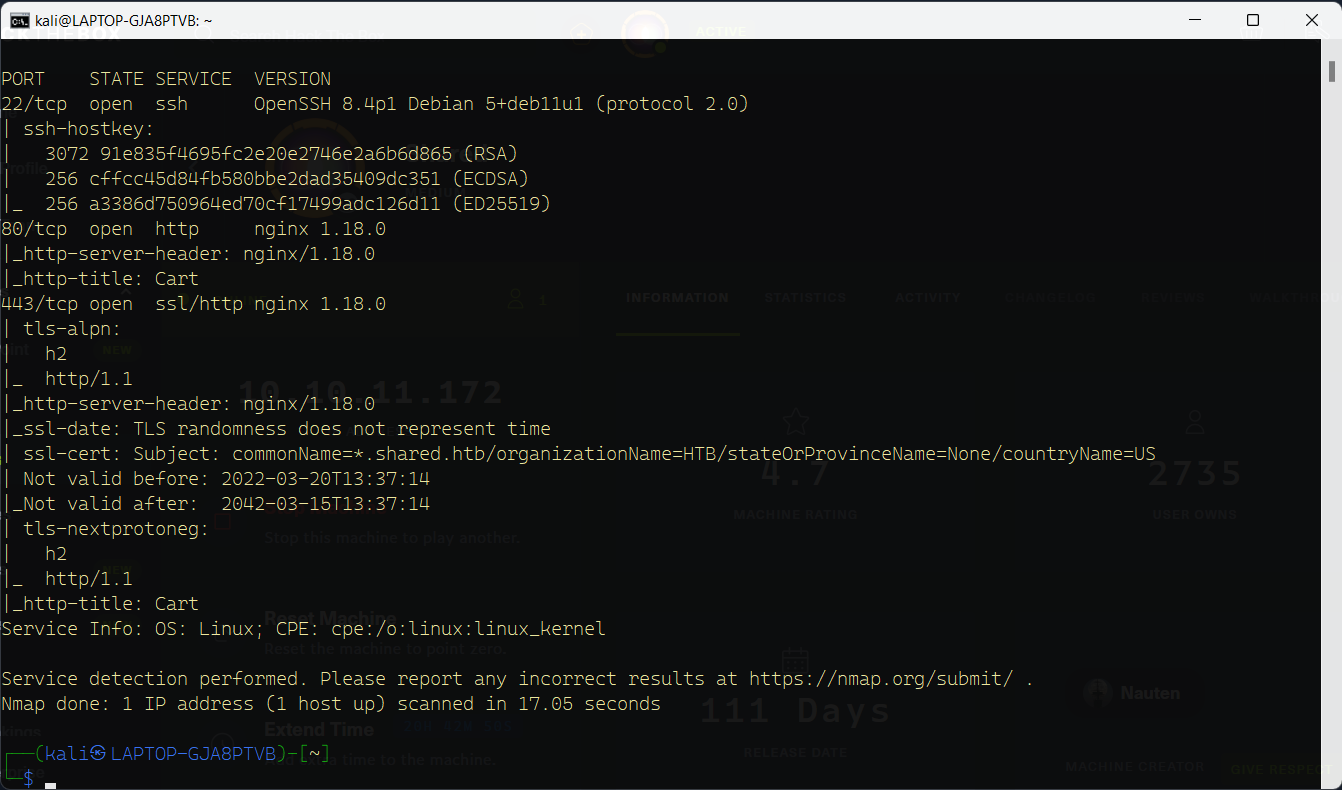
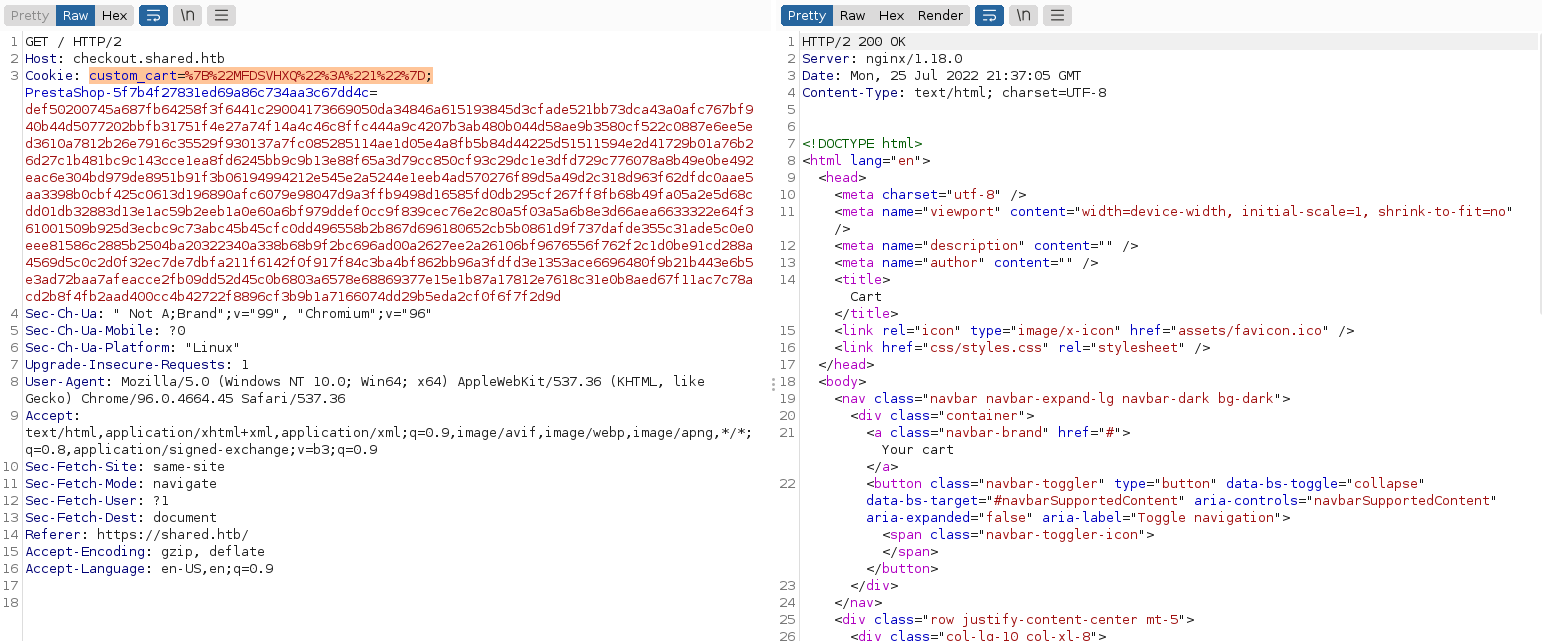
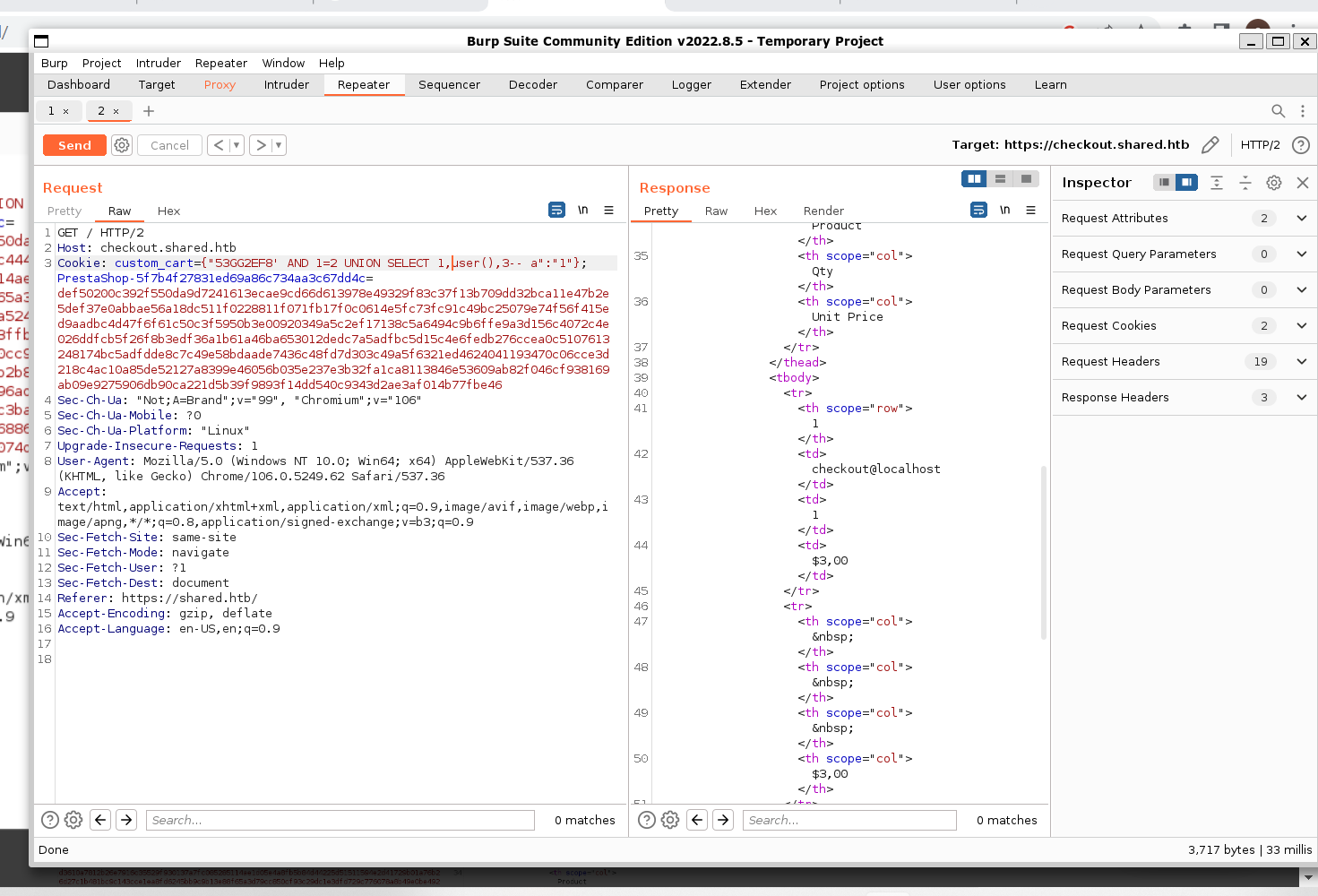
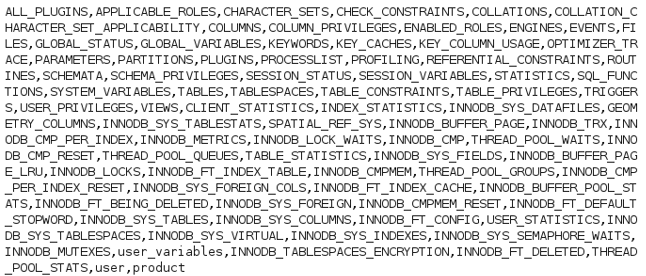
Target pinged

1. Nmap -sS -p- 10.10.11.172
2. Nmap -sS -sV -sC -p22,80,443 10.10. 11.172
3. Explore website, and try to place an order
4. Check request with burp request

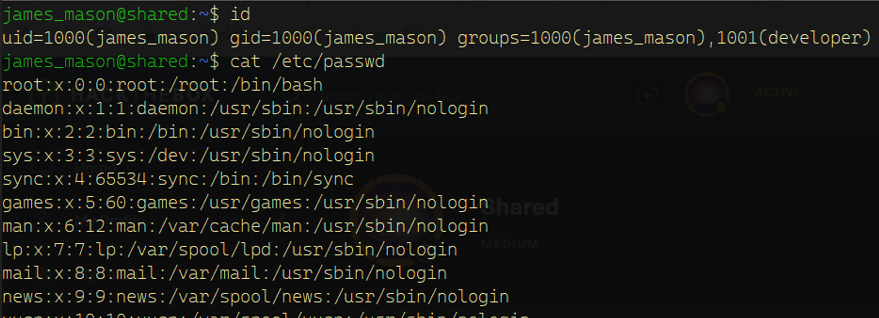


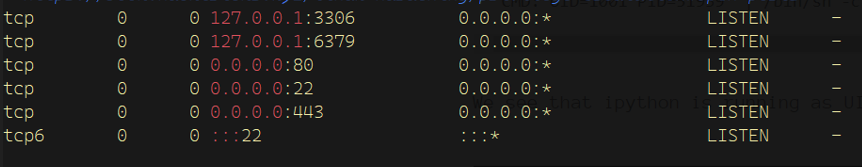
1. Ctrl Shift U, to decode, and test for SQL injection vulnerability.
2. Injection found, now lets dump all the users with their information.





This look a username and a hash. Lets decrypt the password with an online decrypting tool.

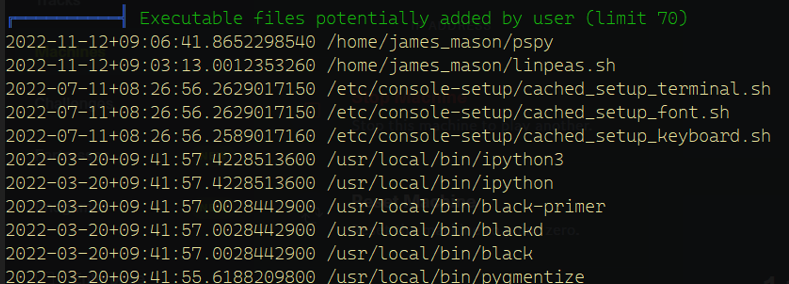
Now we can use these credentials to log in the user james\_mason using ssh. 

Lets upload linpeas.sh to get more information:



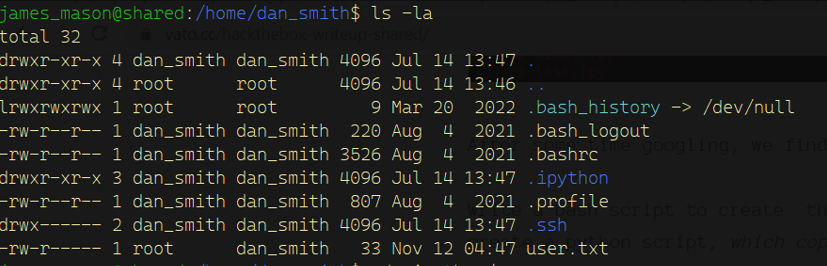
Here we that mysql and redis are running locally and port 80, 22, and 443 are running externally.

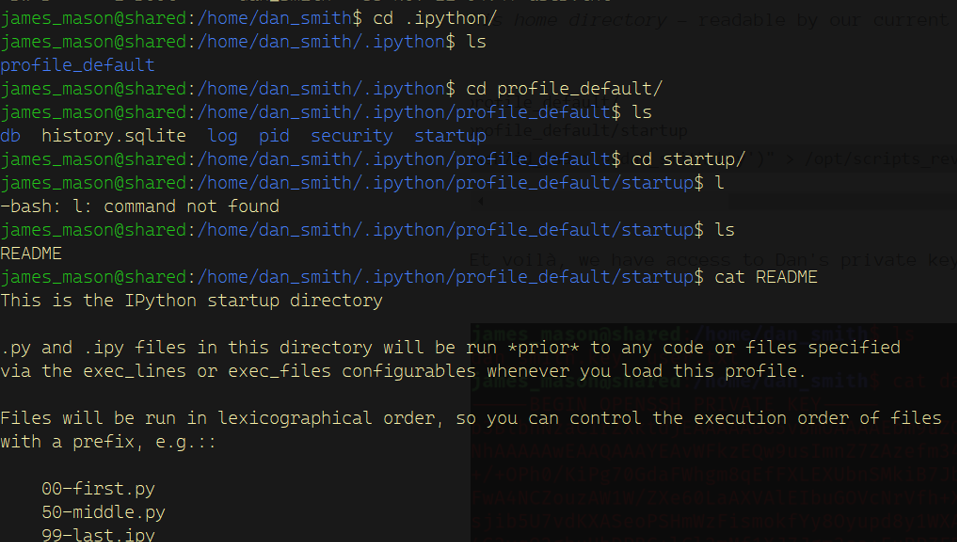
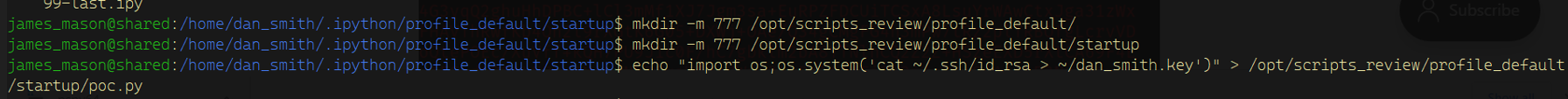
We also see dan smiths uid is equal to 1001.

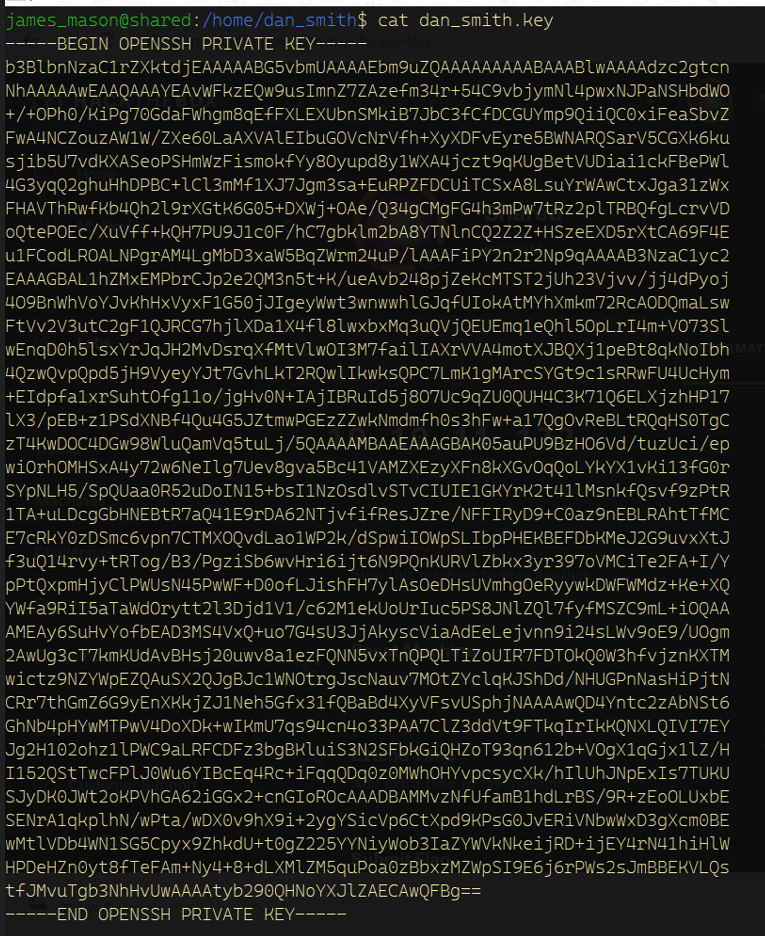


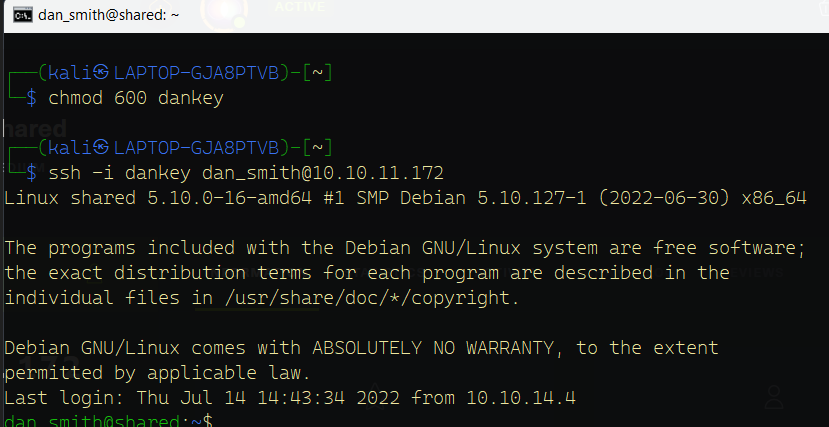
here we see that the user has ipython installed.

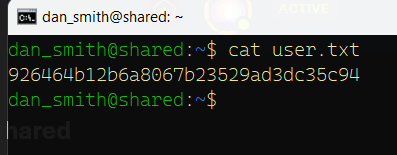
Interactive Python is a command shell for interactive computing in different languages.

Now if we start searching for our user flag, we’ll find it in the user: dan\_smith. But we don’t have the permissions to access it. If we try ls -la we can see all hidden directories: 

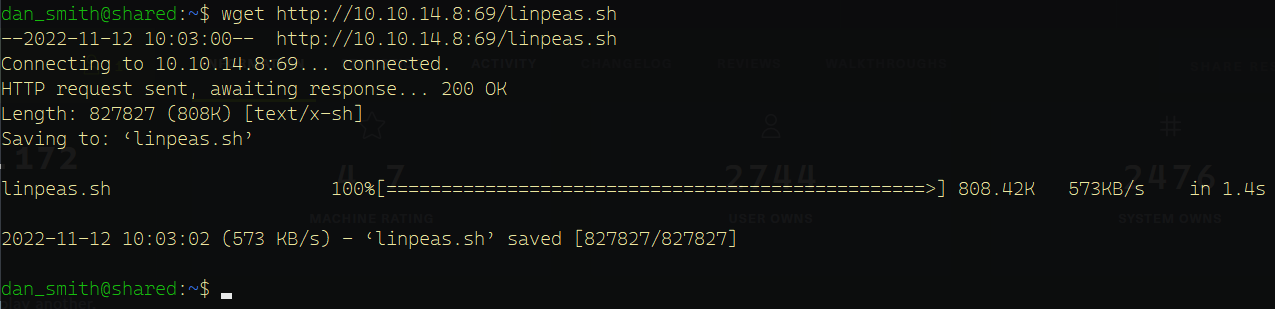
1. Lets access .ipython:
2. Now if we create the same directories found in .ipython and give a code, if executed, we’ll get dan\_smiths private key.
   1. There’s a directory named scripts\_review, I guess code firstly gets executed there, so lets put those directories and the script there.

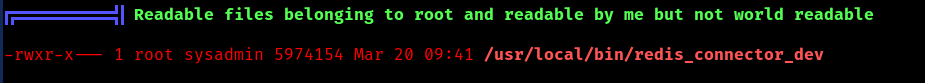
Now if we check dan\_smiths directory we’ll find his private key

1. Now go back to your own machine and create the next: dankey.key, and paste the private key found in james machine in there.
2. Ssh to dan\_smith with private key as password
3. Grab the first flag

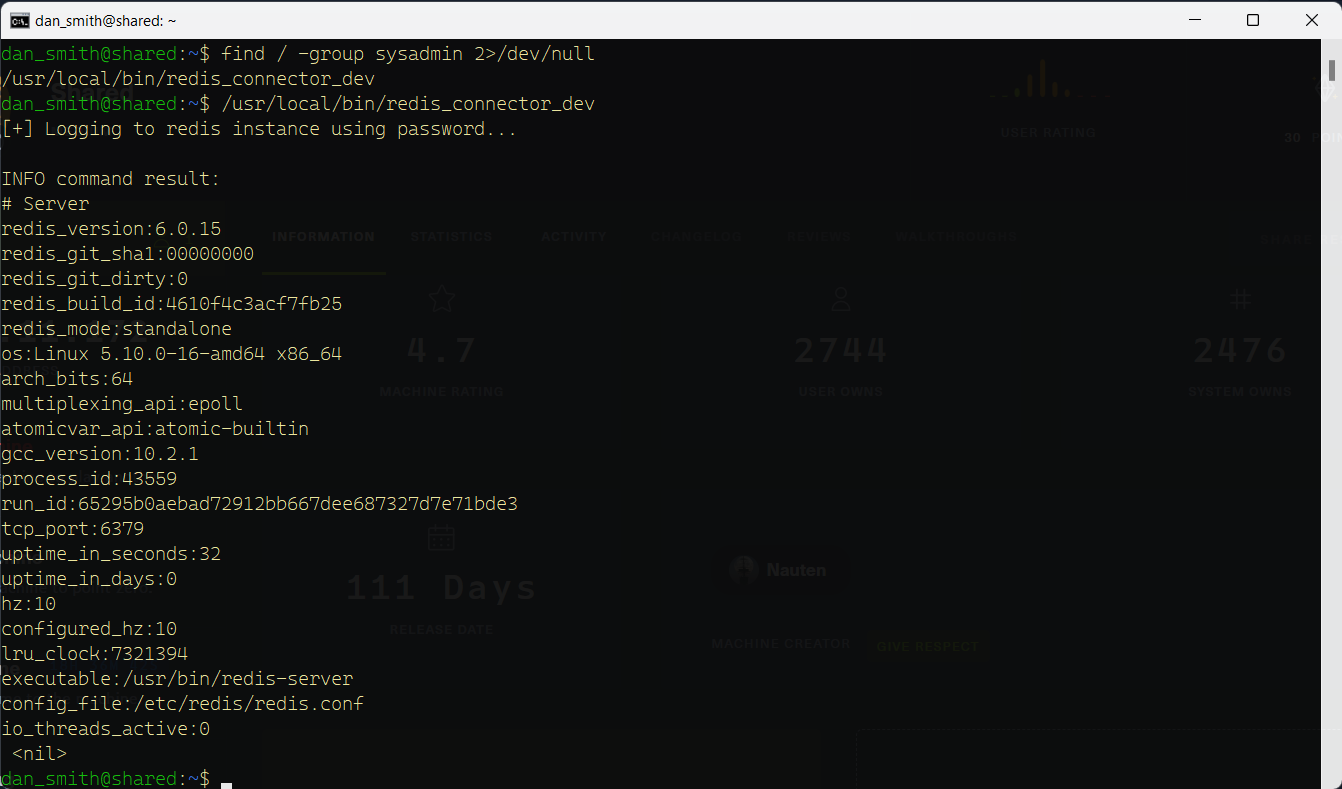


1. Now upload linpeas.sh here:



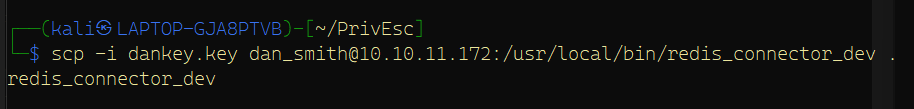
After executing linpeas, you’ll see:

Redis connector program belonging to root, that we also can access. Interesting, let’s execute it:



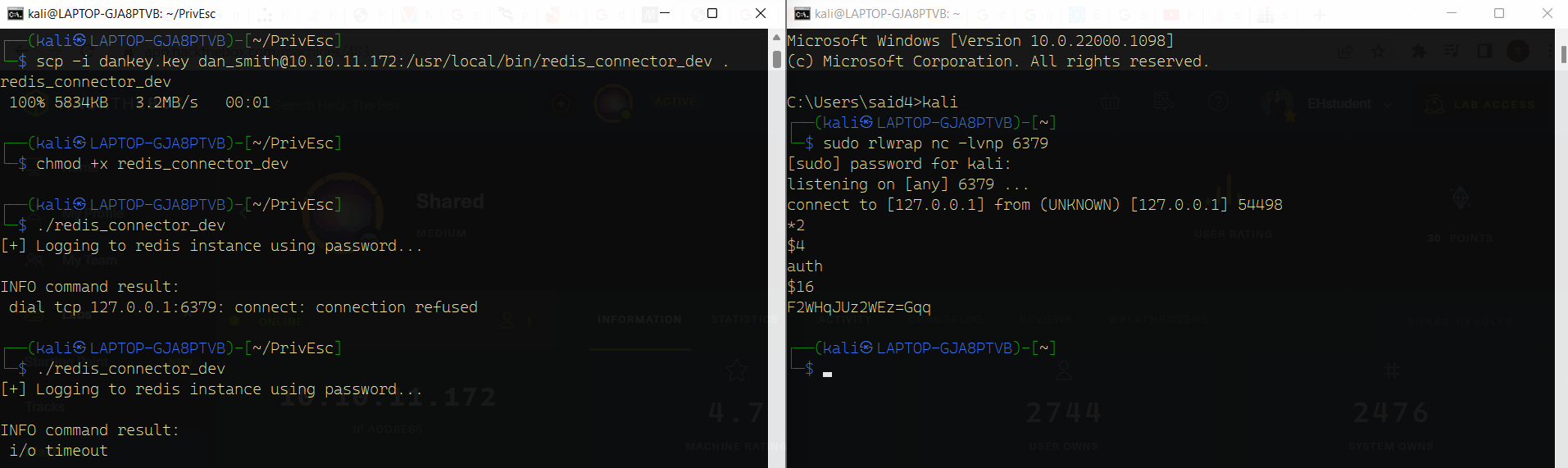
So first of all it connects to redis instance using the right password (so the program contains the redis password), and then it executes the INFO command.

Lets first of all get the program on our machine.

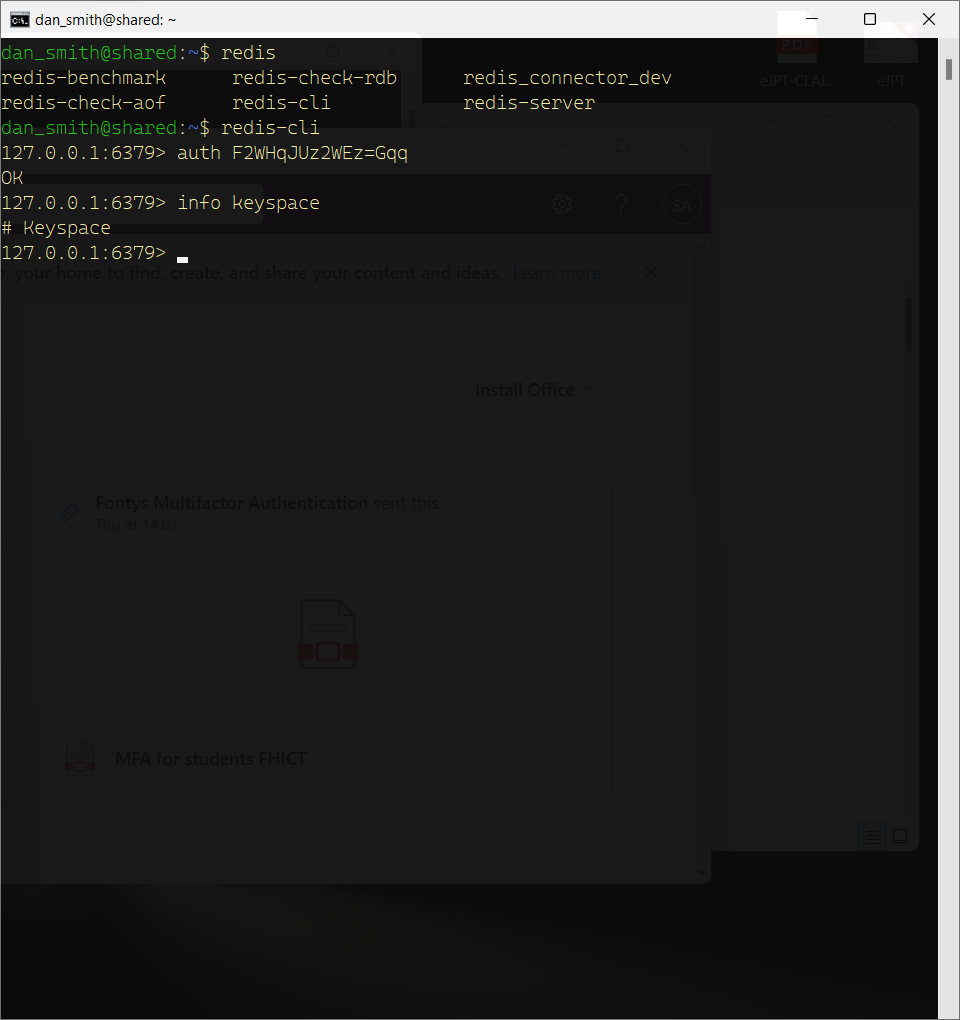


So we can try to get the password with strings (strings redis\_connector\_dev) (prints program in strings): nothing but gibberish.

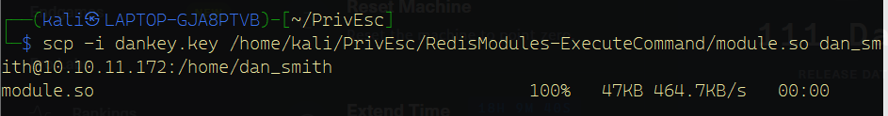
1. What we also can do, is run the program and start a listener on the port the program sends all info to, which redis, which is port 6379.

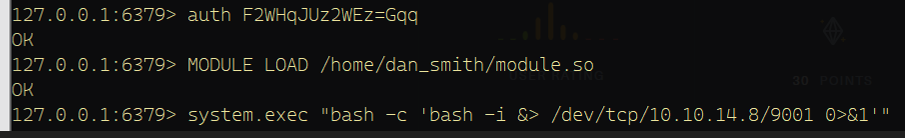


It worked and we’ve got the password.

1. Now we can go back to dan smiths machine, and start redis\_cli
2. In order to execute bash commands via redis, we need to use this [repository](https://github.com/n0b0dyCN/RedisModules-ExecuteCommand).

Git clone, cd, make

1. Upload module.so

1. Start a listener on your own machine
2. Import module in redis and start a reverse shell
3. grab root flag

