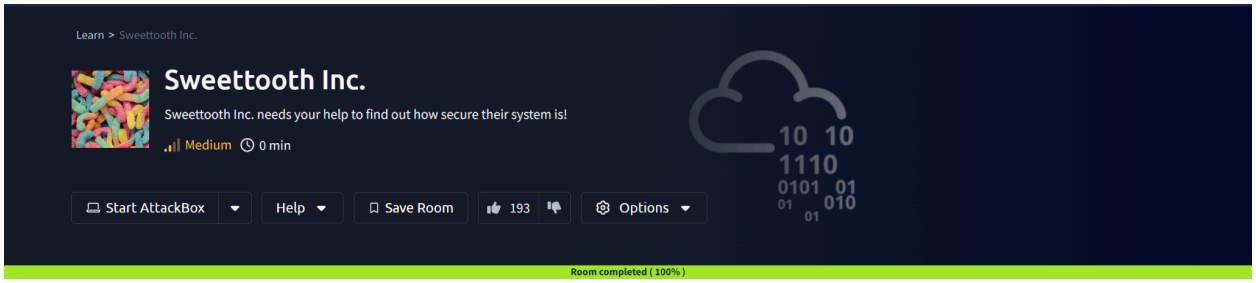
SWEETTOOTH INC

ASSIGNMENT REPORT



Peter Kinyumu,
cs-sa07-24067,
June 26th, 2024.

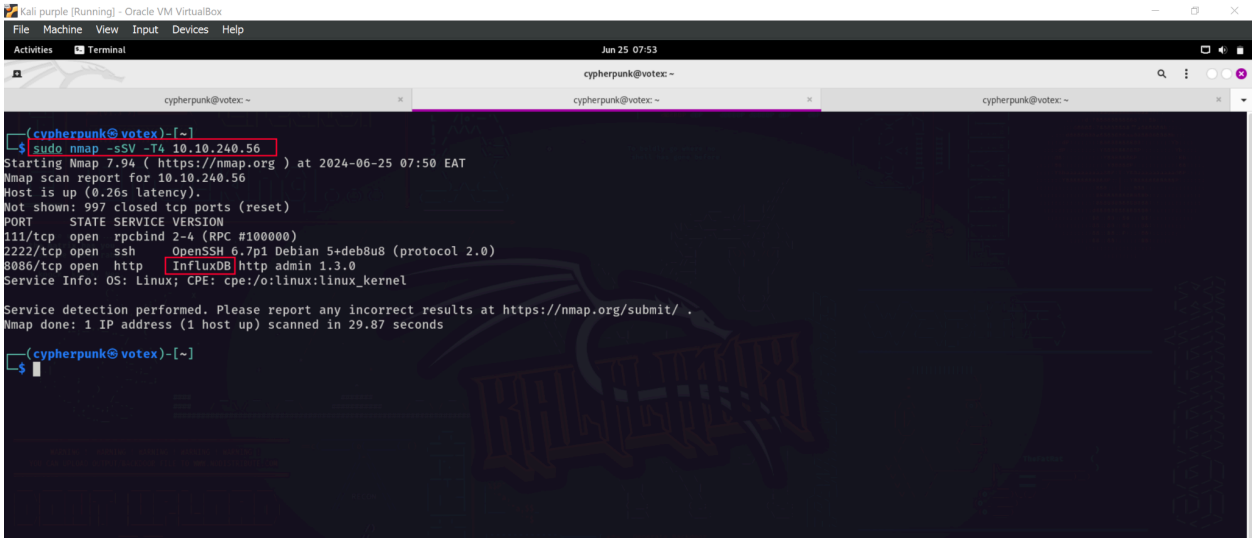
1. INTRODUCTION

This room was an exploratory learning experience on attacking a poorly configured influxDB. There is no step-by-step guide on what to do. You have a goal to achieve, say get a valid user to the database and it's up to you to find out how to do that. A backbox penetration testing setup.

2. ANSWERS TO QUESTIONS

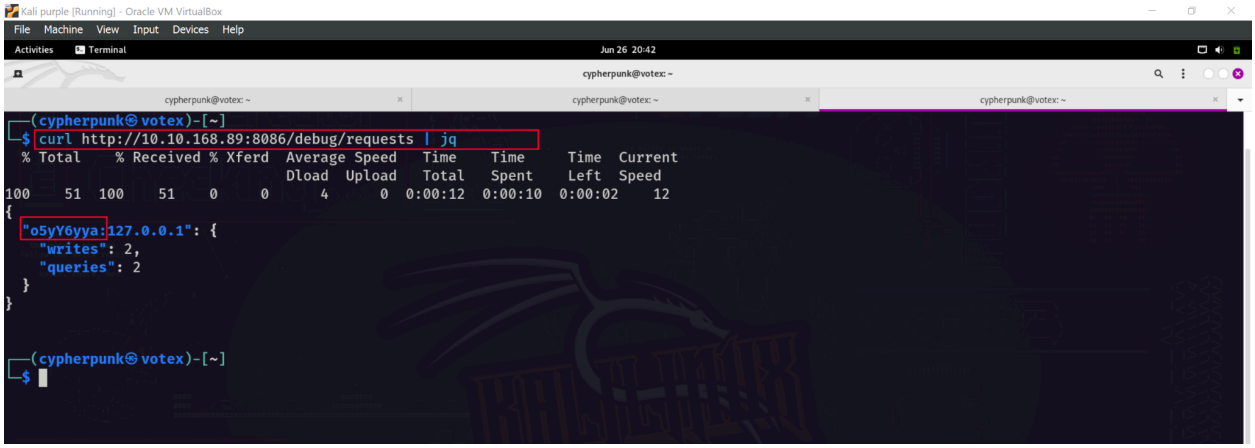
Enumeration

a. **Do a TCP portscan. What is the name of the database software running on one of these ports?**
   - Perform an Nmap service and version scan on the target. `nmap -sSV -T4 <target_ip>`
   - InfluxDB is running on port 8086.



Database Exploration

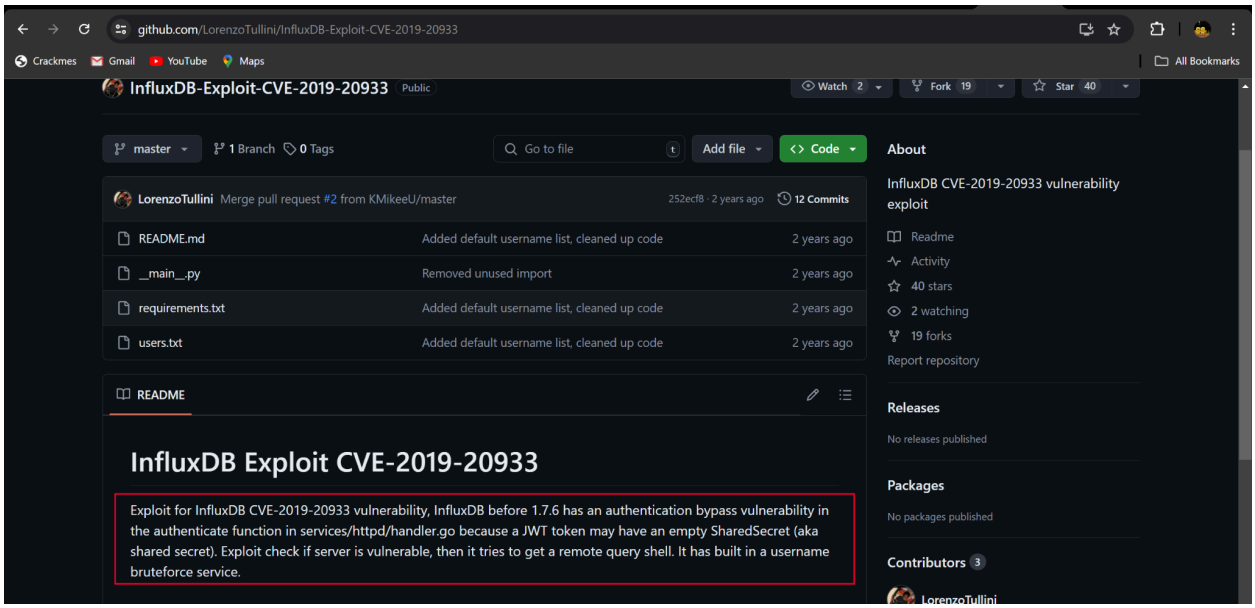
a. **What is the database user you find?**
   - InfluxDB has the **/debug/requests** endpoint that provides debugging information related to the queries being handled by the server.
   - When access to this endpoint is not restricted based on the database configuration, we can exploit this configuration to perform enumeration and expose more information about the database e.g the username ash shown below.



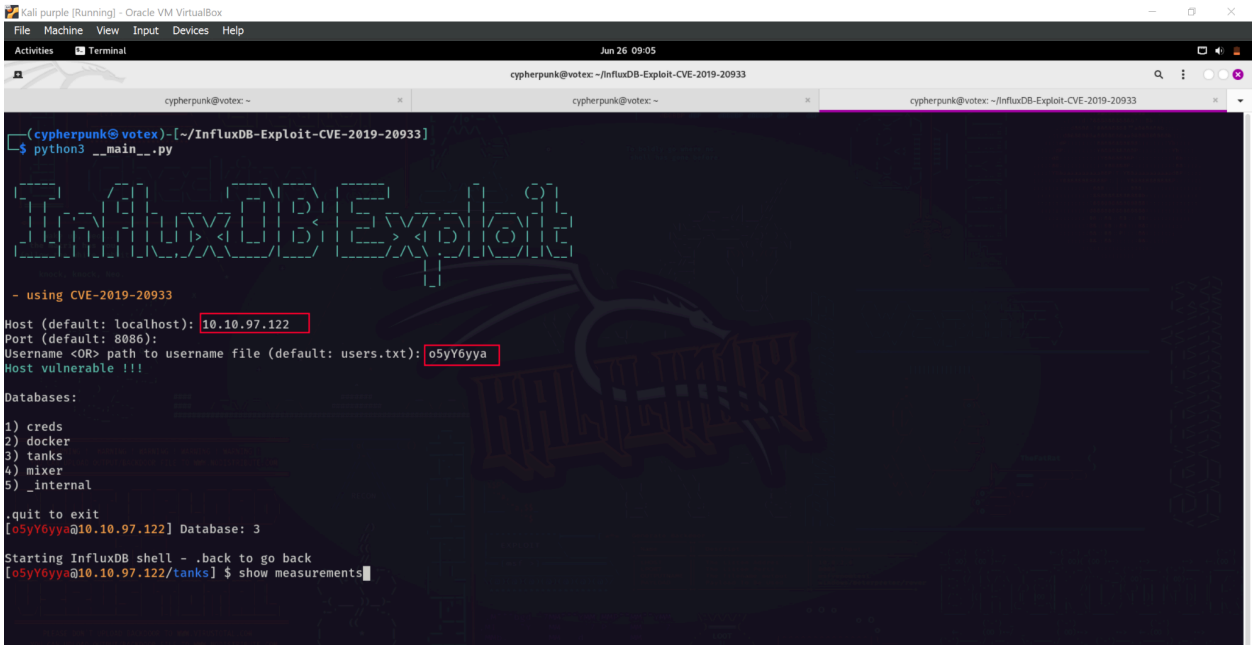
b. What was the temperature of the water tank at 1621346400 (UTC Unix Timestamp)?

- This question required access to the influx databases and measurements(tables), which I didn't have directly with the found user.
- However, I found a vulnerability affecting influxDB versions below 1.7.6 which i used to gain access to the database. You just have to clone it and run some setup commands as explained in the repository.

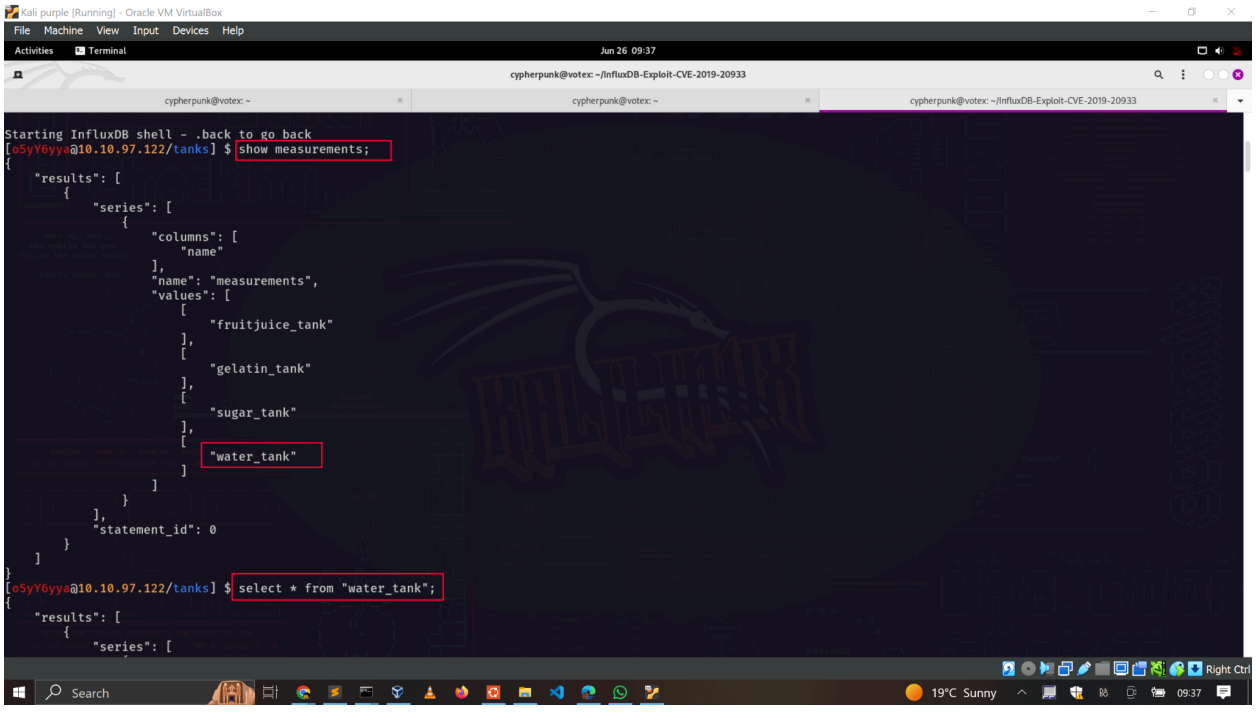
Link => https://github.com/LorenzoTullini/InfluxDB-Exploit-CVE-2019-20933



- Run the Python program and input the host IP address and the user we identified earlier then select the **tanks** databases.

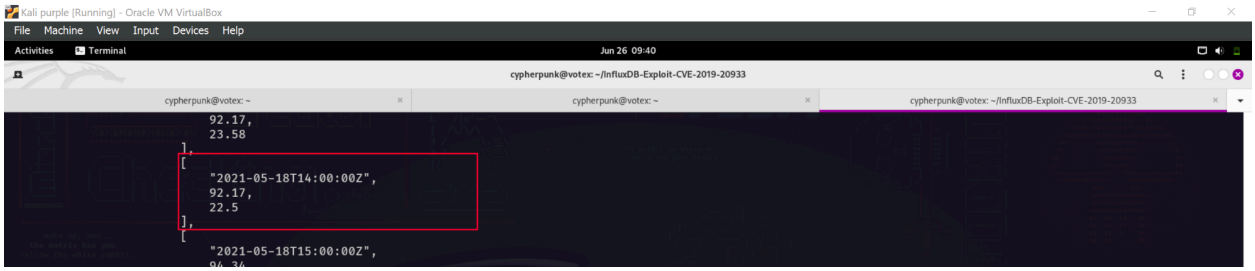


- If you view the tables in the database you will notice the **water_tank** table in question.
- We can then retrieve fields from the table with the SELECT statement as shown.



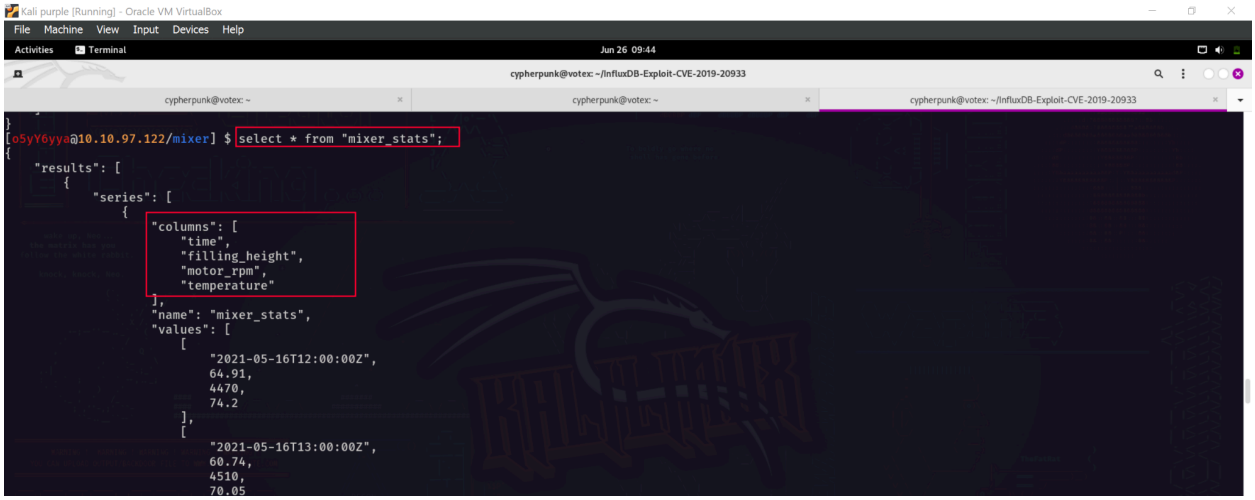
- The timestamp was in UNIX time so you need to convert that to the ISO standard form which is **2021-05-18T14:00:00Z**

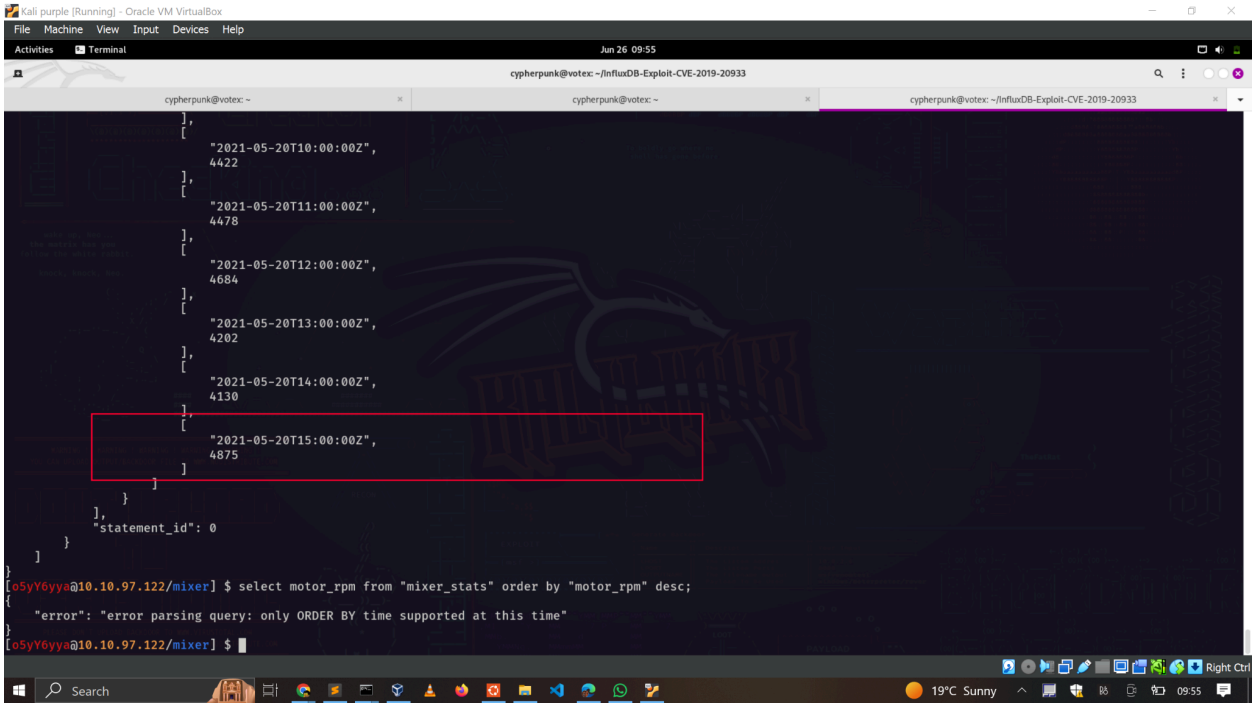
- The temperature at the specified time was **22.5.**



c. What is the highest rpm the motor of the mixer reached?

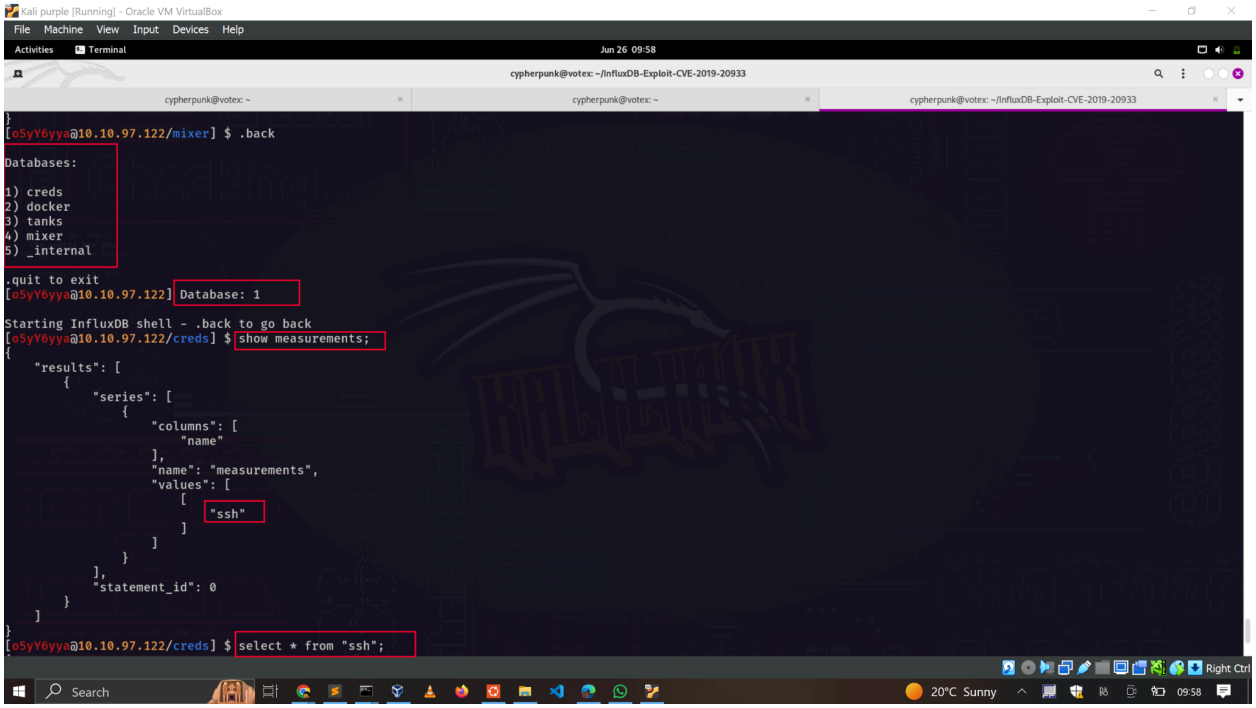
- Choose the mixer database
- From the mixer database run **show measurements** command to view the tables that exist in the database. The **mixer_stats** table will be available.
- Retrieve data from the **mixer_stats** table to get the highest rpm the rotor reached.

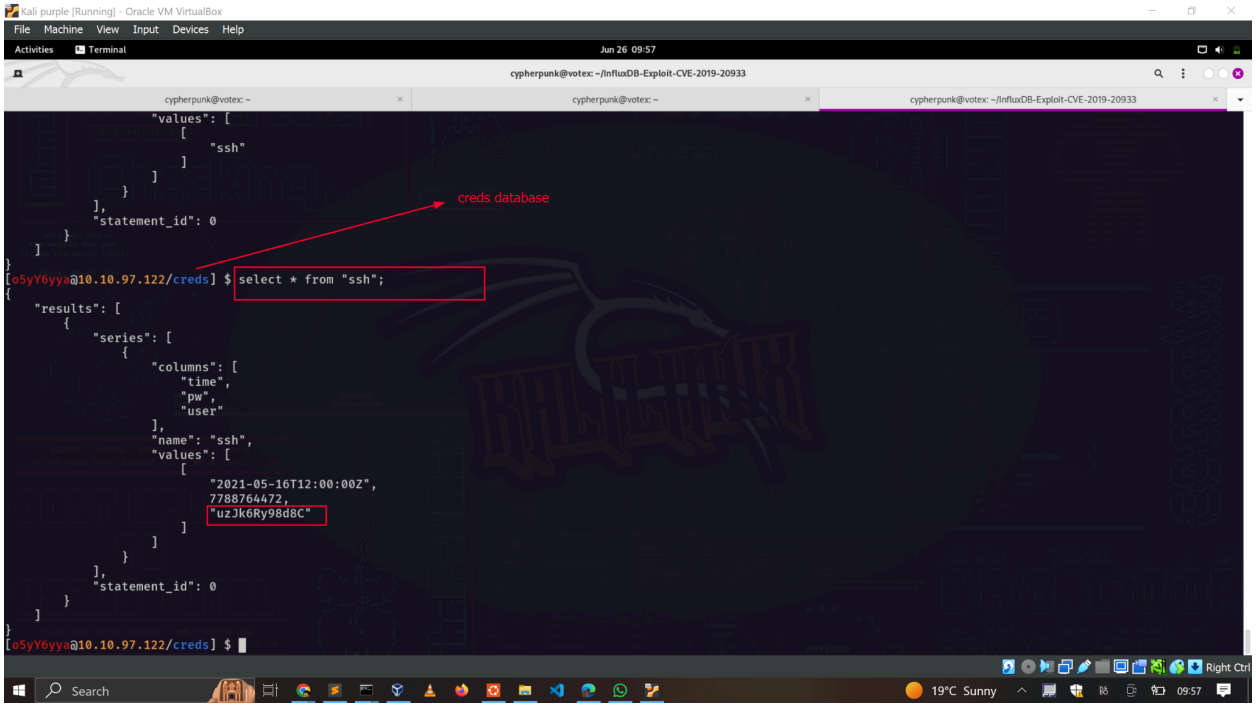




d. What username do you find in one of the databases?

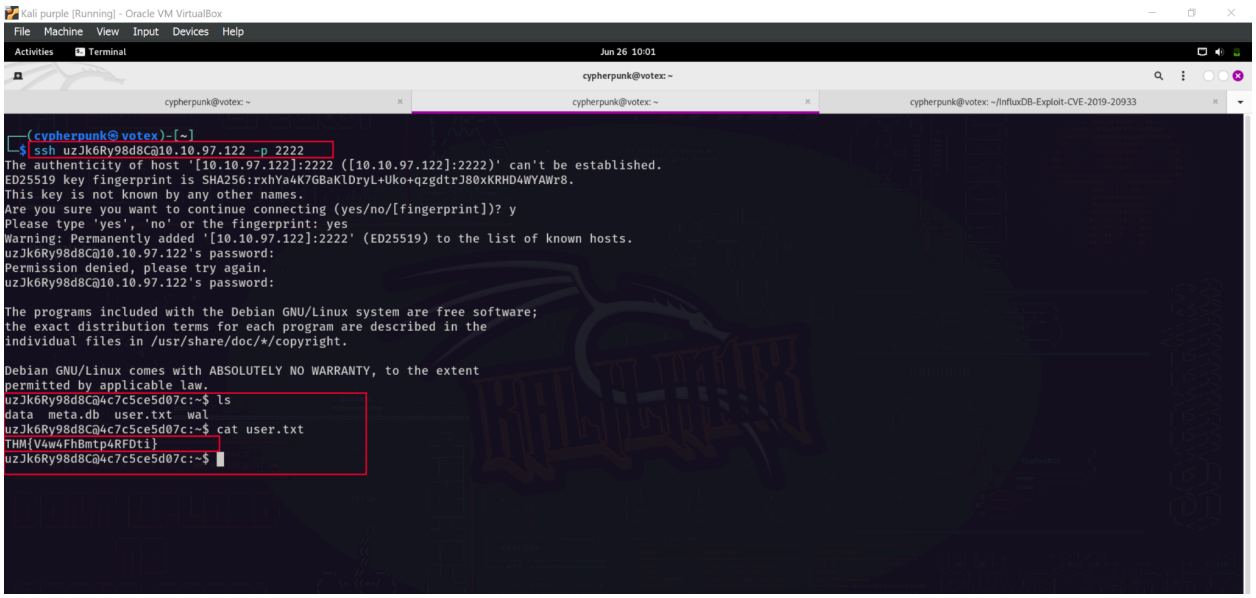
- Choose the creds database
- Retrieve details from the **ssh** table.





e. user.txt

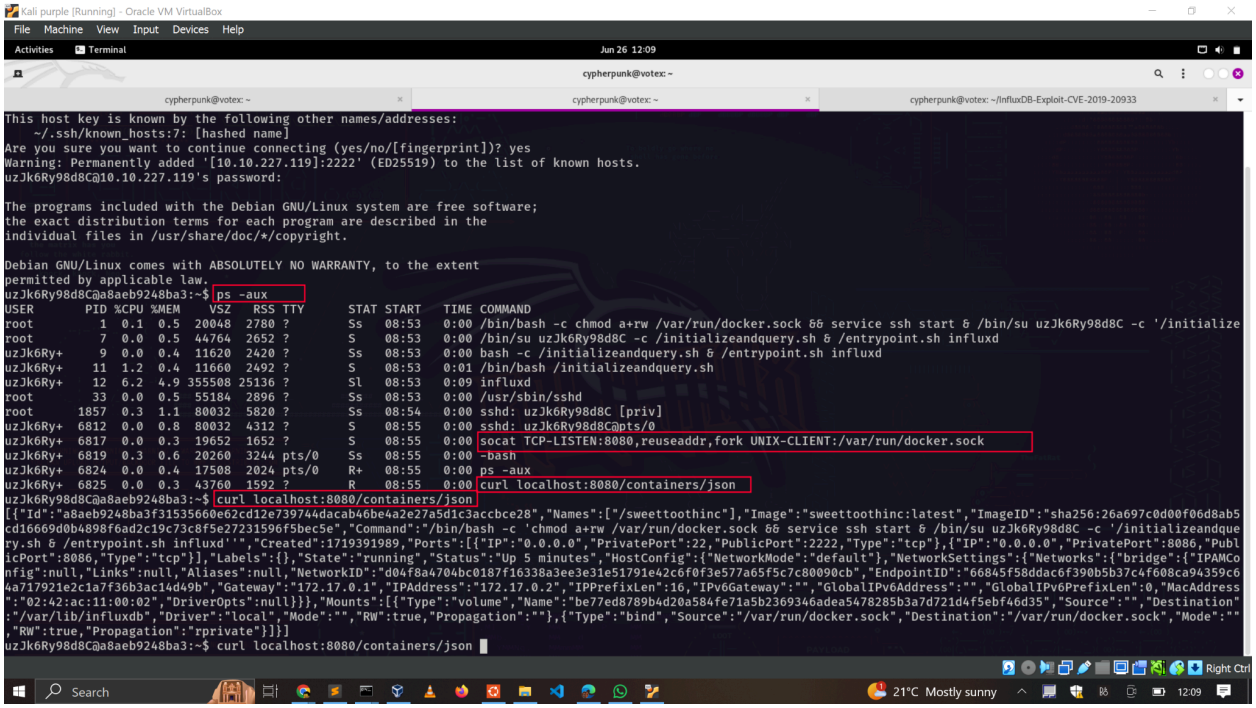
- This question requires us to read the user.txt file.
- Having secured the ssh credentials from the previous question, we can use those creds to ssh to the target and read the specified file.



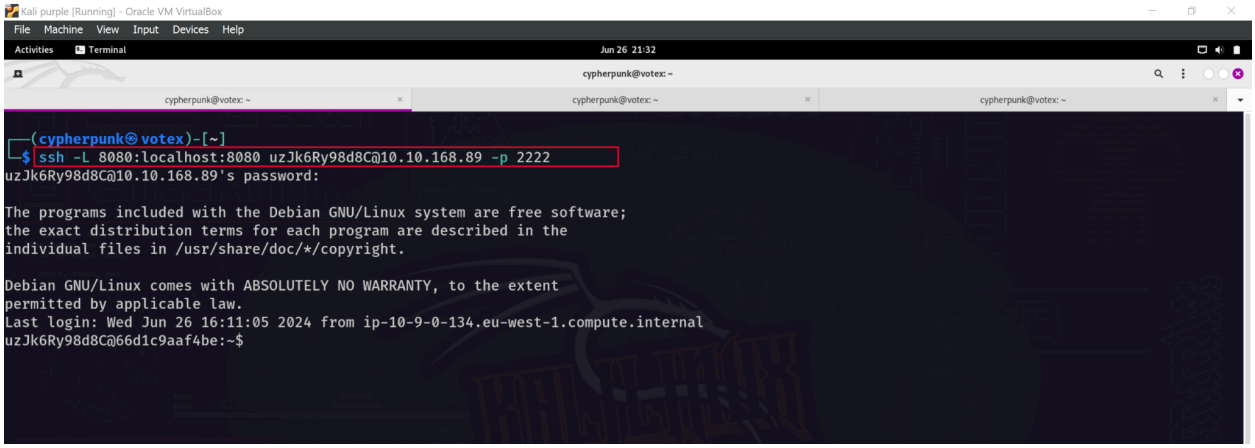
Privilege Escalation

a. /root/root.txt

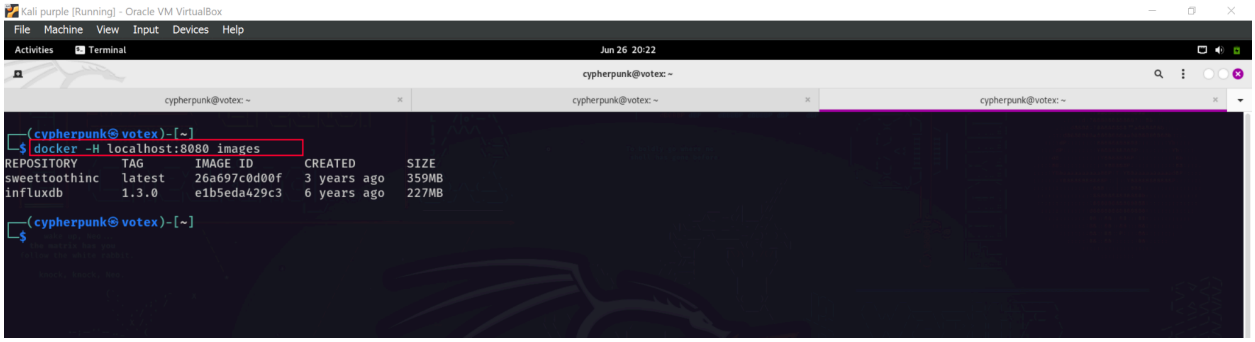
- We are required to gain privilege escalation and read the file root.txt in the rot folder.
- First thing I noticed, the command sudo was not available in the machine and I had no permissions to view the root folder.
- I decided to check the running processes to see if I could find a way to elevate my privileges somehow someway.
- With **docker** among the running processes, it hinted that we are probably in a docker environment, and I also noted the docker demon is running locally on port 8080.



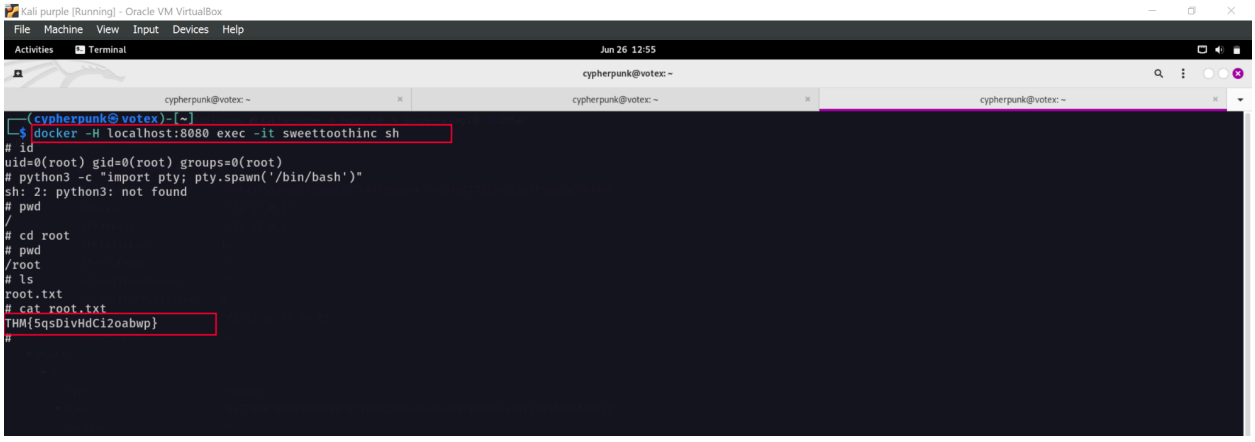
- After long hours of research, I found a way to abuse docker and gain privilege escalation. However, it required a docker command, which wasn't available in the remote machine.
- Since the docker demon is running locally on port 8080, I resolved to set up local port forwarding and bind port 8080 on my attack machine to port 8080 on the remote machine since I have docker installed on my machine.



- With that, it became easier to connect to the docker instance and run commands like listing the images that are available.



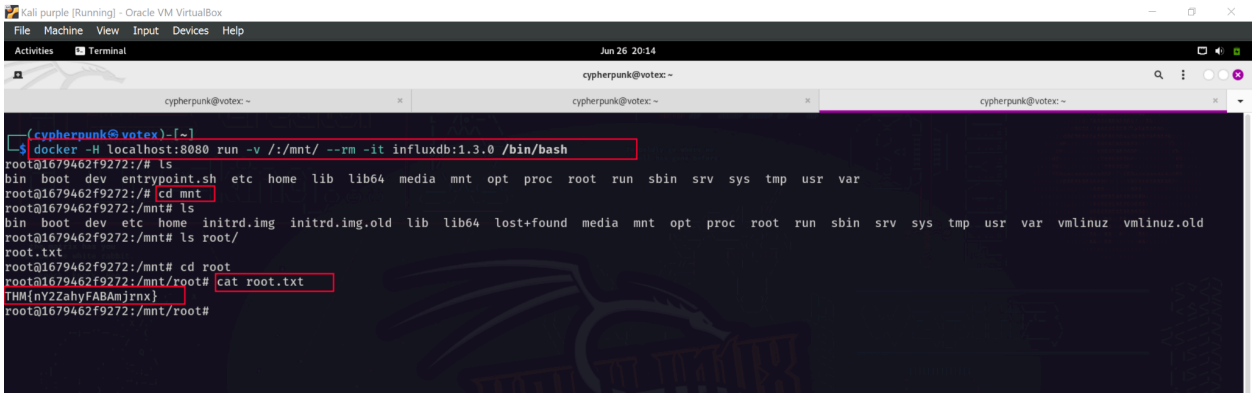
- We see there are two images. If we connect to the first one and execute the **sh** command, we get shell access with root access to the container and can retrieve the flag.



Escape

a. The second /root/root.txt

- This challenge required us to break out of the docker container and gain access to the underlying host system.
- One way we can do that is by mounting the root file system of the underlying host into the container.
- By default, containers run as root since the docker demon also runs as root.
- So, by mounting the root file system into the container, we will have root access to all the files, and we can modify them as we want.

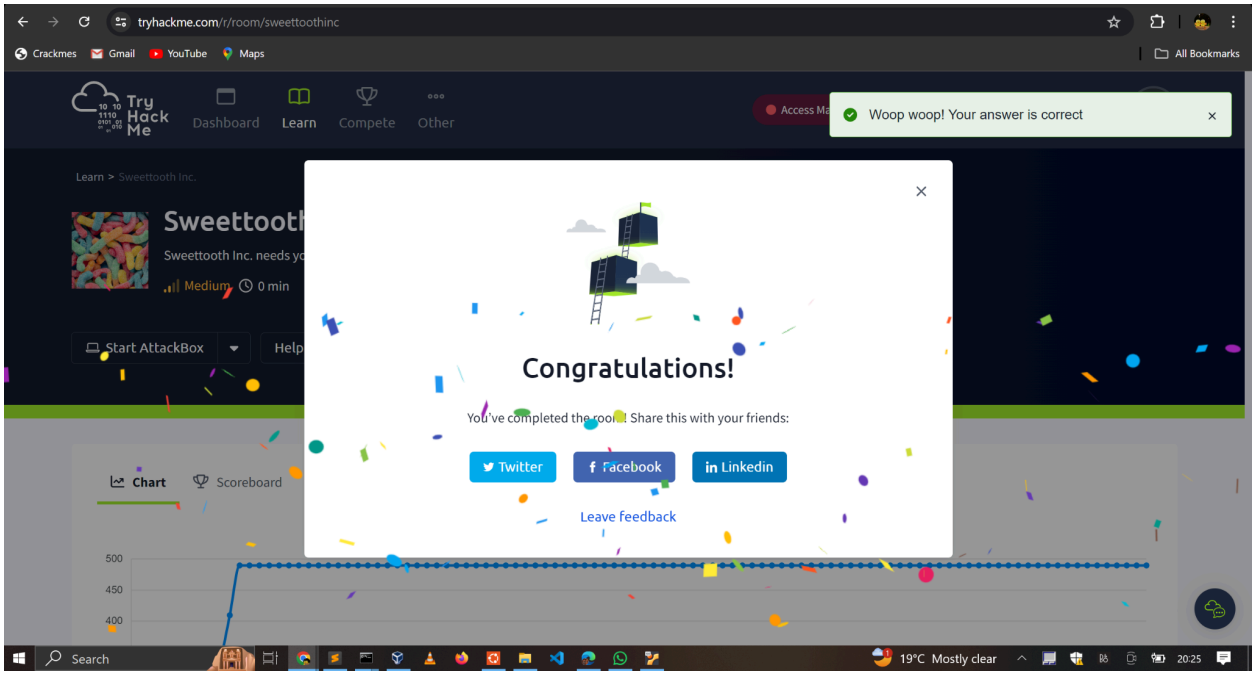


The above command `docker -H localhost:8080 run -v /:/mnt –rm -t influxdb:1.3.0 /bin/sh` connects to the docker demon running on localhost with the -H option, starts a new container from the specified influxdb image, mounts the root directory of the host machine to the /mnt director of the container with the volume(-v) switch and lastly starts an interactive bash shell in the container.

With that we are able to access the **/mnt** directory and manipulate the files as we need, also view the flag.

3. MODULE COMPLETION

https://tryhackme.com/p/c1ph3rbnuk



4. CONCLUSION

This was my best assignment so far. I have learned a lot, from how to interact with the influxDB, to finding weaknesses in it, gaining access, exploiting docker to gain elevated privileges and even owning the system it runs on. The challenge to grope around looking for answers really built my mindset into that of an attacker and also improved my research skills. Looking forward to more of these “do it yourself” sets of challenges.