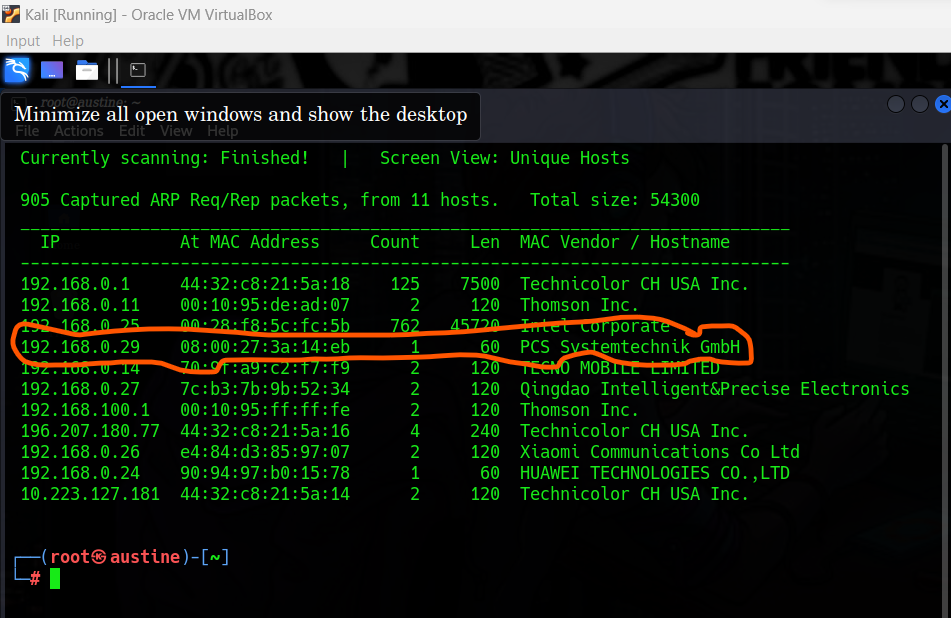
**EMPIRE: LUPINONE VULNHUB**

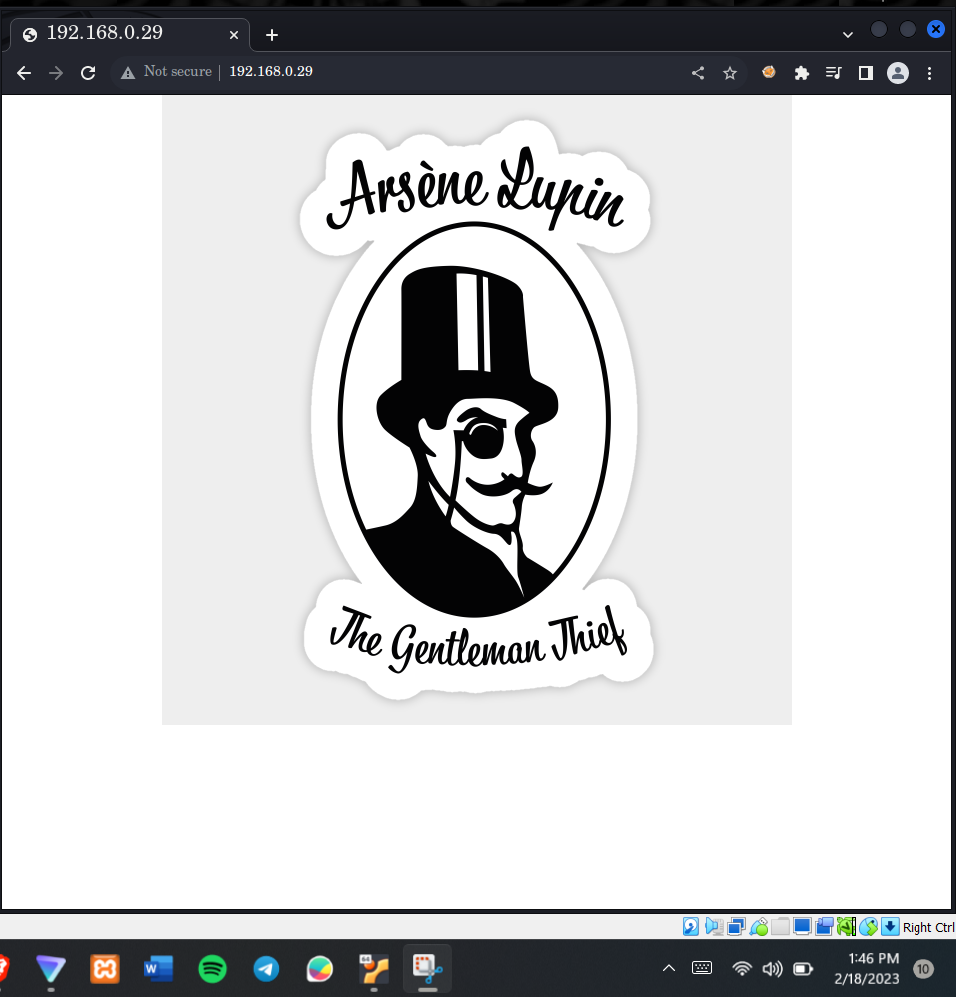
In this presentation, I will provide a breakdown of Empire: Lupin one vulnerability box.

Disclaimer: In order to keep this tutorial as short as possible I will go straight to the point where I found a breakthrough. It’s recommended that you ensure you explore the machine on your worn.

**Step 1: Using net discover to find the IP of the target machine. For me it’s 192.168.0.29**



**Step 2: Visiting the URL of the target machine where we find the homepage.**



**Step 3: Using the ffuf tool to search for directories found in the server.**

Since this machine does not have a login page, or any exploitable database, we use the ffuf tool to discover the possible directories found in the machine.

Command: **ffuf -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u** [**http://192.168.0.29/FUZZ.txt**](http://192.168.0.29/FUZZ.txt)

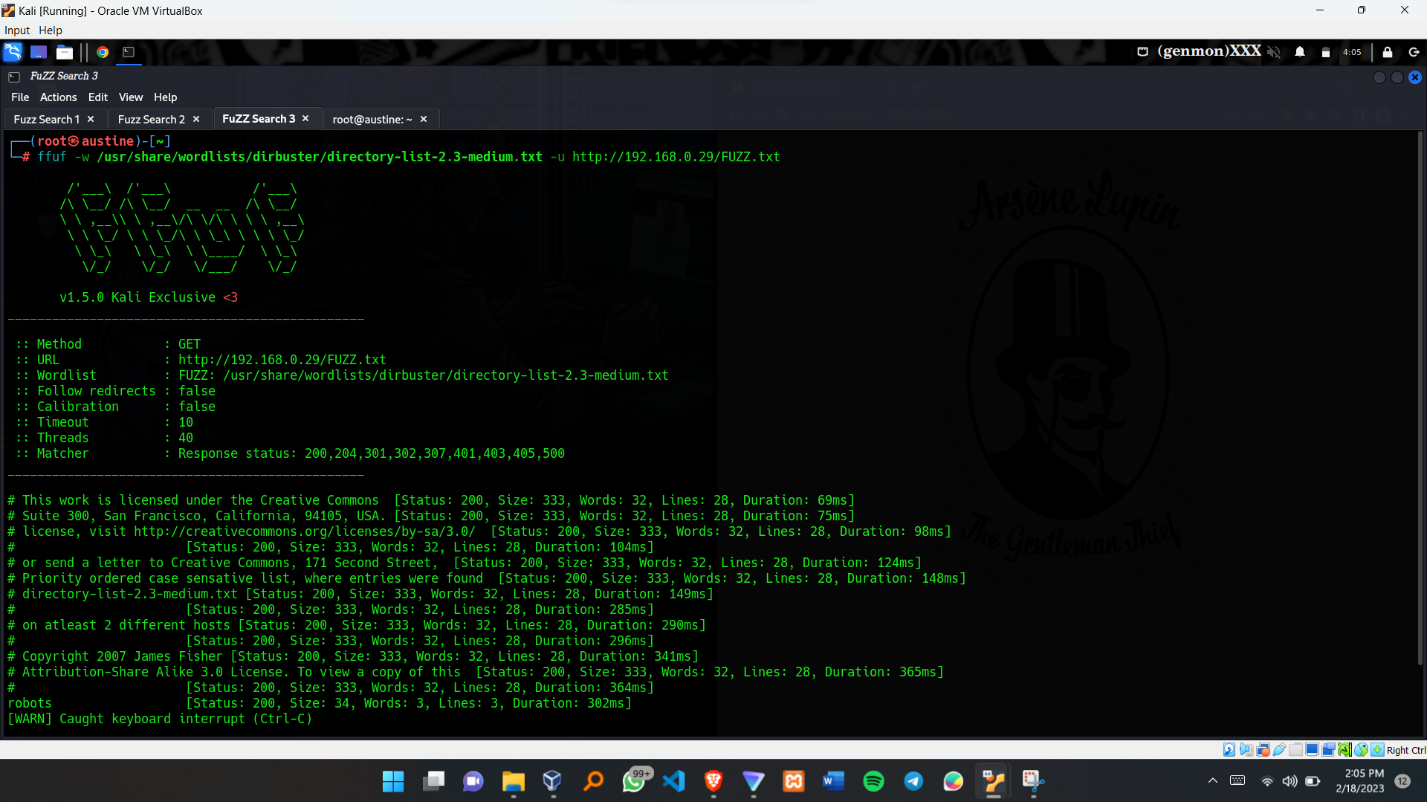
**Code explanation,**

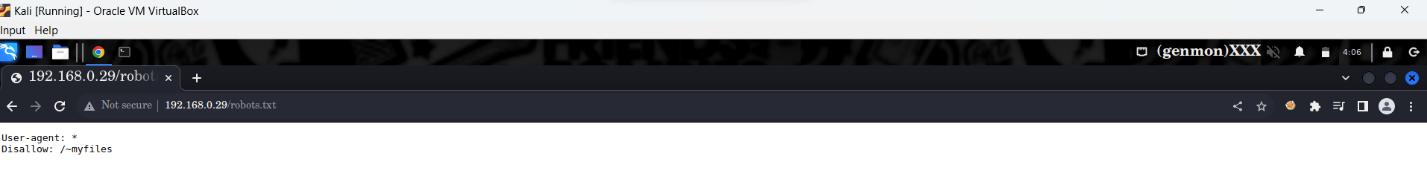
-w Dictates that the next parameter to be the wordlist that should be used.

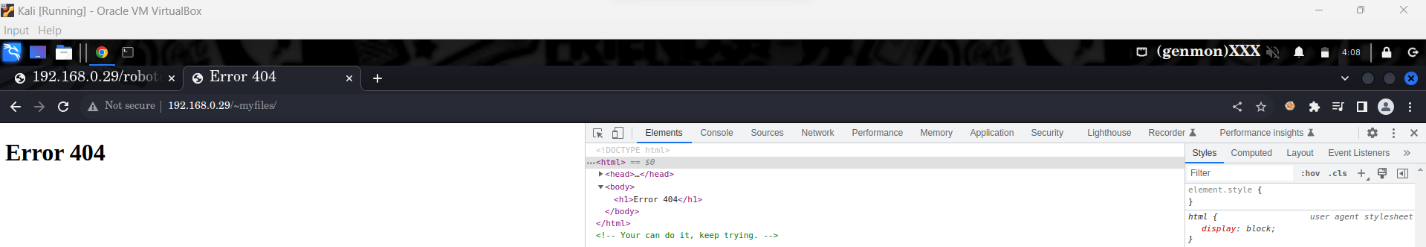
-u Expected the next parameter to be the URL

The FUZZ.txt is strategically placed where you would like to discover the directories.

However, you can simply use FUZZ without any extension if you believe that the directory may not have the .txt extension.



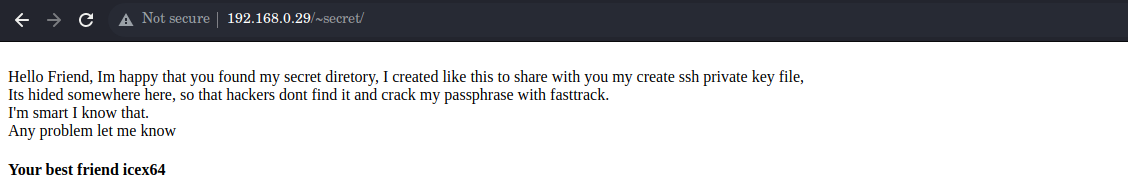
We found the robots.txt from the search which we shall open 

It provides us with ~myfiles pathway which we shall follow below. 

This pathway does not provide us with a progressive move but at least it gives us a motivational text “You can do it keep trying ”if we inspect the page.

**Step 4: Found another directory using the ffuf tool**



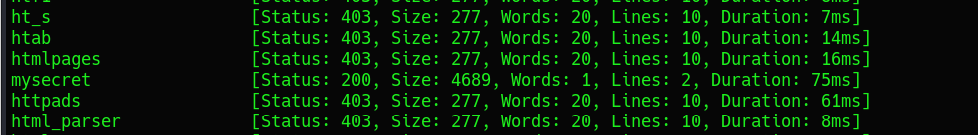
Navigating to the secret path. Now we find some reliable information. 

This message prepares us with the following important information

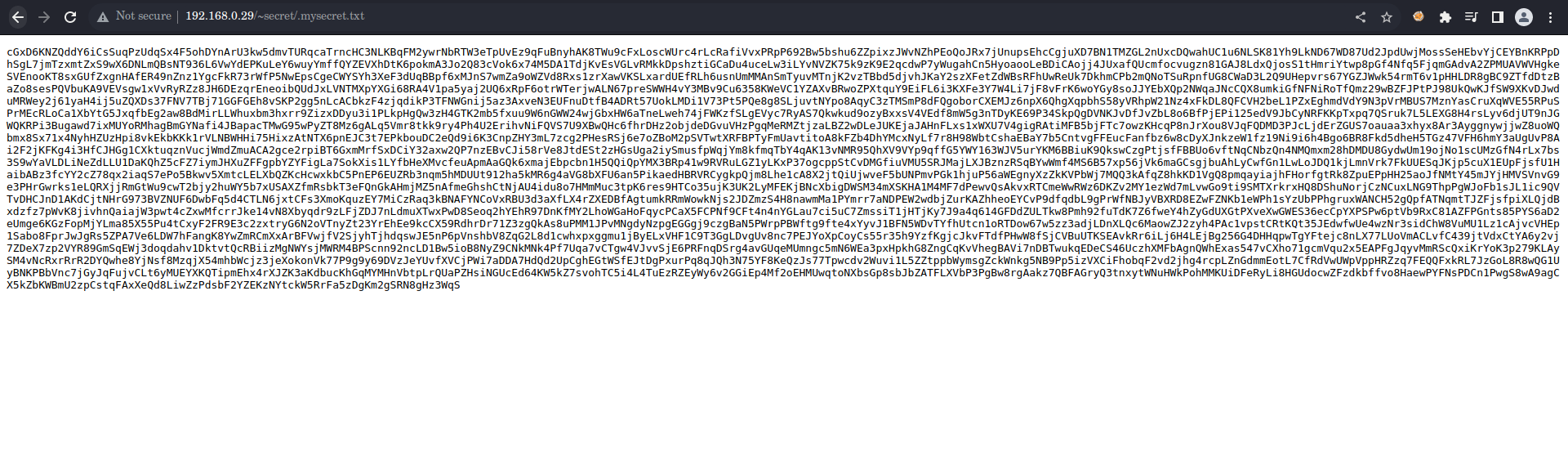
1. We are looking for ssh private key file
2. The ssh private key is found within this folder but is hidden ( it will have a dot before the name of the file)
3. The use of fasttrack would be applicable while cracking the passwords.
4. Best friend, which we can interpret as our ssh user is icex64

**Step 5: Still using the fuff tool we found the mysecret directory.**

Command : **ffuf -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u** [**http://192.168.0.29/~secret/.FUZZ.txt**](http://192.168.0.29/~secret/.FUZZ.txt)



Opening the directory, we find the encrypted message that contains the ssh private key.

We have to decrypt this message so that we are able to find the ssh private key needed.

Thing to consider before decrypting any encrypted message.

1. You have to know the encoding format that has been used.
2. The tools you will use to decrypt the message once you know the format used.

Step 6: Open up the browser and search for the convenient cipher decrypting tool.

In our case we shall use: **Cipher Identifier - Decrypt a Message**

It’s found in google and its URL is <https://www.dcode.fr/cipher-identifier>

Paste the encrypted message and once it has been analyzed, the results will point out that the most probable format used is base 58.



**Step 7: Now decrypt the message using a Base 58 decrypting tool.**

The results will clearly display the ssh private key. See the results below.

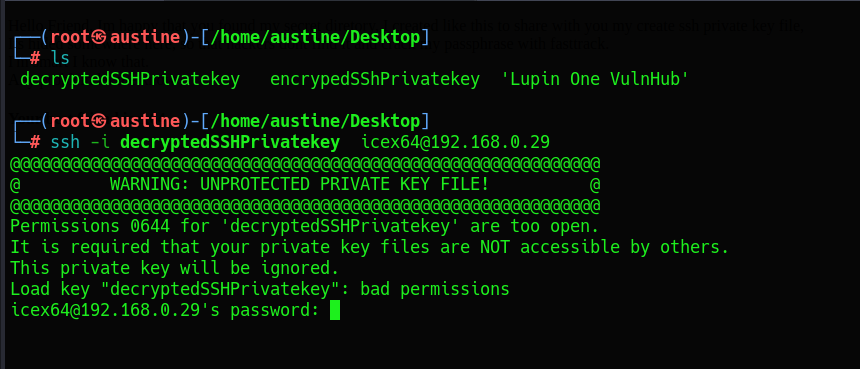
Copy and paste the results into your preferred working directory.



**Step 8: Try ssh login into the target**

You will encounter this error in case you try to login in using the ssh private key without changing its mode.

You are required to change its mode such that other users can’t access it since it’s a sensitive piece of data. In some cases, you may also be required to change the ssh private key into a hidden file within the directory it resides.



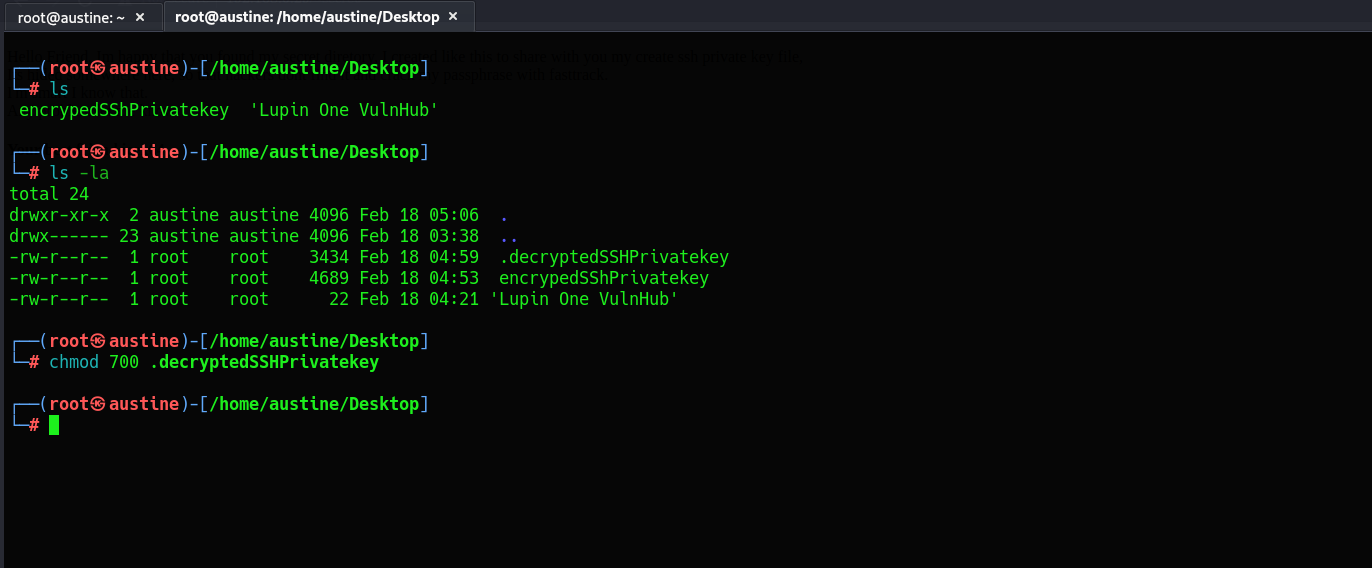
**Step 9: Changing mode and making the ssh private key Hidden**

To make the file hidden, just ensure that you use the syntax (.nameoftthefile ).

Notice the dot before the name.

Change mode command: **chmod 700 .nameoftheprivate key**

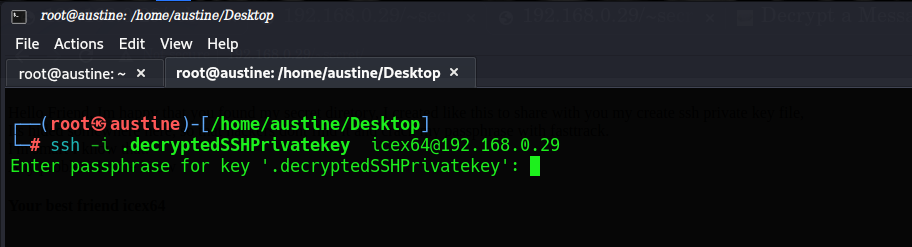
The 700 mode ensure that only you have the write to read, write and execute the file will other lack any access to the file.

.

**Step 10: Try to login again using the private ssh key.**

We are being asked for a passphrase which we don’t know yet.

During ssh private key generation, the user is usually asked to make type out their passphrase (optional). However, if we try to leave the passphrase blank, we get an error login in.

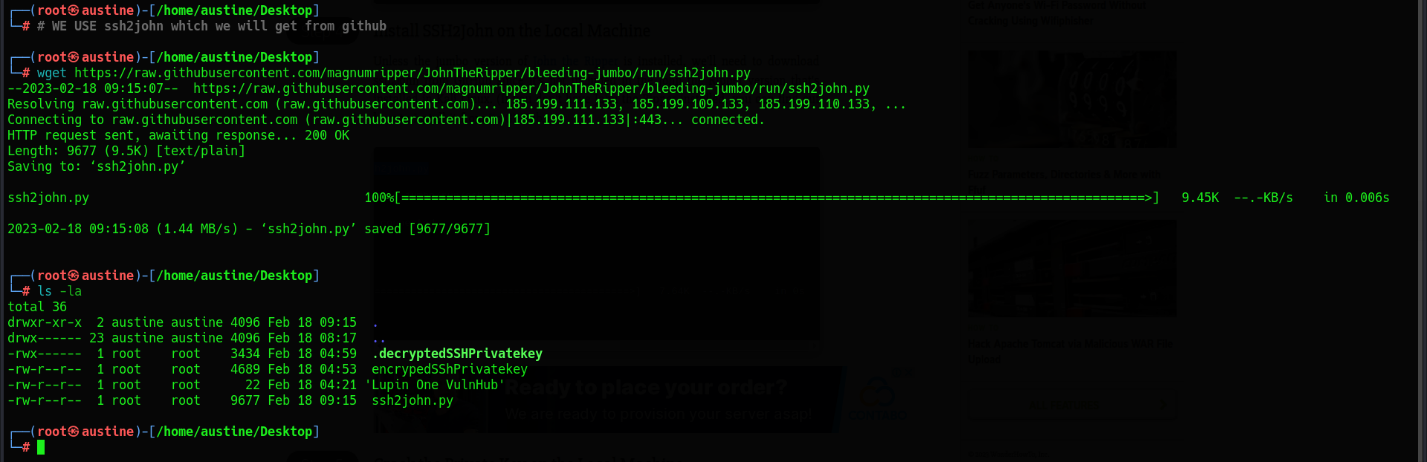


**Step 11: Using the ssh2john**

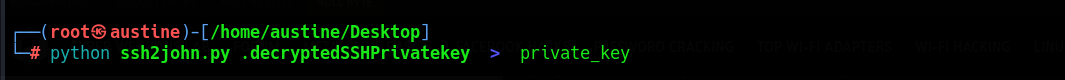
To get ssh2john, use the following link:

**wget** [**https://raw.githubusercontent.com/magnumripper/JohnTheRipper/bleeding-jumbo/run/ssh2john.py**](https://raw.githubusercontent.com/magnumripper/JohnTheRipper/bleeding-jumbo/run/ssh2john.py)

**NB: Ensure it resides in the same folder to where your ssh private key is located. For simplicity or you can input the path where your ssh private key resides.**



Generating a new hash format using the ssh2john.

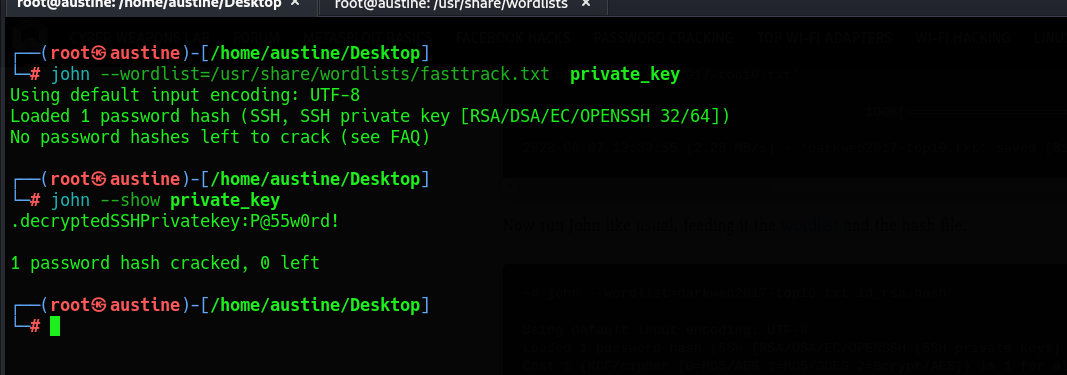


The command above will generate a new hashed file that we will use to crack the passphrase.

In my example, I have saved the new hashed file as private\_key.

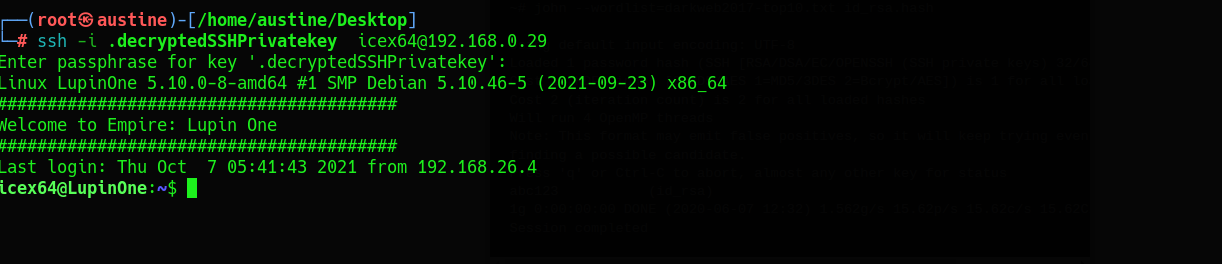
**Step 12: Using john the ripper.**

NB: We are using the fasttrack file since it was hinted out to use in the previous section (step 4). If you are successful, you should be able to see the passphrase as : **P@55w0rd!**



**Step 13: WE ARE IN!!!**

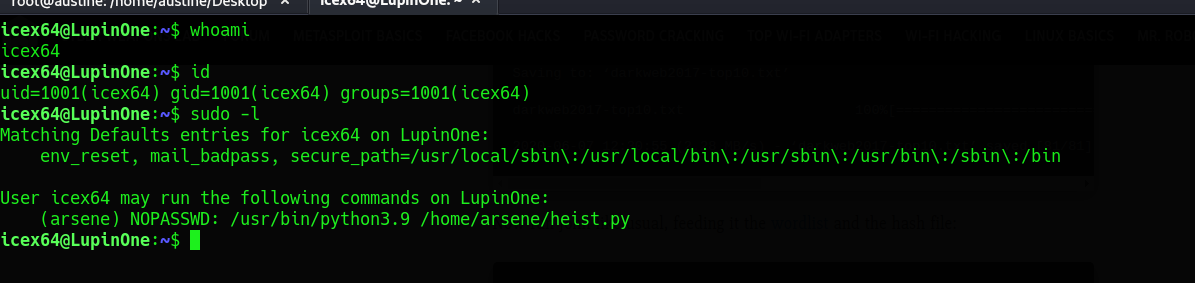
We have successfully logged in into the system as icex64



Step 14: Check the command that icex64 can run with sudo privileges.

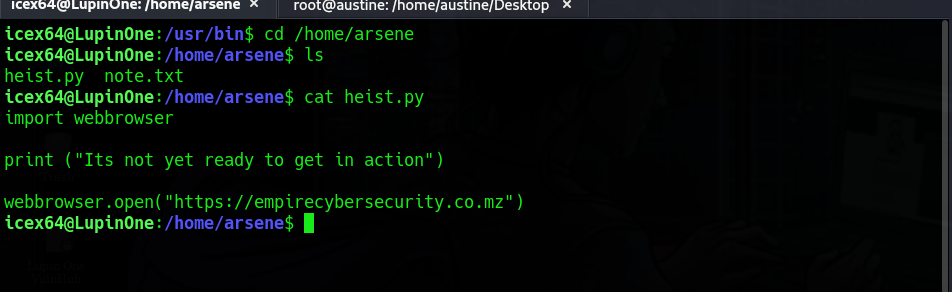
**NB: Feel free to explore around the system, you might discover things that I didn’t. Though I highly doubt it**.

We notice that icex64 can run **/usr/bin/python3.9 /home/arsene/heist.py**

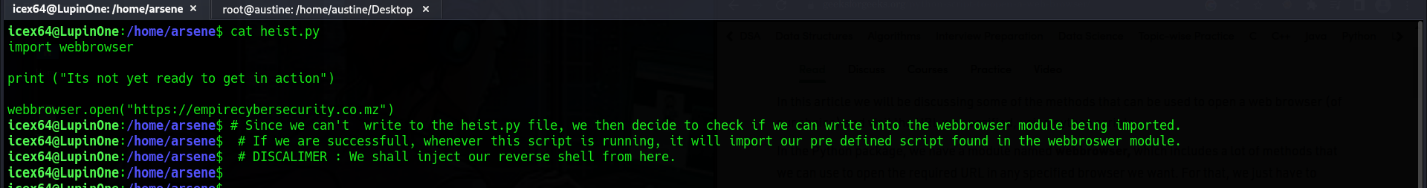


**Step 15: Looking at the contents of heist.py**

It’s a simple python script that prints the message “ It’s not yet ready in action “ and proceeds to open the browser to the specified link in the script.

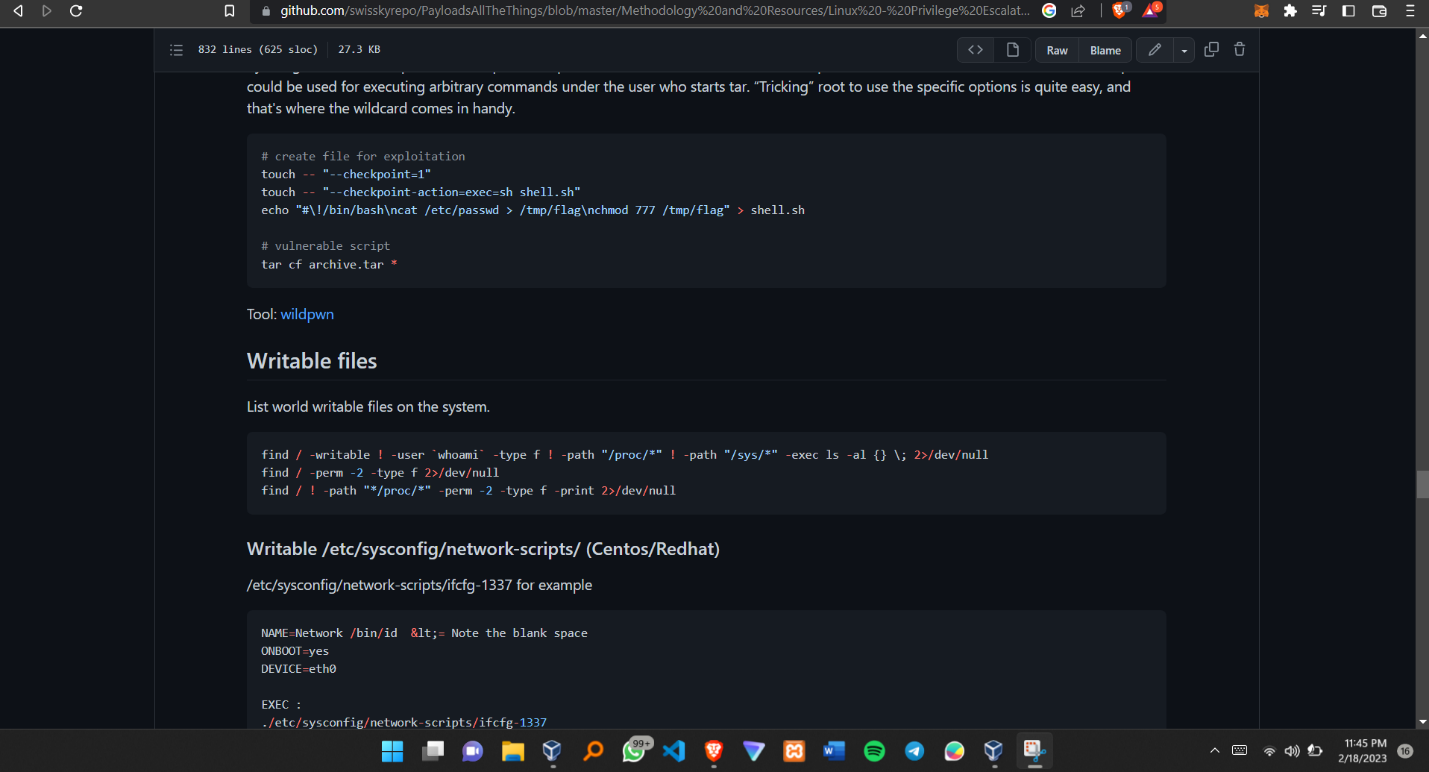


Step 16: Explore for vulnerabilities.

Since this script is importing the webbroswer module, it would be a great way if we could inject our script in the webbroswer module such that when the heist.py script is run, we can import our malware into the system.

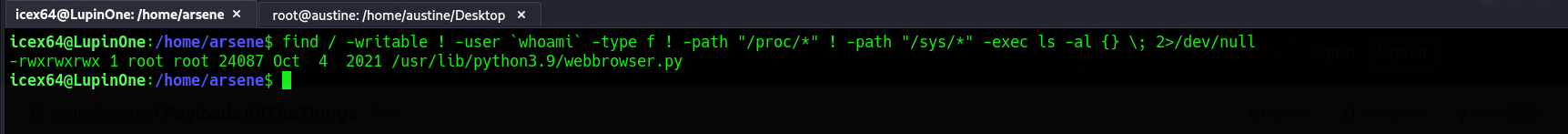
**Step 17 : Check if the webbroswer file is writable.**

GitHub repo below

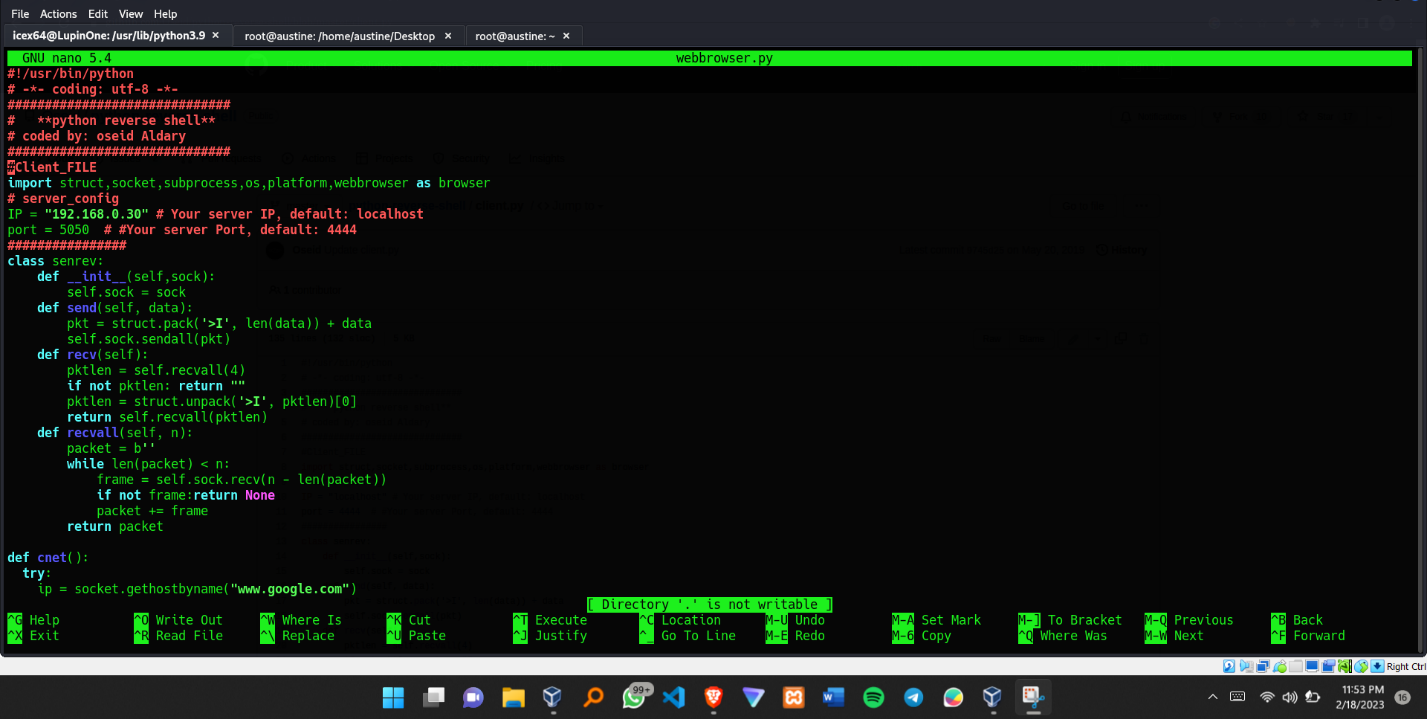


Running the command in the system.

And true enough, we find that the webbroswer.py file is writable.

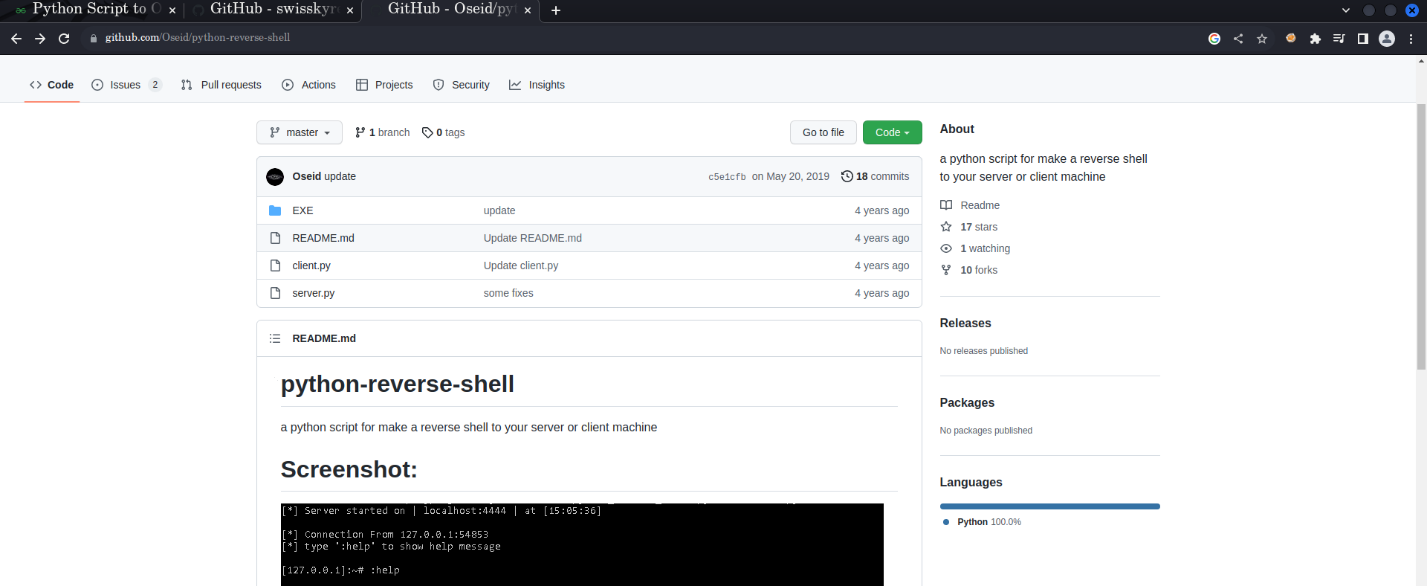


**Step 18: Replacing the contents of the webbrowser.py with the python reverse shell script.**

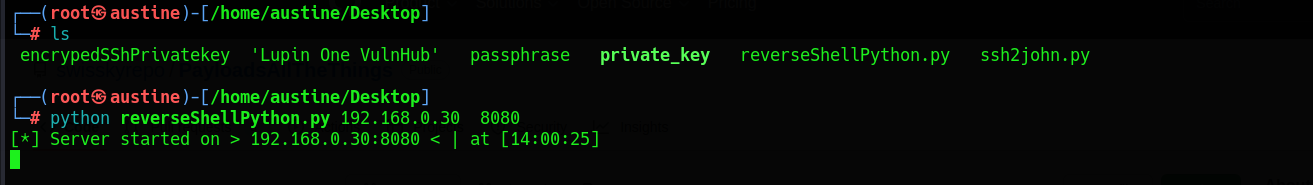
After finding the path where the webbrowser.py is located, we go ahead and replace the file with a python reverse shell script

**NB: Make sure you change the ip and the port number.**

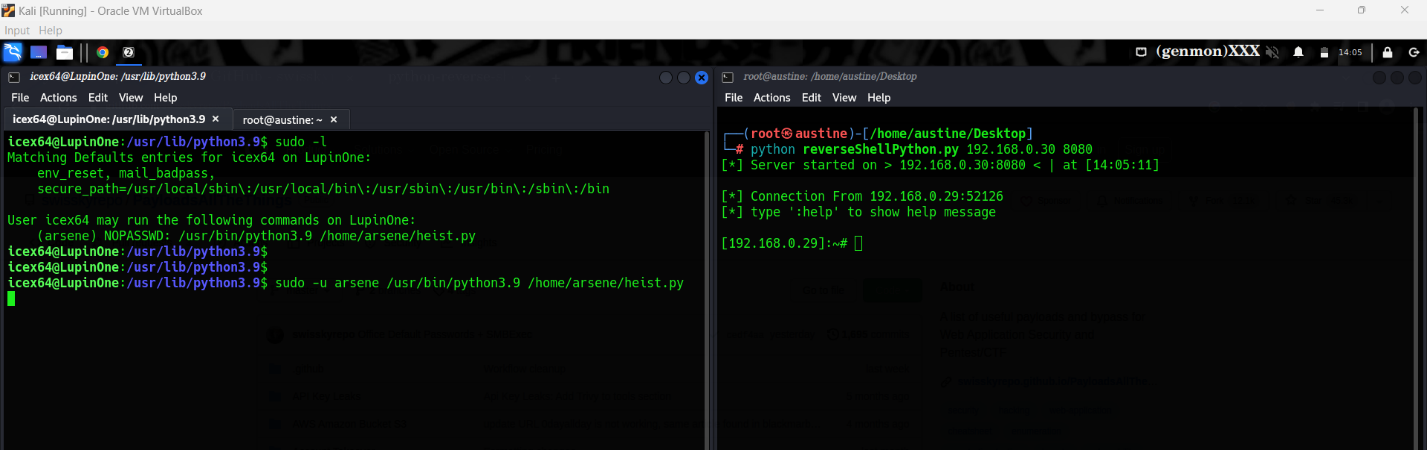
The GitHub to where I got the reverse shell script.

****

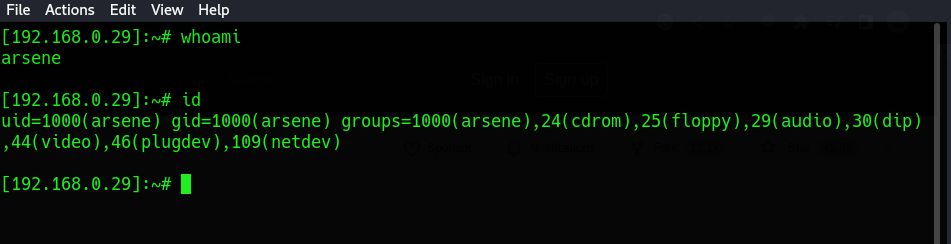
**Step 19: Run the listener. Make sure the IP and Port used in the client and server file are the same.**



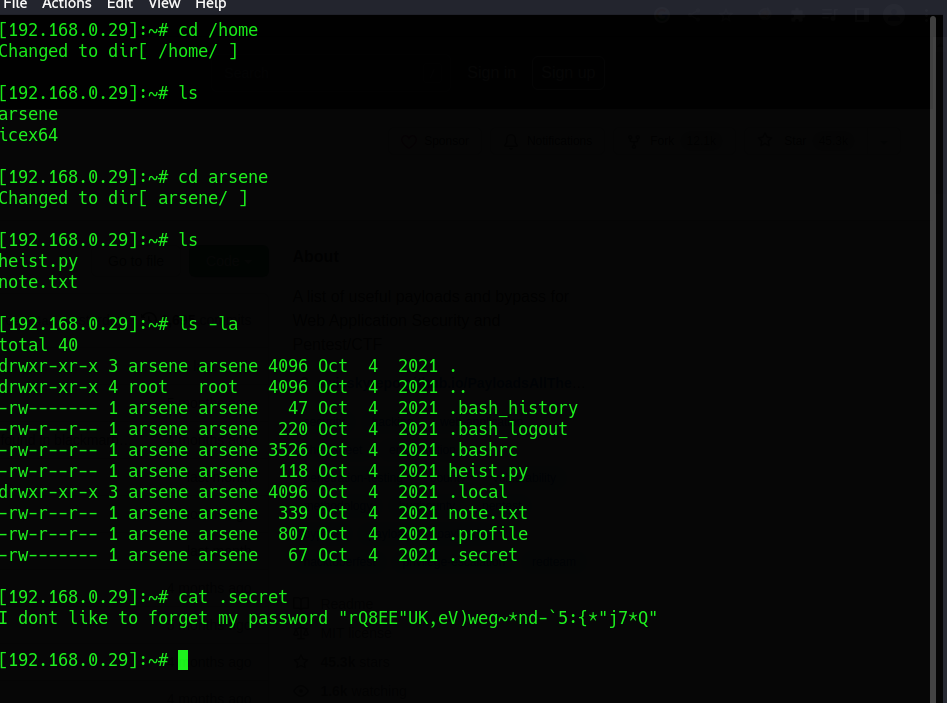
**Step 20 : Run the sudo command that icex64 can run as arsene.**

As discussed earlier, this will, import the webbroswer.py file that contains our injected reverse shell thus prompting us to get as a successful reverse shell.

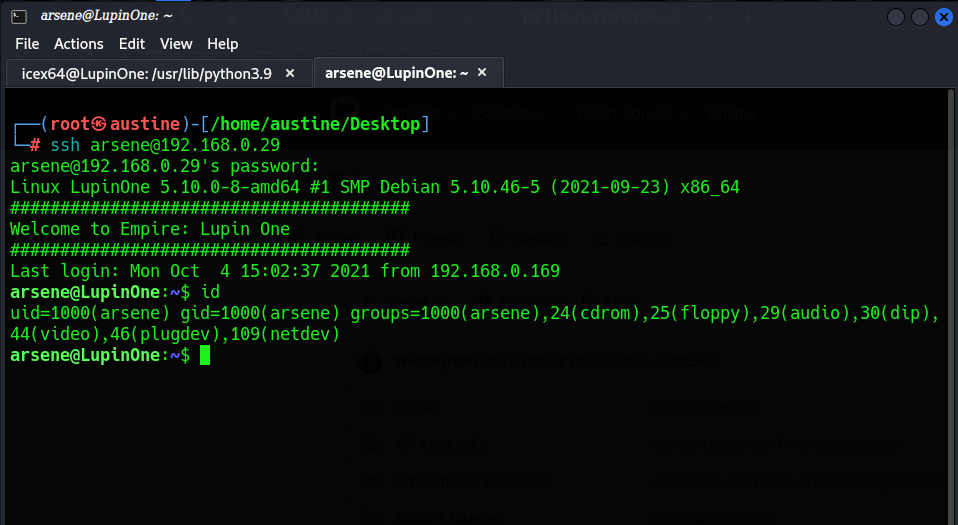
**Step 21 : Check you current Id and now we can get arsene’s password.**



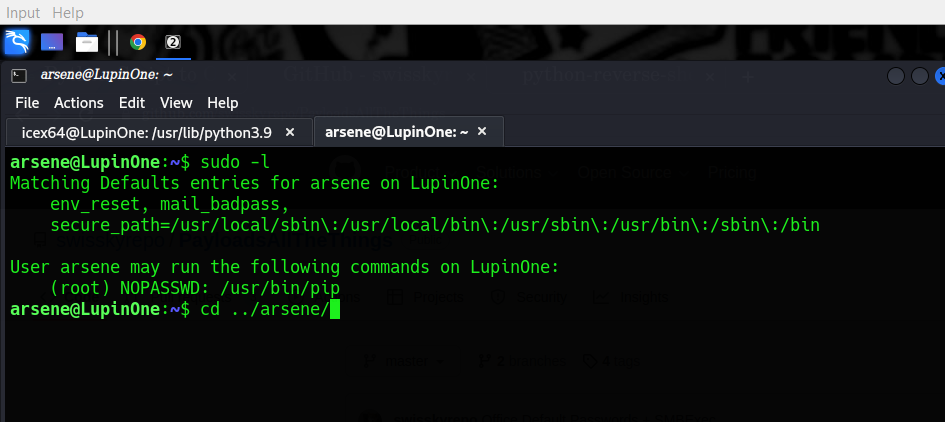
Arsene’s password is contained in a hidden file called secret. As seen below.



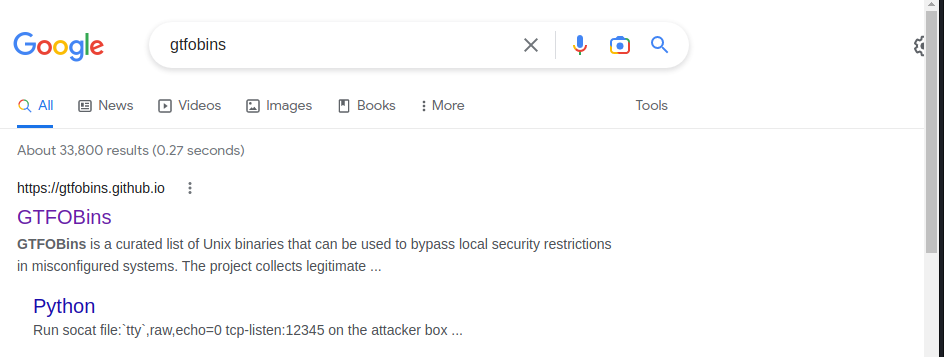
Since the reverse shell sometimes is not stable and tends to terminated abruptly, we can login into the system using ssh and using arsene as the username. We login using the password we previously obtained from the secret file above.



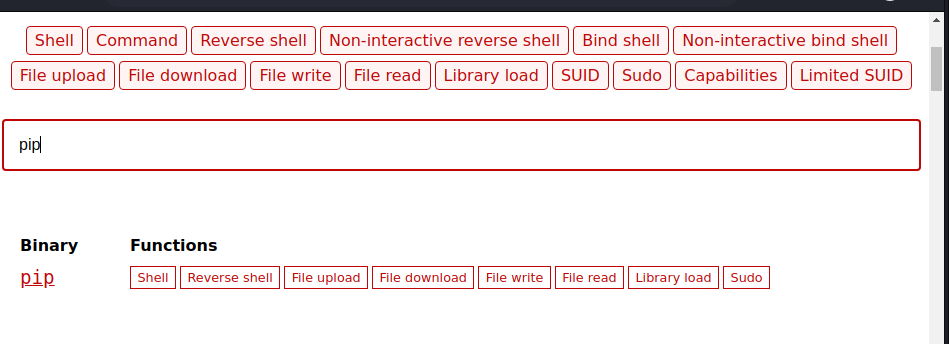
**Step 22: Check the commands that arsene can run as root.**

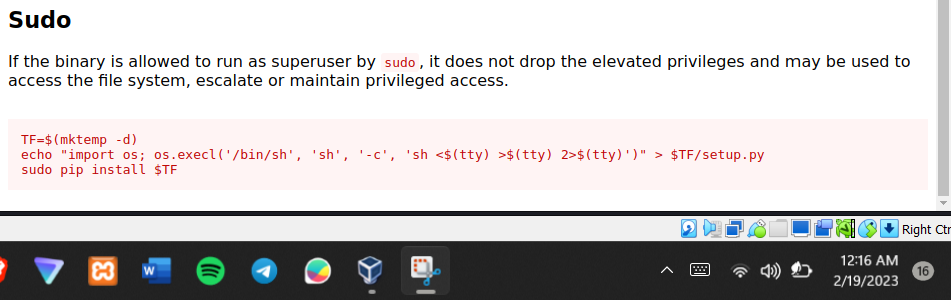


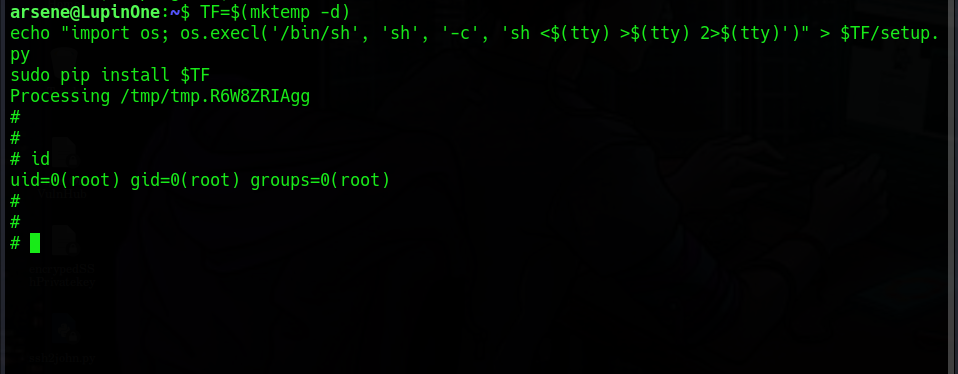
**Step 23: Privilege escalation using GTFOBins**



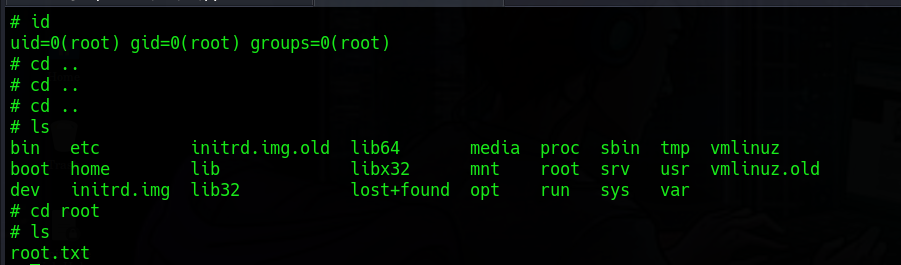
From the sudo command above, arsene can run **/usr/bin/pip** and using the GTFOBins we search for priviledge escalation that contains the pip



Under the pip category we choose sudo since that what we want.

Copy and paste the command above in the terminal and immedietly you will have been escalated to root.

**Step 24: We get the root.txt file**



**Step 25: WE DID IT TOGETHER. HURRAY**

