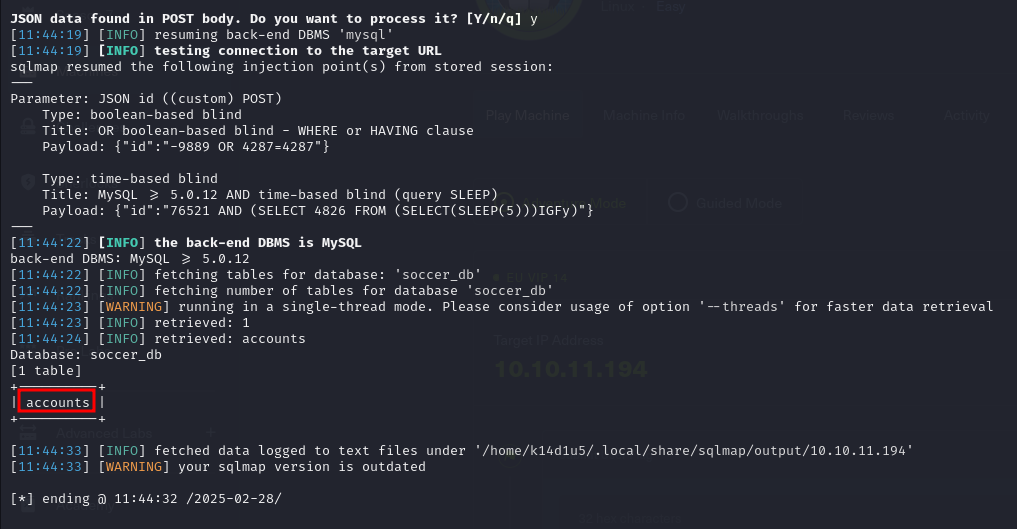


Figure - Table identified

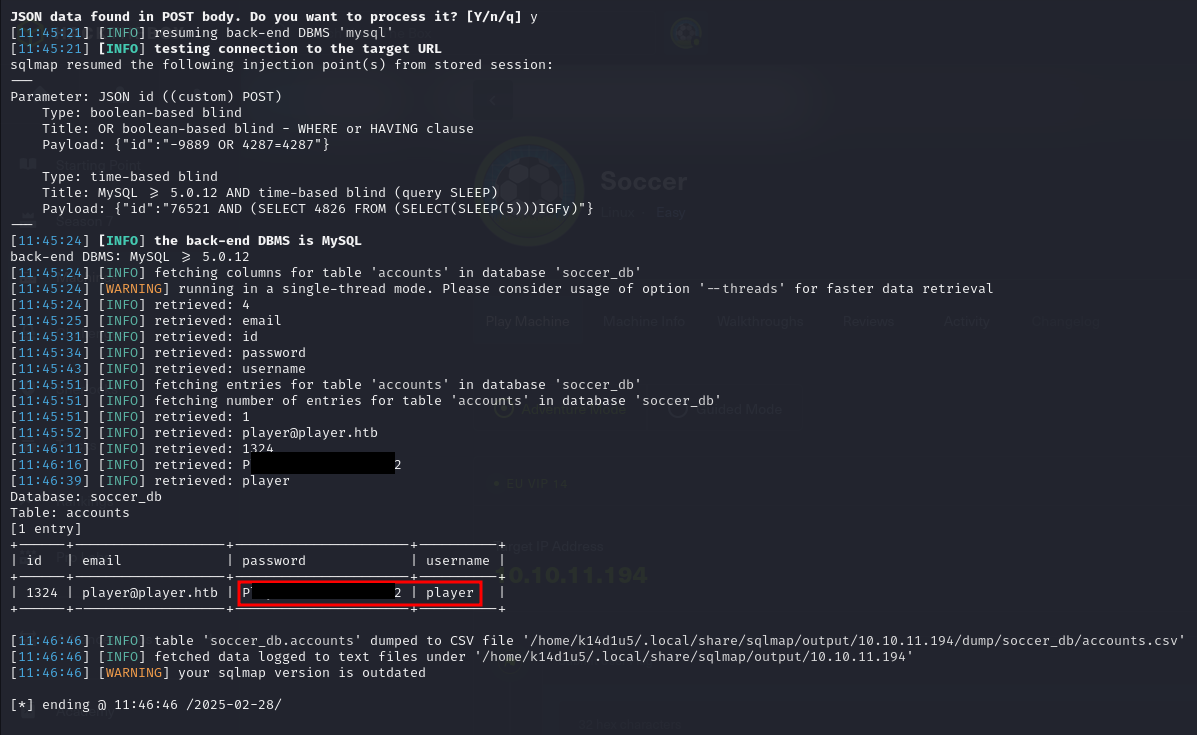


Figure 13 - Credentials found

Finally, using these credentials, I was able to connect to the target via SSH with the user and I was able to retrieved the user flag (I forgot the screenshot about the user flag).

# **Privilege escalation**

At this point I needed to find a path to escalate my privileges. Analyzing the filesystem, I found out that my user was able to execute as root the command, via binary:

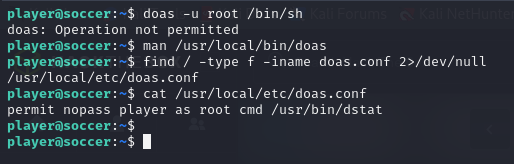


Figure 14 - Doas configuration

Looking for a possible exploit for command on the Internet, I found one very interesting:

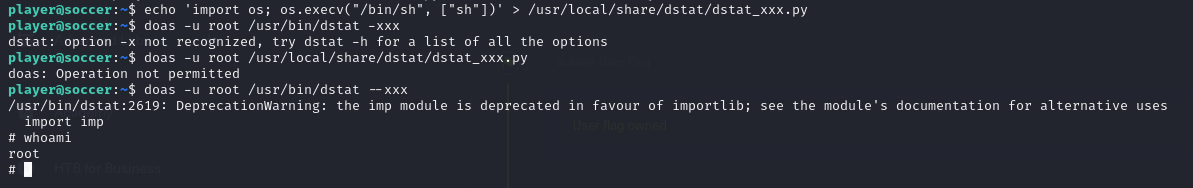


Figure 15 - Privilege escalation

At this point, I just needed to retrieve the root flag:

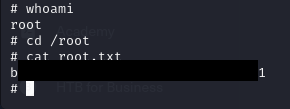


Figure 16 - Root flag

# **Personal comments**

This box was very interesting, but definitely not easy in my opinion. To retrieve the user flag, I needed to implement a port forwarding and an SQL Injection on a web socket. Also, the privilege escalation is achieved analyzing the configuration file and exploiting the command. So, I needed to chain more concept together to understand how to complete the box. I evaluate it as Medium on the HackTheBox platform.

# **References**

1. Chisel: <https://github.com/jpillora/chisel/releases>;
2. Web socket hacking: <https://hacktips.it/websocket-penetration-test/>;
3. Dstat exploiting: <https://gtfobins.github.io/gtfobins/dstat/>.