**Busqueda walkthrough**

# **Index**

[Index 1](#_Toc194439169)

[List of pictures 1](#_Toc194439170)

[Disclaimer 2](#_Toc194439171)

[Reconnaissance 2](#_Toc194439172)

[Initial foothold 2](#_Toc194439173)

[User flag 2](#_Toc194439174)

[Privilege escalation 3](#_Toc194439175)

[Personal comments 4](#_Toc194439176)

# **List of pictures**

[Figure 1 - nMap scan results 2](#_Toc194439161)

[Figure 2 - Web application exploit 2](#_Toc194439162)

[Figure 3 - User flag 3](#_Toc194439163)

[Figure 4 - Credentials and domain found 3](#_Toc194439164)

[Figure 5 - Sudoer permission 3](#_Toc194439165)

[Figure 6 - Database details 3](#_Toc194439166)

[Figure 7 - Privilege escalation exploit 4](#_Toc194439167)

[Figure 8 - Root flag 4](#_Toc194439168)

# **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:

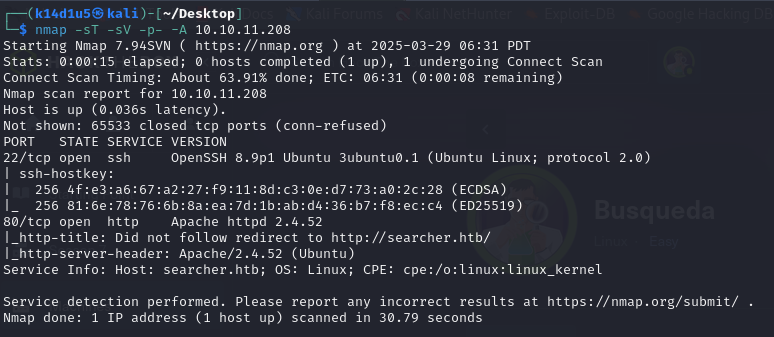


Figure 1 - nMap scan results

Open ports are 22 and 80. So, SSH service was enabled on this box and I found a web application running on port 80. Also, nMap recognized Linux as operative system.

# **Initial foothold**

I had very few options to explore this box. I started to analyze the web application. When I browsed to it in the browser, I found out that the application was developed by Flask and Searchor 2.4.0.

# **User flag**

As usual, I searched for some web hidden content on the web application, but I didn’t find anything. Since I learnt about the technologies used to develop the web application, I looked for some interesting exploit against these technologies on the Internet. Luckily, I found an interesting exploit against Searchor 2.4.0:

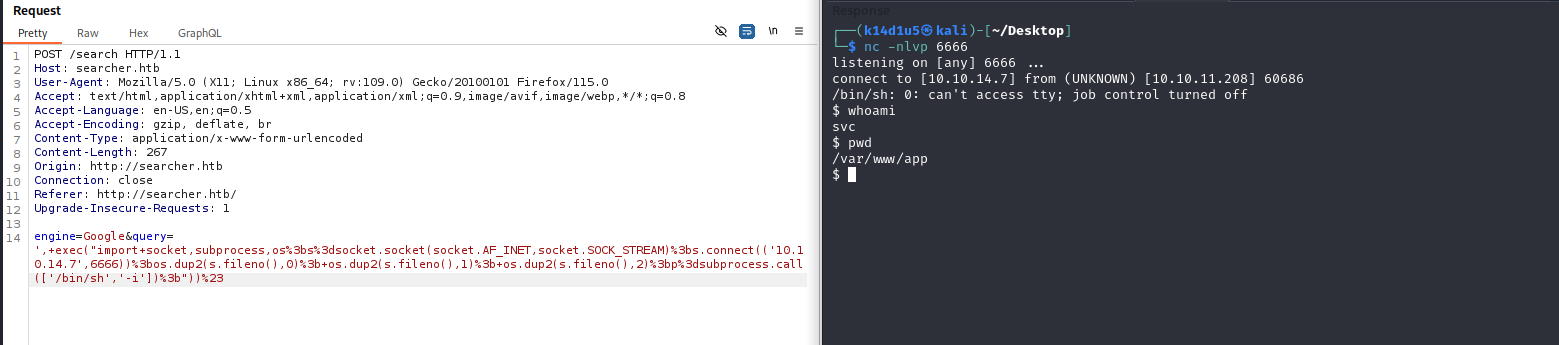


Figure 2 - Web application exploit

At this point, I was able to retrieve the user flag yet:



Figure 3 - User flag

# **Privilege escalation**

Now, I needed to escalate my privileges. I looked for some interesting information on the file system. After a bit of effort, I started to analyze the git instance I had. So, I read the git configuration file:

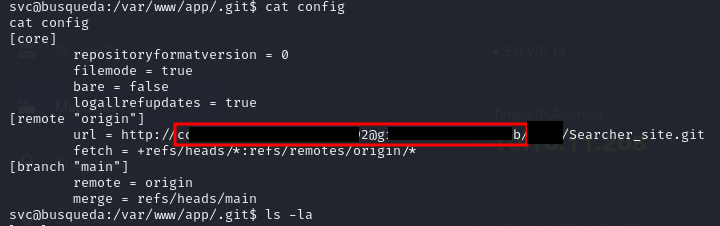


Figure 4 - Credentials and domain found

I was very surprised I found credential and a new subdomain in it. These credentials are useful to be logged in the domain I found, but it was not very helpful. Also, I tried to use them to log via SSH to the target. In this way, I obtained a more interactive and stable shell. In this shell, I checked the sudoers:

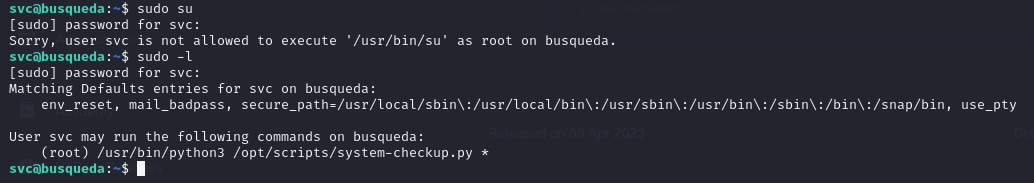


Figure 5 - Sudoer permission

I found out that my user is able to run with sudo a specific python script. Even if I wasn’t able to read the code, I studied this script behavior and, looking for more information about the action I was able to pass to it, I was able to retrieve a lot of information about the docker container. In particular, as shown in the following picture, I found information about the database and the login to it:

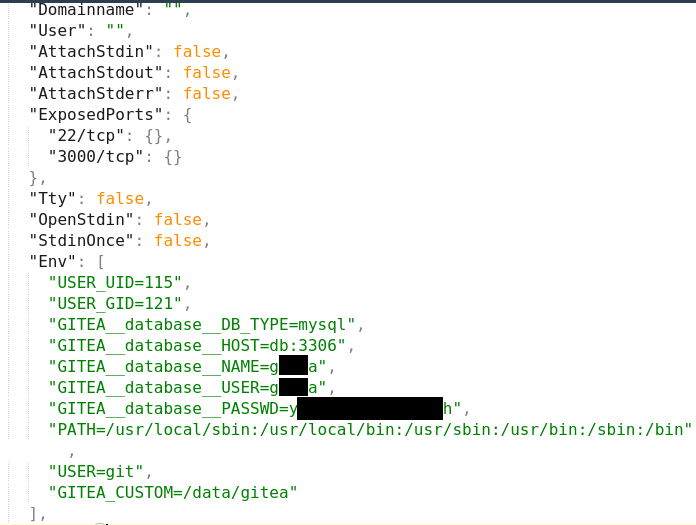


Figure 6 - Database details

At this point I investigated the database and I luckily found credentials. However, I was not able to decode them. After a bit of time, I tried to understand what I could do, I thought about the database credentials could be useful to log in in the new domain I found in the git configuration file. That worked and I obtained the login to it as administrator. The scripts I found in there was exactly the scripts I was able to run as sudo with my non privileged user. Luckily, I was able to read and analyze it that scripts. Since the script has a relative path to invoke a different script, I just needed to create that script in a malicious way:

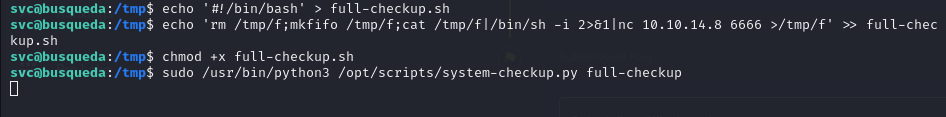


Figure 7 - Privilege escalation exploit

Finally, I was able to retrieve the root flag:

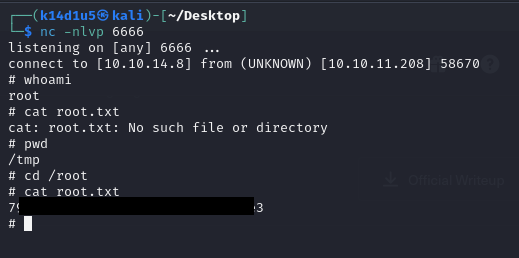


Figure 8 - Root flag

# **Personal comments**

I consider this box very interesting. I really liked it. The user flag was easy to retrieve. On the other hand, the root flag was a little bit challenging. In fact, was really important to consider git as an attack vector, something where you can find very interesting information. In my opinion, this is not very predictable. For these reasons, I evaluate “Easy” the user flag and “Not too easy” the root flag.