**LAB 4: TCP IP ATTACK**

Contents

[Environment Setup 3](#_Toc118056952)

[Dockers **Error! Bookmark not defined.**](#_Toc118056953)

[Task 1 3](#_Toc118056954)

[Task 1.1 6](#_Toc118056955)

[Task 1.2 10](#_Toc118056956)

[Task 1.3 10](#_Toc118056957)

[Task 2 12](#_Toc118056958)

[Manual Attack 12](#_Toc118056959)

[Automatic Attack 16](#_Toc118056960)

[Task 3 18](#_Toc118056961)

[Manual Attack 18](#_Toc118056962)

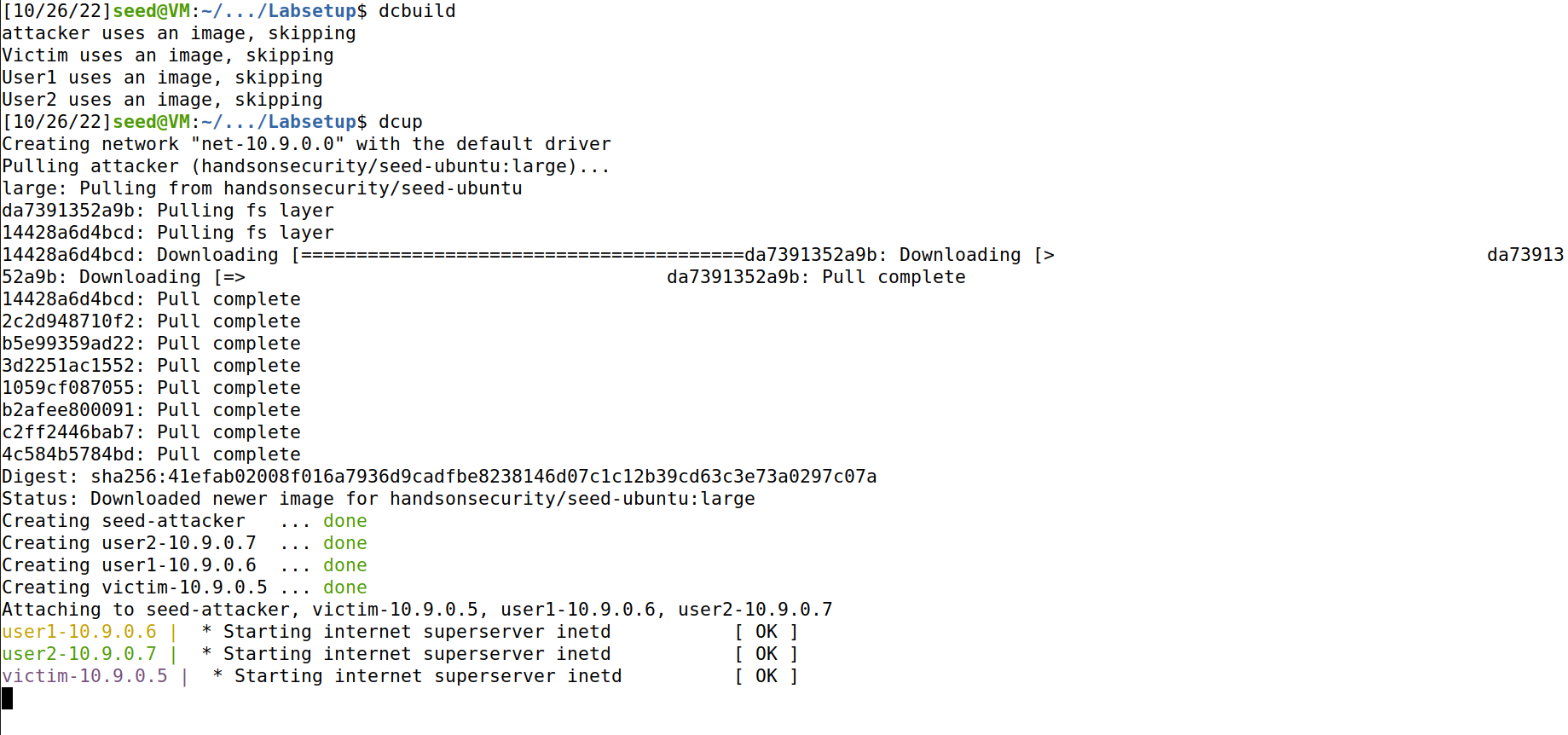
[Automatic Attack 23](#_Toc118056963)

[sTask 4 28](#_Toc118056964)

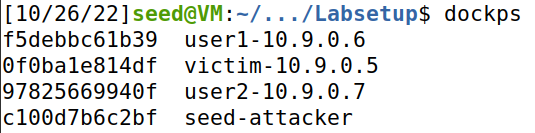
# Environment Setup

# Task 1

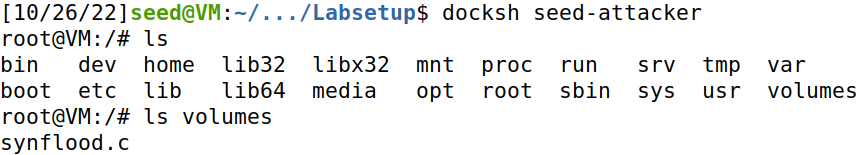
* Setting up Containers by opening new terminals while setting up the users displayed below including the attacker, victim and users.



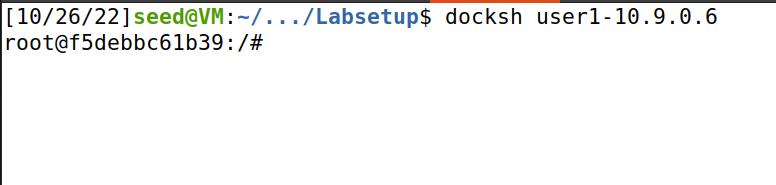
* Checking Dockers Available



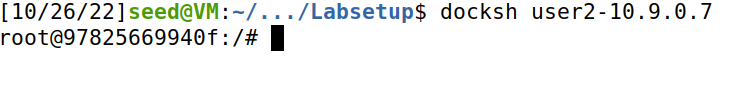
* Attacker Docker



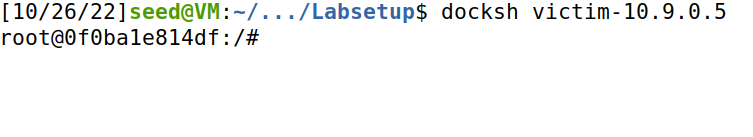
* User1 Docker



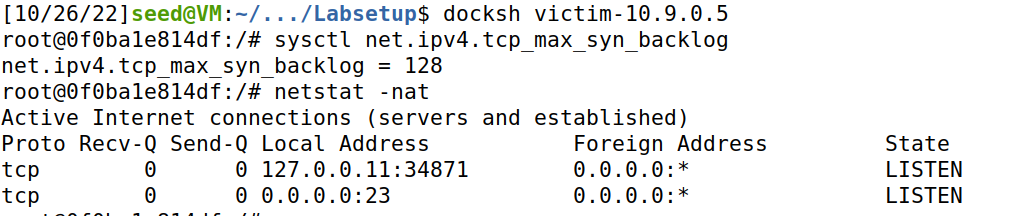
* User2 Docker



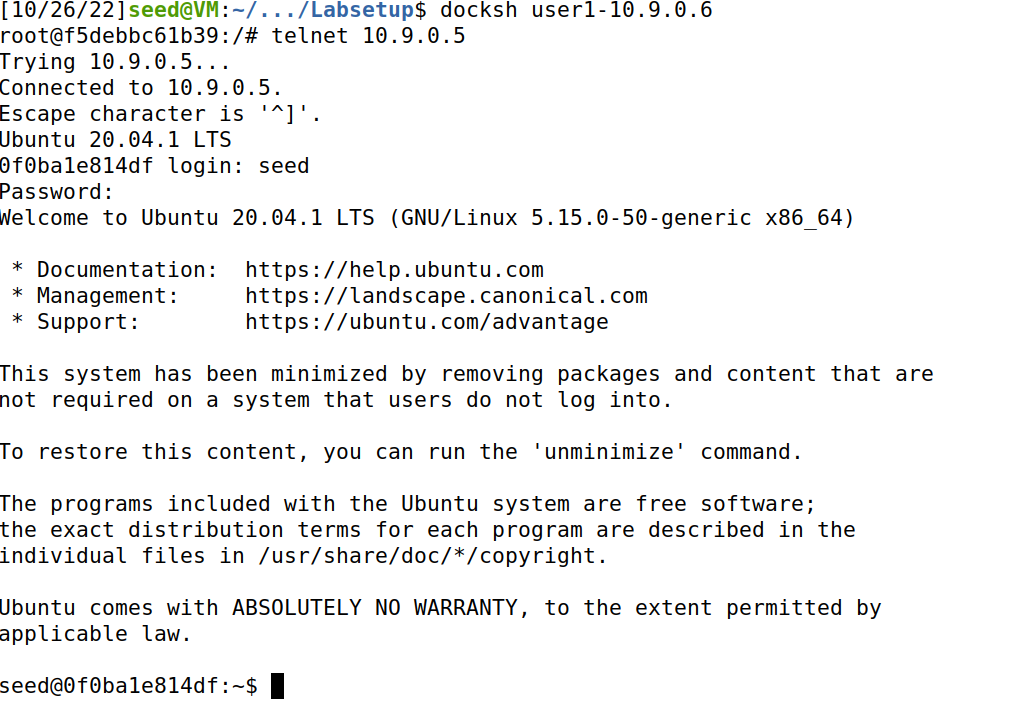
* Victim Docker



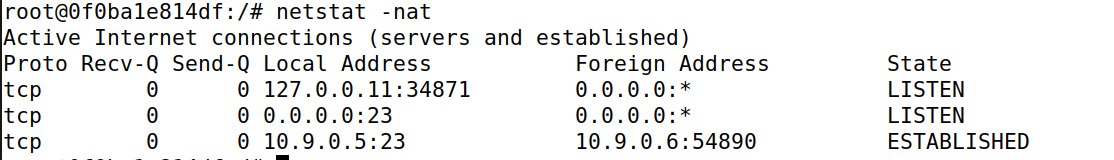
* Checking the size of the Backlog and the services available. Here we can see telnet service available for use which is with the following address”*0.0.0.0::23”.*



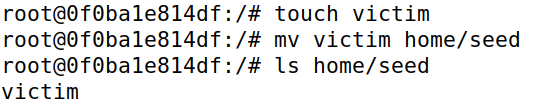
* Setting-up TCP connection between **user1** with **victim**



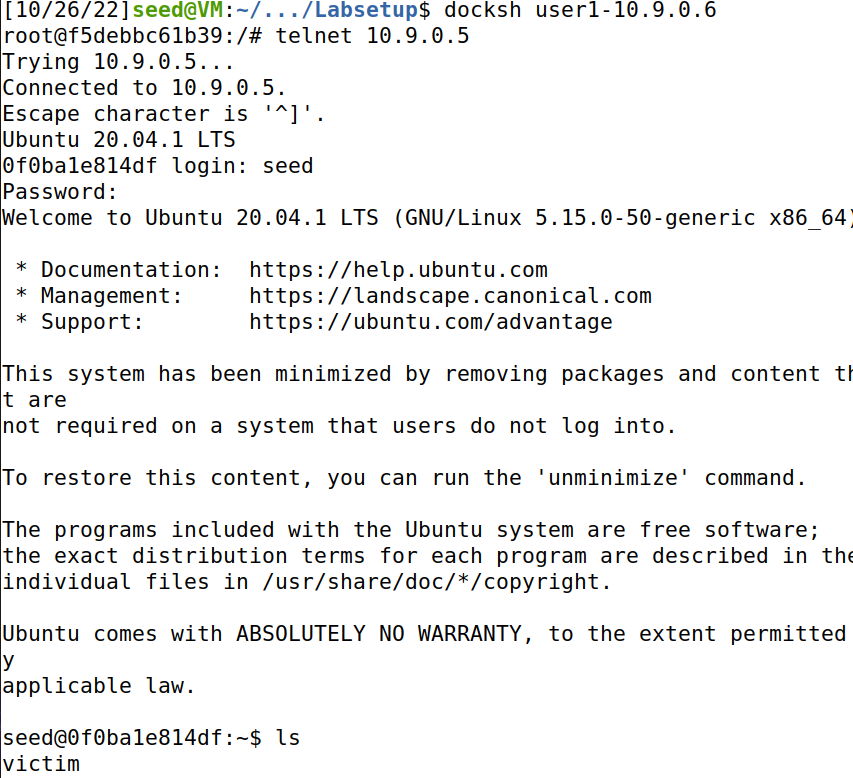
* We can clearly see that the connection has been established in the screenshot below



* To verify the connection let’s make a file named victim on victim’s machine and store in the address shown below

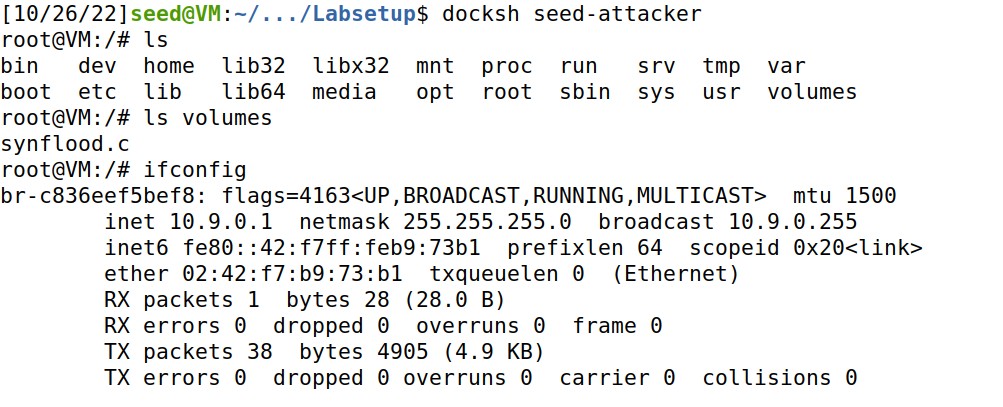


* Moving on checking in **user1**’s machine we can clearly see the file present there.

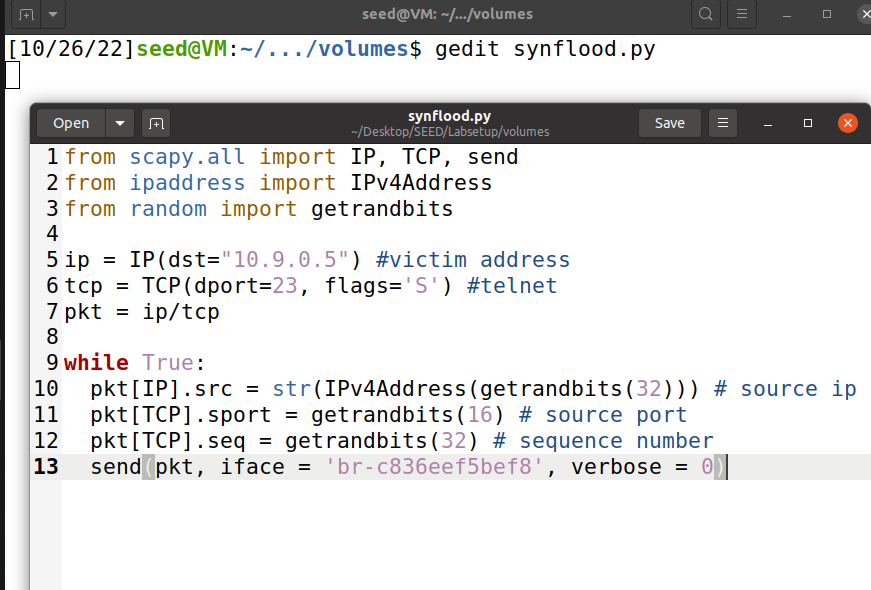


## Task 1.1

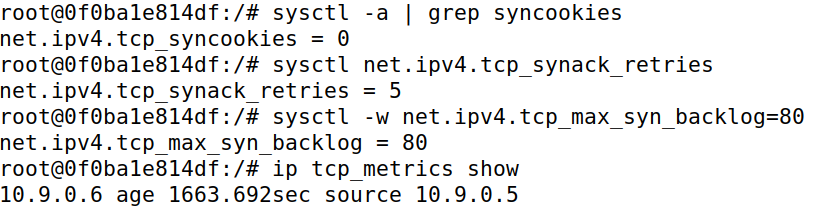
* Checking the Attacker Interface



* Created a file named *synflood.py* which is packet containing victim ID, service to target and attacker’s interface



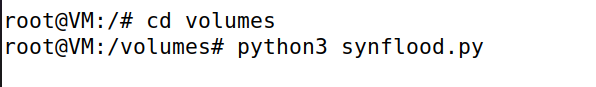
* Checking syncookies and retries while modifying backlog size on **victim**’s machine, in order to increase the chances of attack’s success.



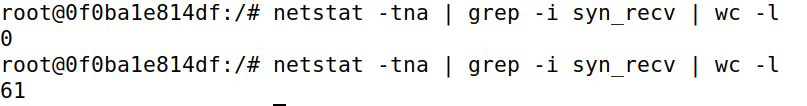
* Before launching attack let’s check how many syn packets did the **victim** received.

****

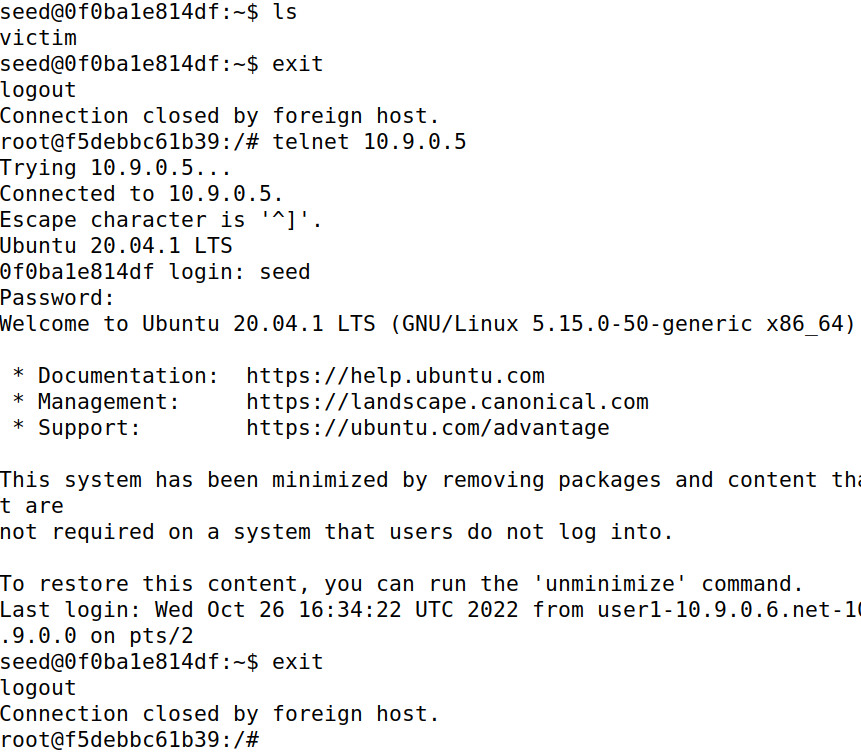
* Launching the Attack



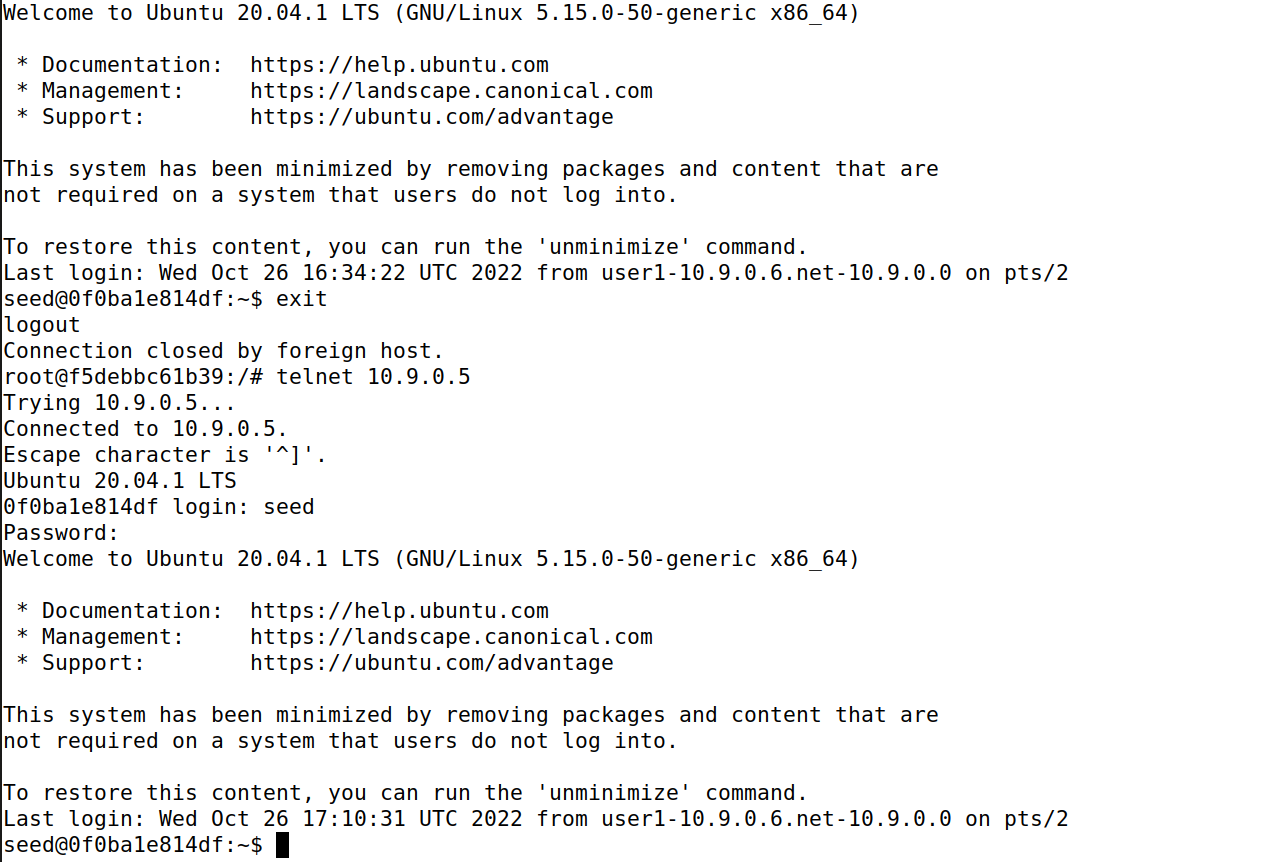
* After the Attack Victim Received 61 Packets



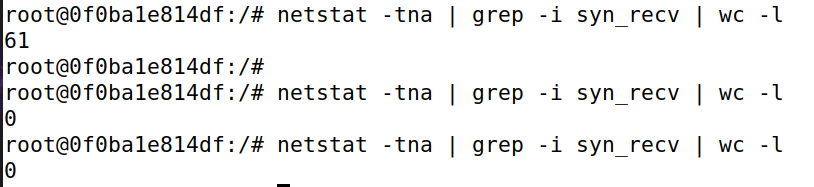
* Now while trying to disconnect **user1** which is connected to the **victim**’s machine which we are allowed to disconnect.



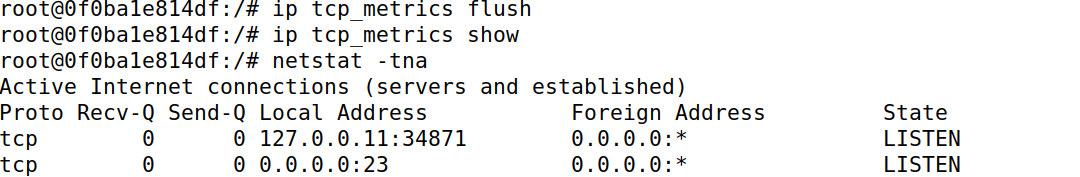
* Now I launched the attack again and tried logging out and reconnecting which I was successfully able to do but it means the attack failed.



* After stopping the Attack the syn packets received went back to normal after a little while



* In order to make the attack a success lets flush the memory and try again

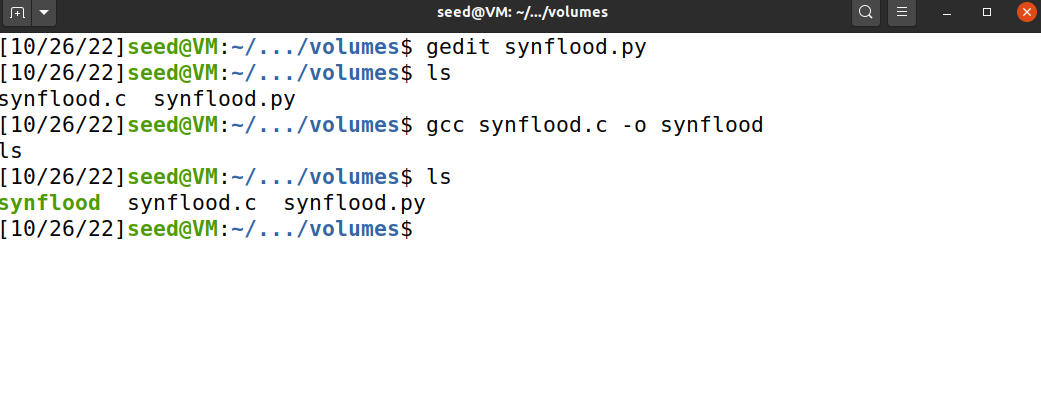


* After running the attack again I found out that the attack didn’t allow the **user** to connect to the **victim**/

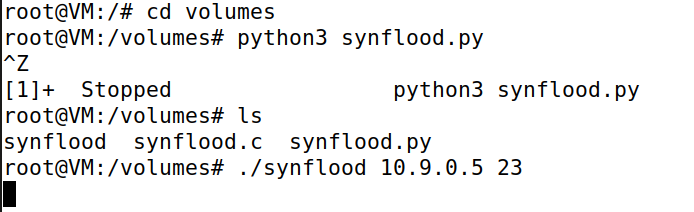


## Task 1.2

* Now moving on with the code in C file provided in the lab setup. We first made an executable by compiling the C file.



* Now Launching the Attack



* Packets being received by the Victim



* This time the attack worked without any problem as I couldn’t connect the **user1** to the **victim** machine



## Task 1.3

* Now placing syncookie countermeasure



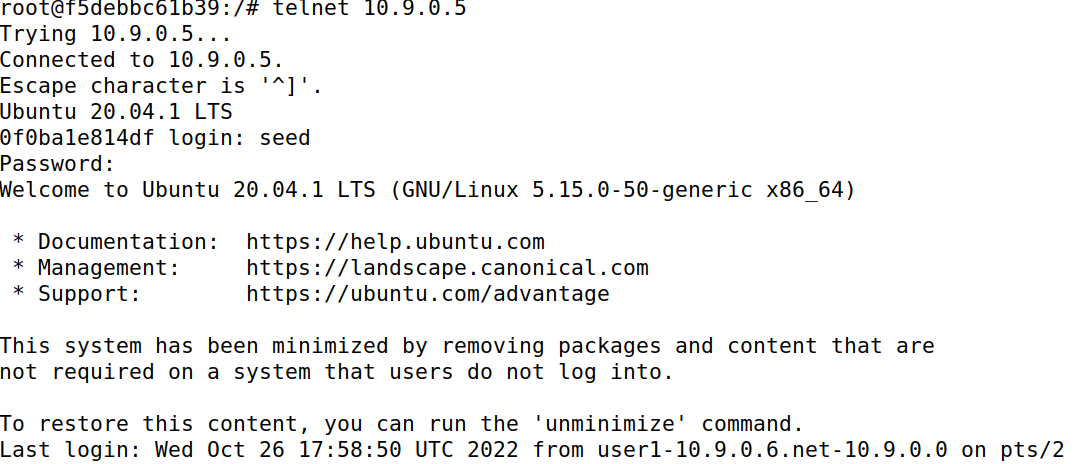
* Launching the attack with python code



* After launching the attack with python file the number of packets received have jumped to 128 while the backlog remains to 80 size



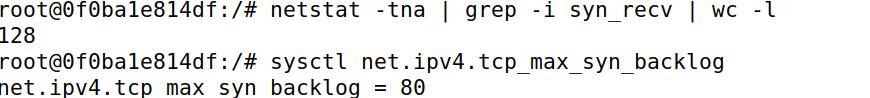
* I was successfully able to login which means countermeasures worked and the attack failed



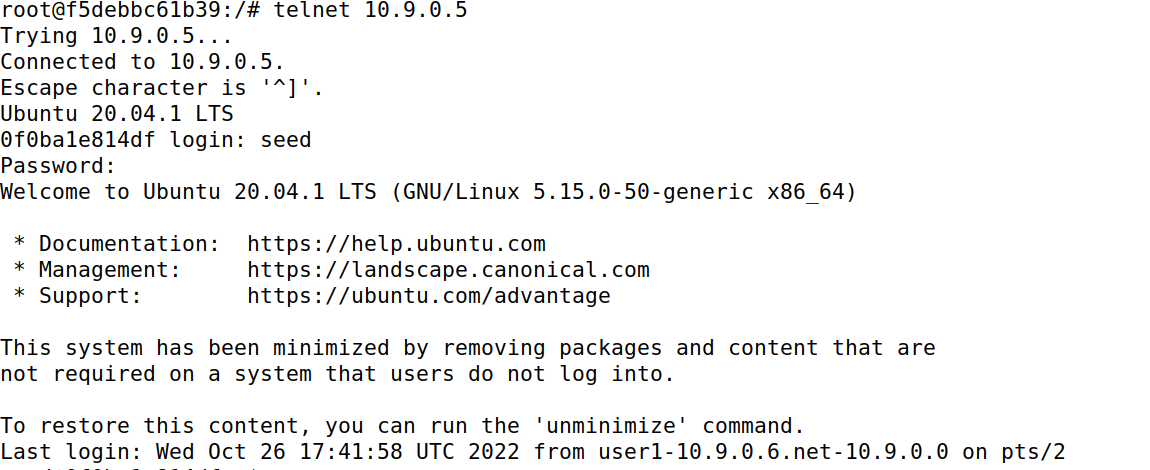
* Launching Attack with C File



* After launching the attack with C file the number of packets received have jumped to 128 while the backlog size remains 80



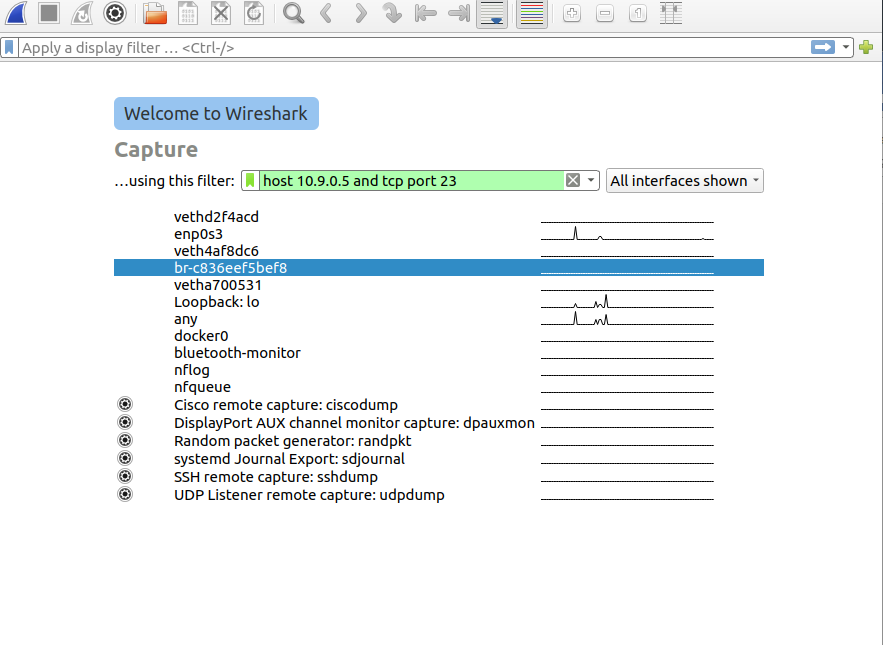
* Moreover the attack failed with countermeasures in place.



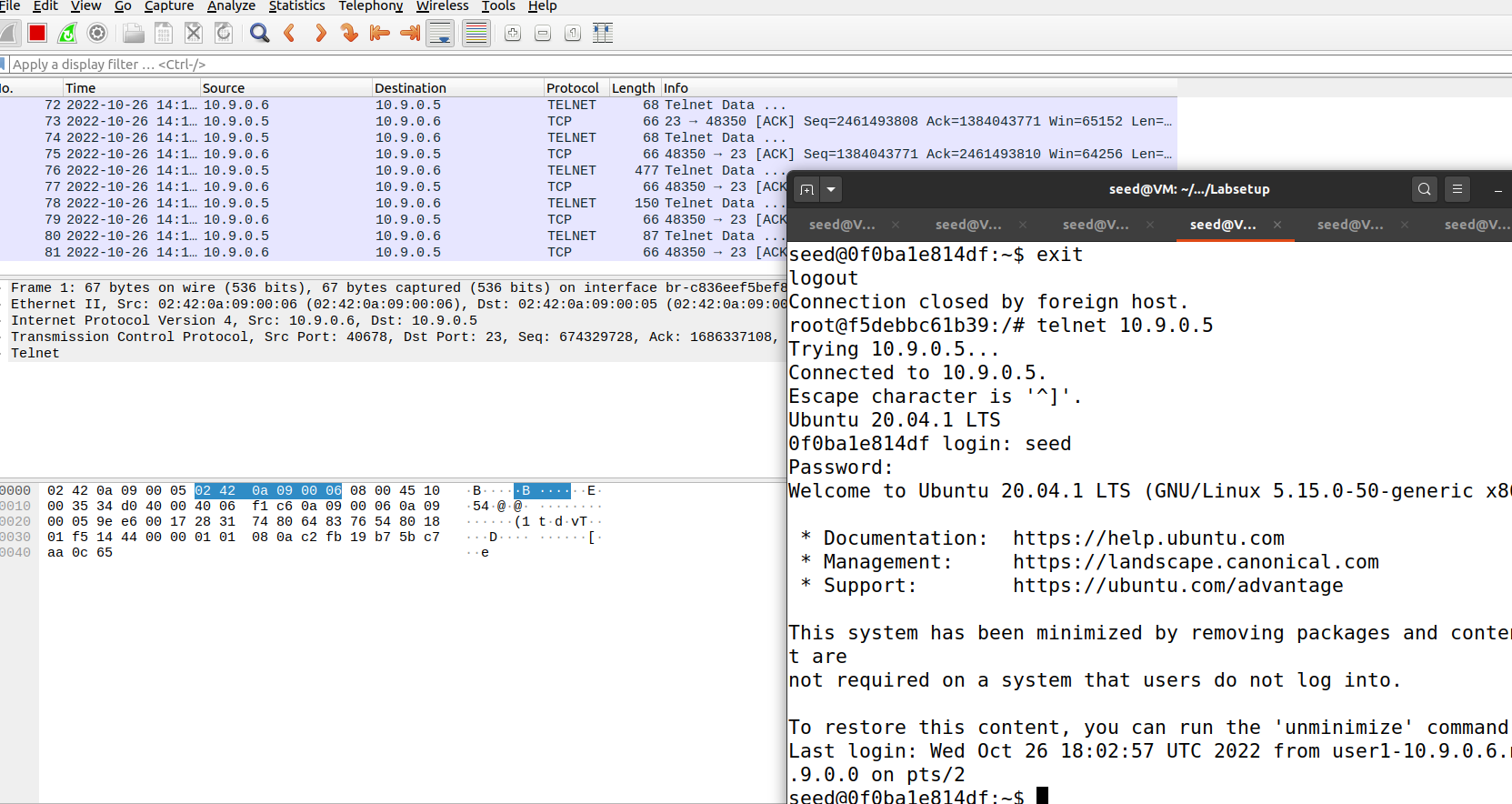
# Task 2

### Manual Attack

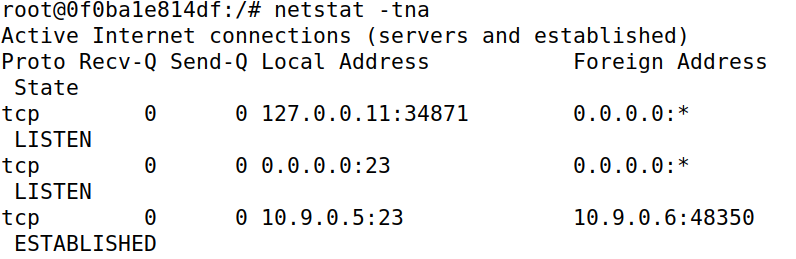
* Now we will open Wireshark and sniff packets from **Attacker**’s address while targeting **Victim**’s machine and telnet port



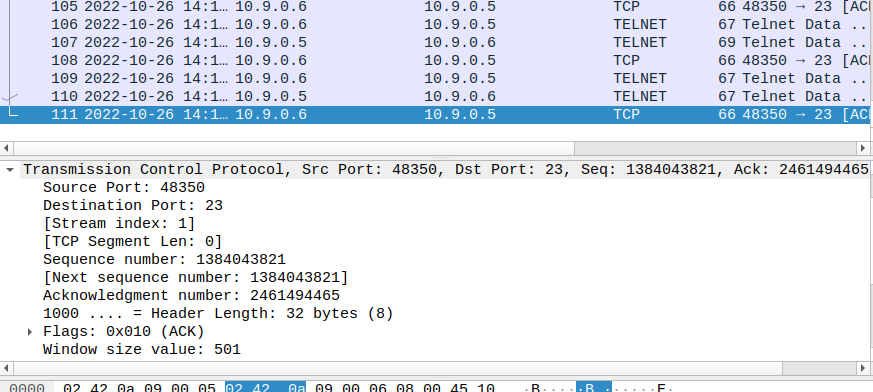
* Now while logging into **victim**’s machine as **user1**, we can observe the packet flow in Wireshark.



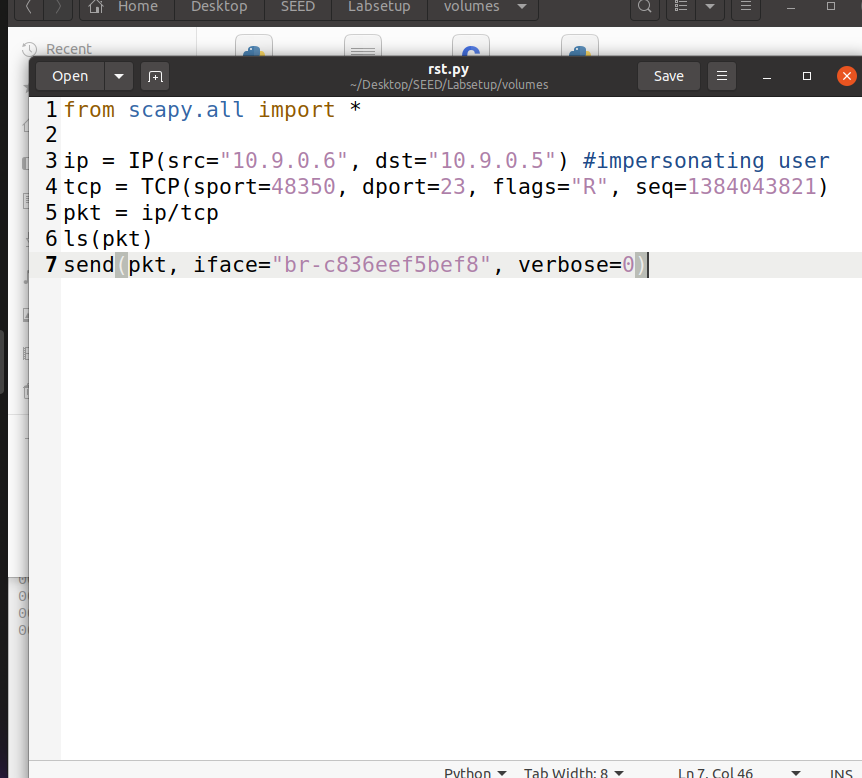
* And the connection is established between **user1** and **victim**.



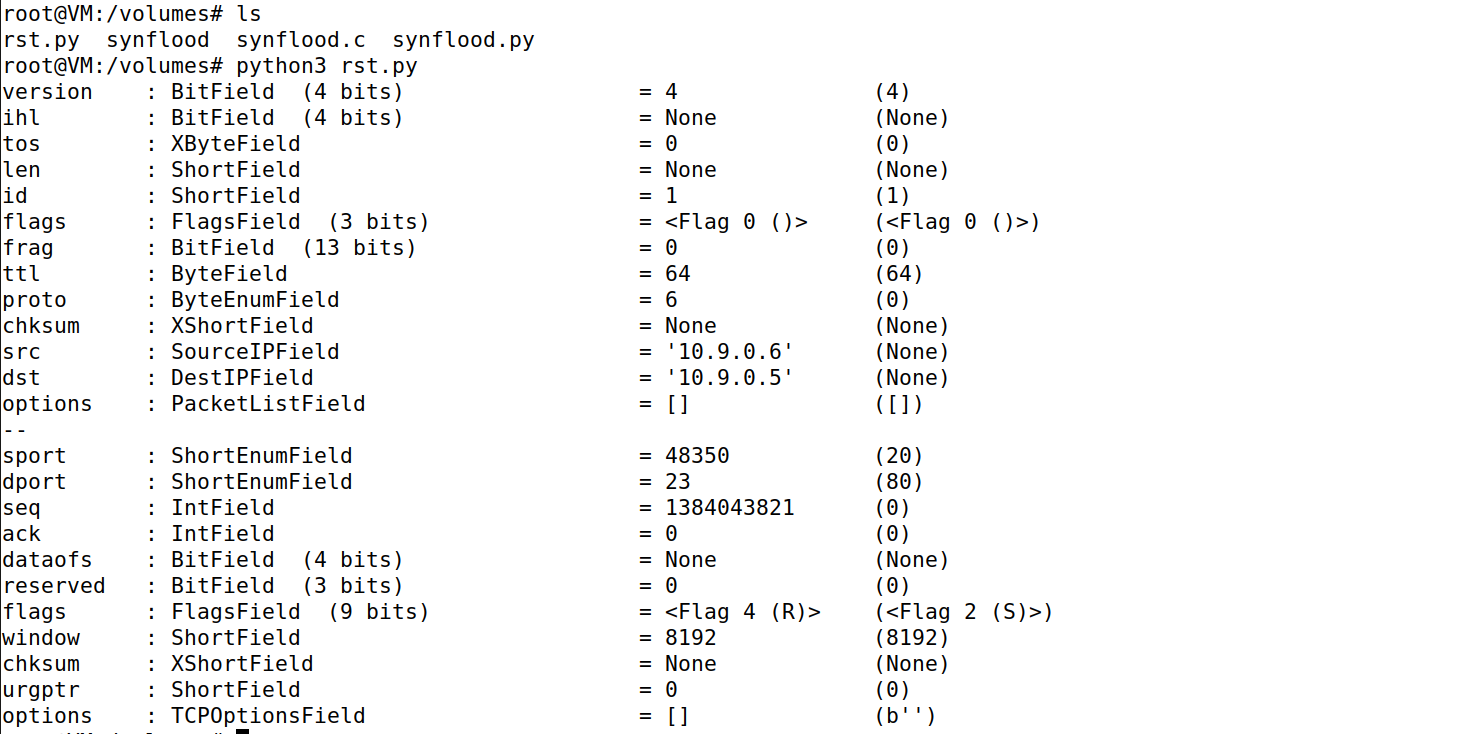
* From Wireshark I have acquired some useful information which will be used in the python script used like source and destination IP and Port, respectively, for the RST attack. Moreover the next sequence number is also an essential part to be used.



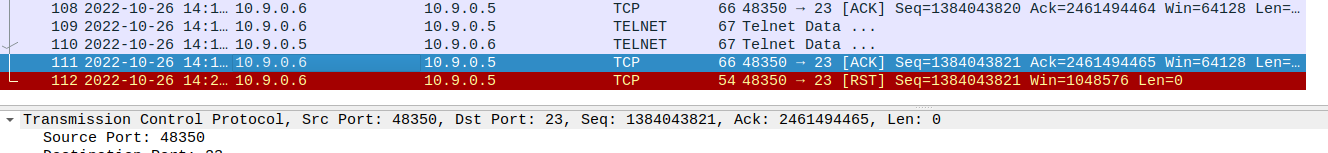
* Now we will edit the provided script to our needs



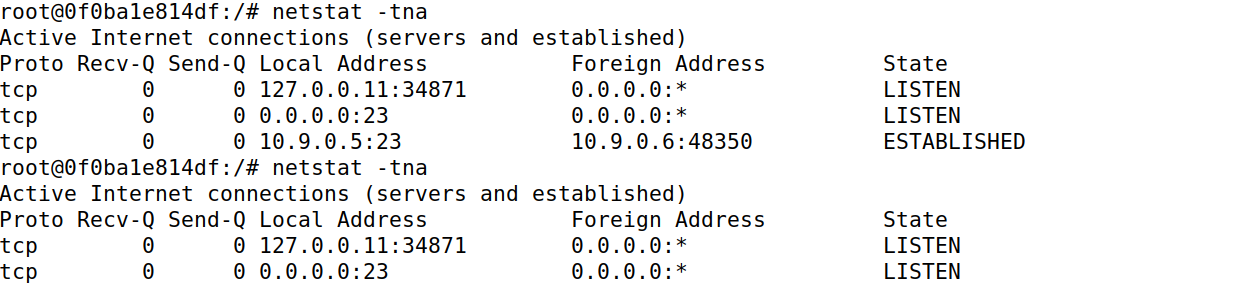
* Now I launched the RST telnet Attack from **Attacker**’s machine



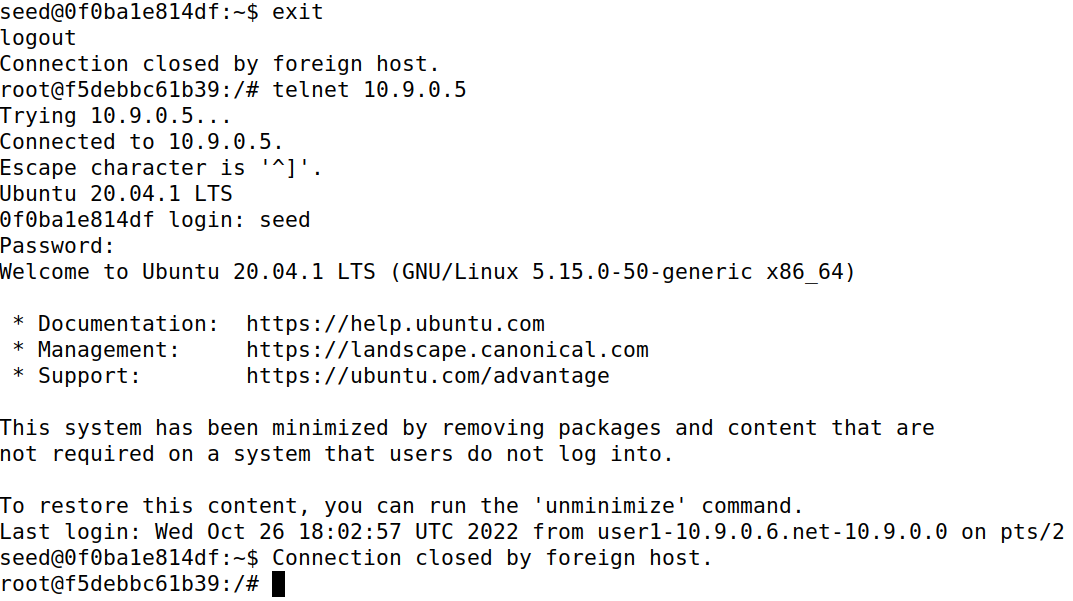
* Here we caught the RST packet in the Wireshark



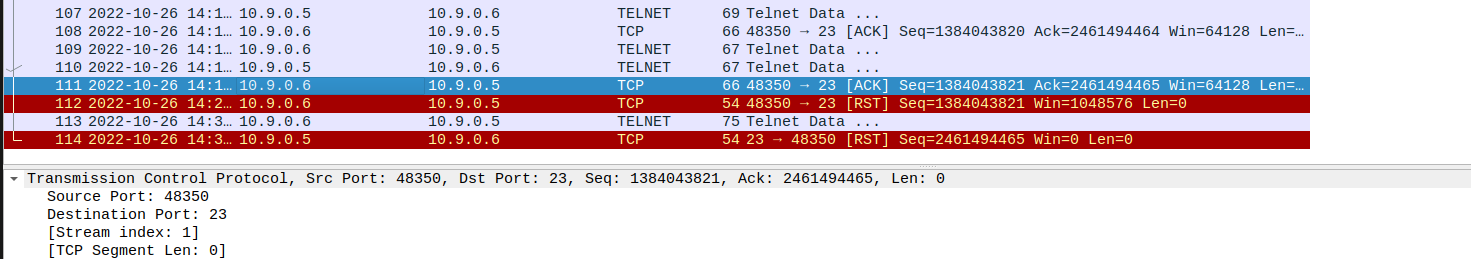
* More importantly the connection of **user1** with **victim** has been broken.



* Even on **user1** terminal we can see the connection broke



* And when I came back to Wireshark I noticed the RST packets have also done their work and have been caught here.

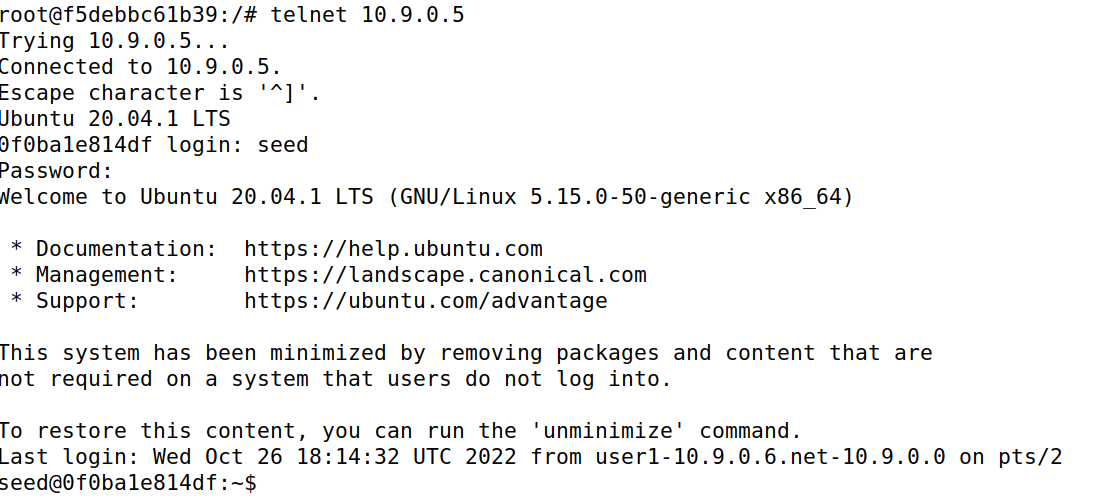


### Automatic Attack

* Now creating a new python file to launch the code automatically with the help of a script where we have added **Attacker**’s interface and filter of TCP and Telnet Port.



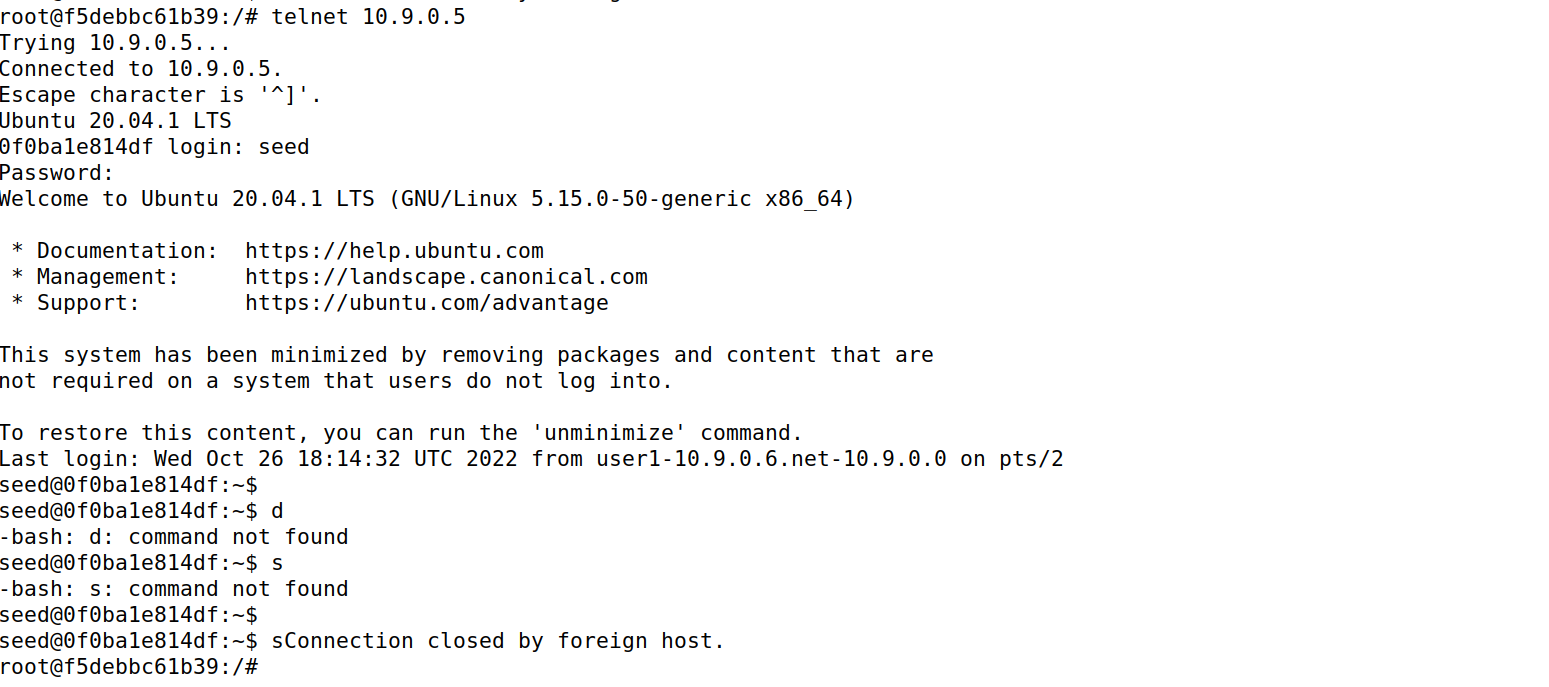
* Reconnecting **user1** with **victim**



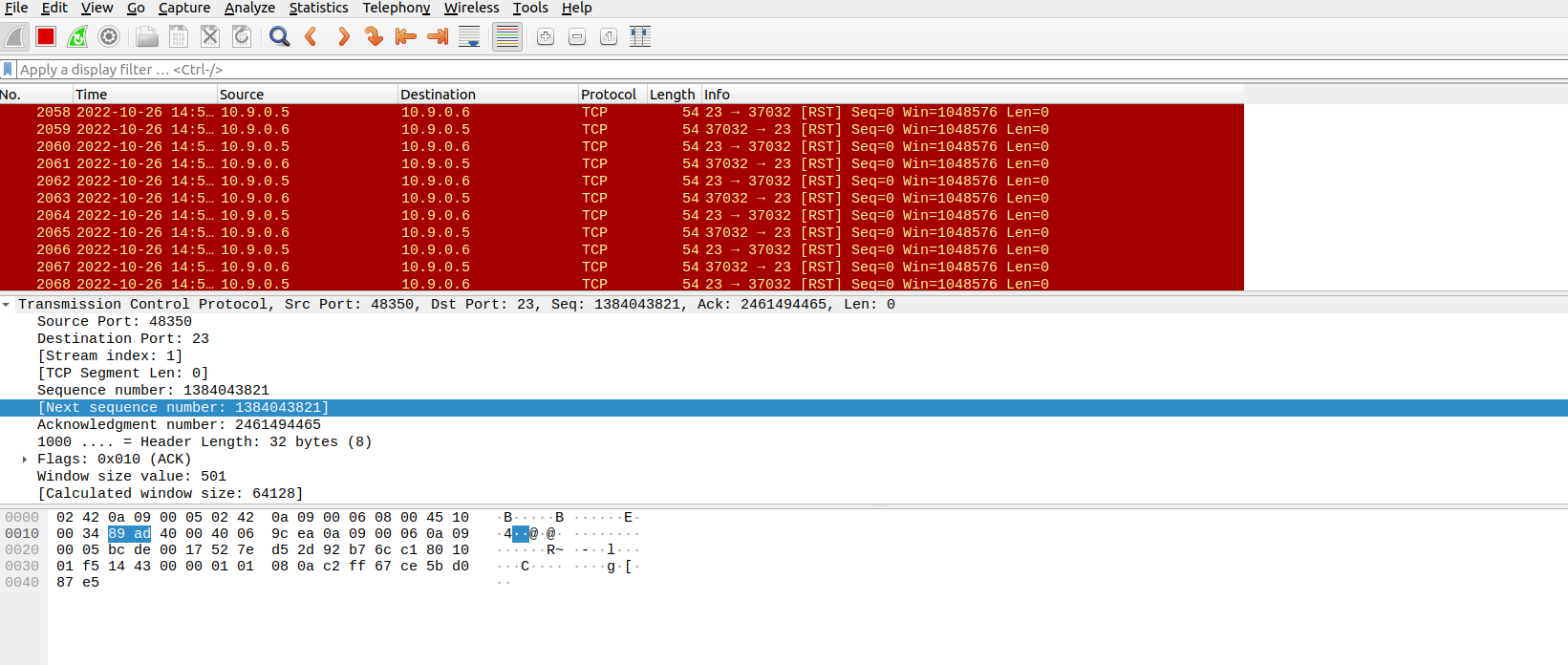
* Launching the Attack



* Now we can see the **user1** again disconnected



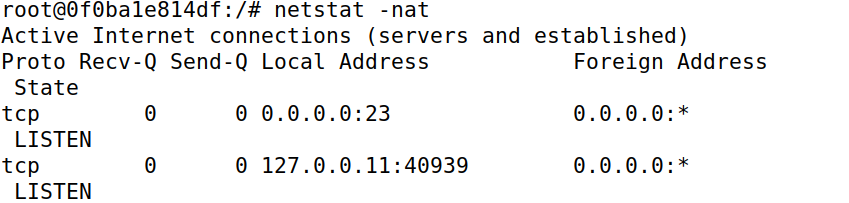
* And the packets coming automatically being captured in Wireshark



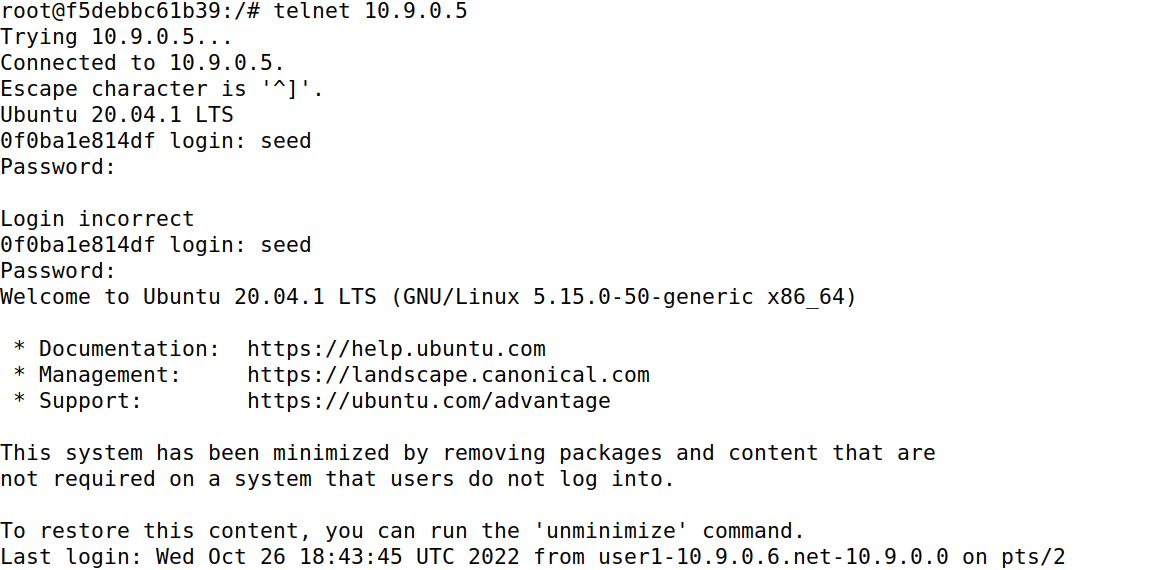
# Task 3

### Manual Attack

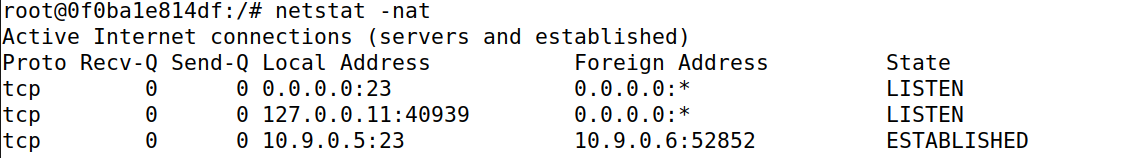
* Here, checking the connections if established with the **victim**’s machine.



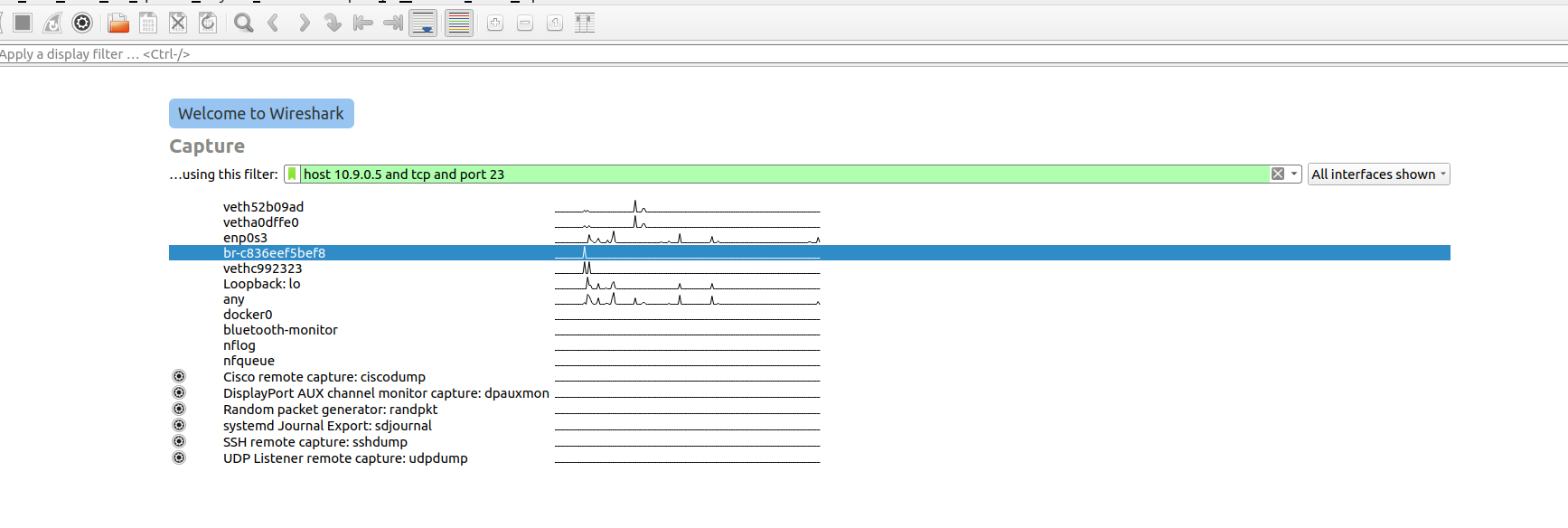
* Connecting **user1** with **victim**’s machine



* Confirming if the connection is established and as displayed in the screenshot below we can see it is established



* Setting up Wireshark for Packet Analysis with a filter which is looking for packet traffic around **victim**’s machine on the telnet port.



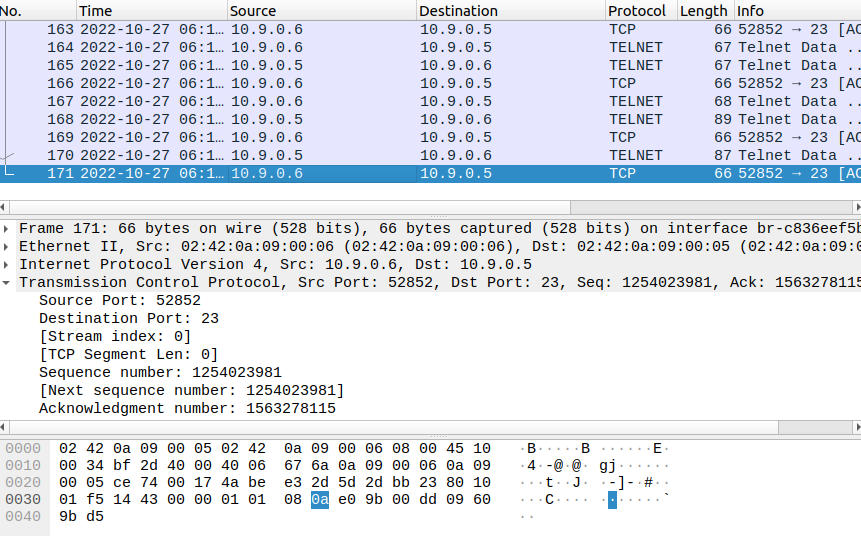
* Now typing a command on **user1**’s machine in order to receive some packets in Wireshark to extract required details.



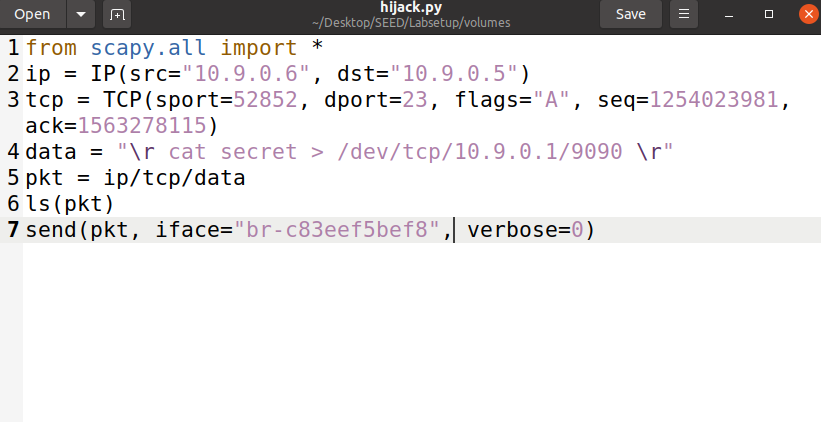
* Obtaining **Attacker**’s IP in order to edit the provided python script in the manual.



* Now we have gained the required information including the port connecting with Telnet Port, next Sequence Number and Acknowledgement Number.



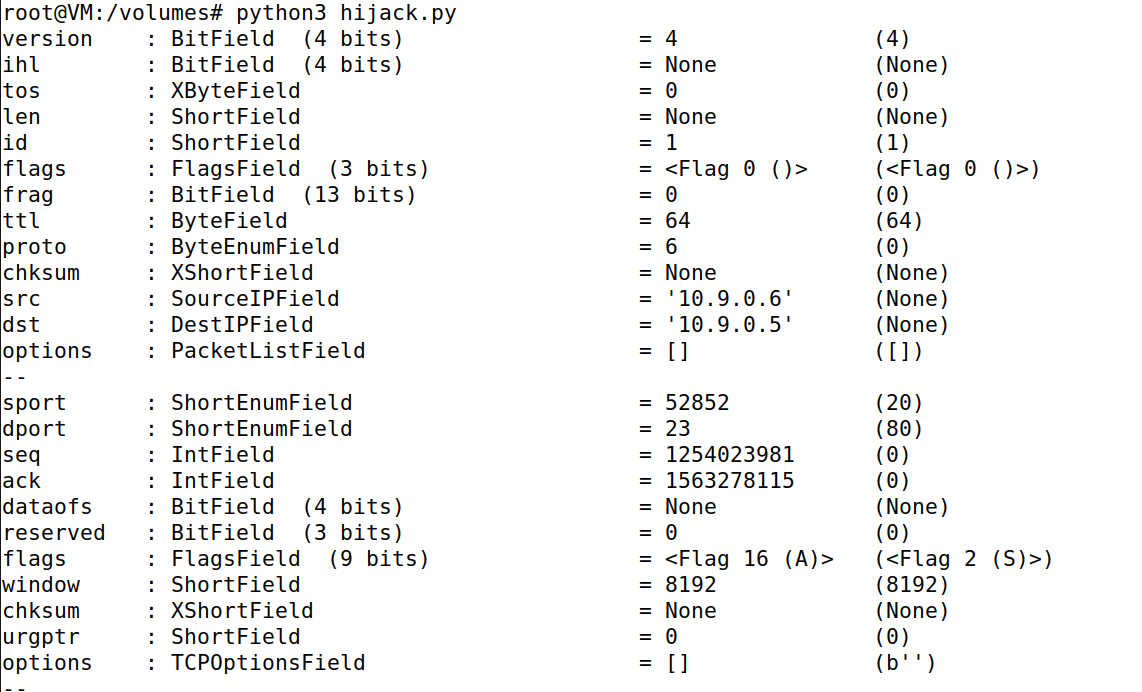
* Now using the above data I have edited the python script.



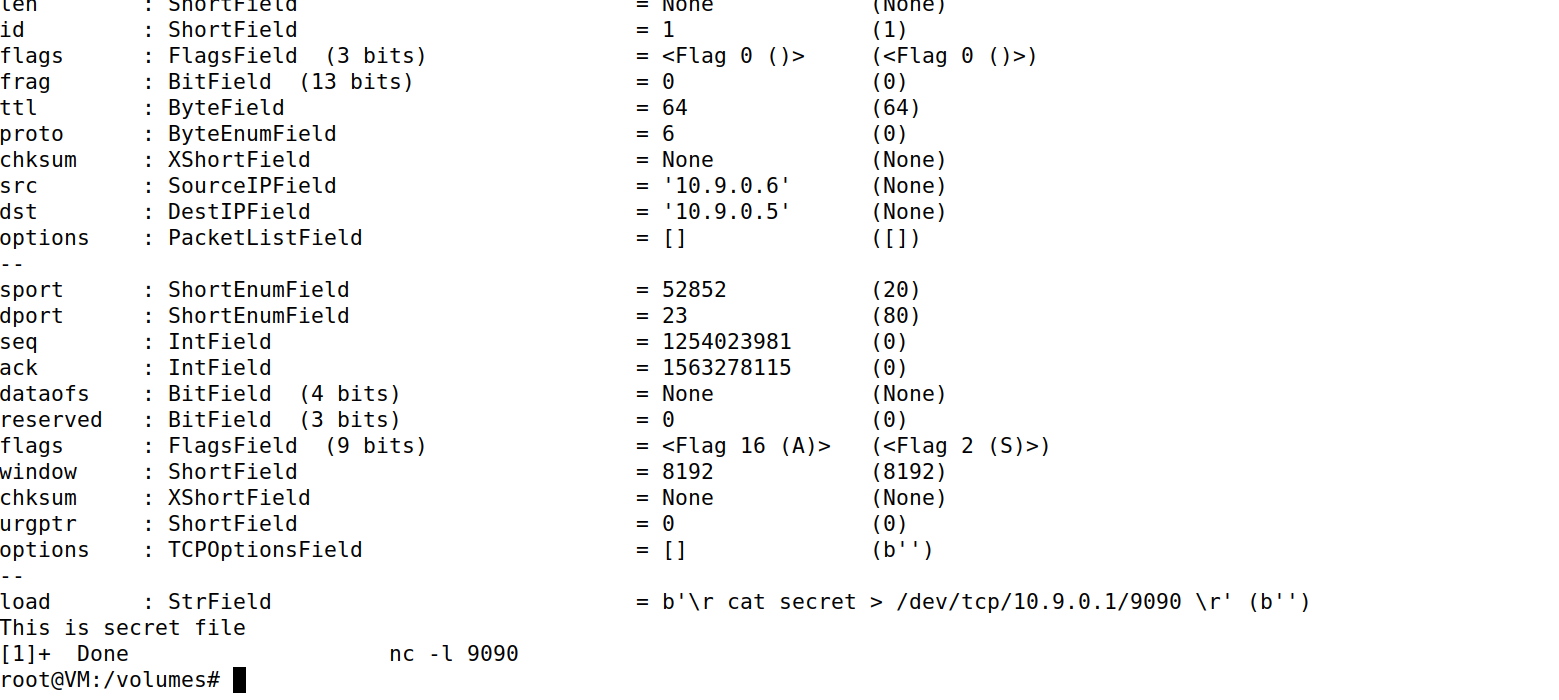
* Before we launch the attack we will run the command to listen to the port 9090 and put the process in background in order to launch the attack



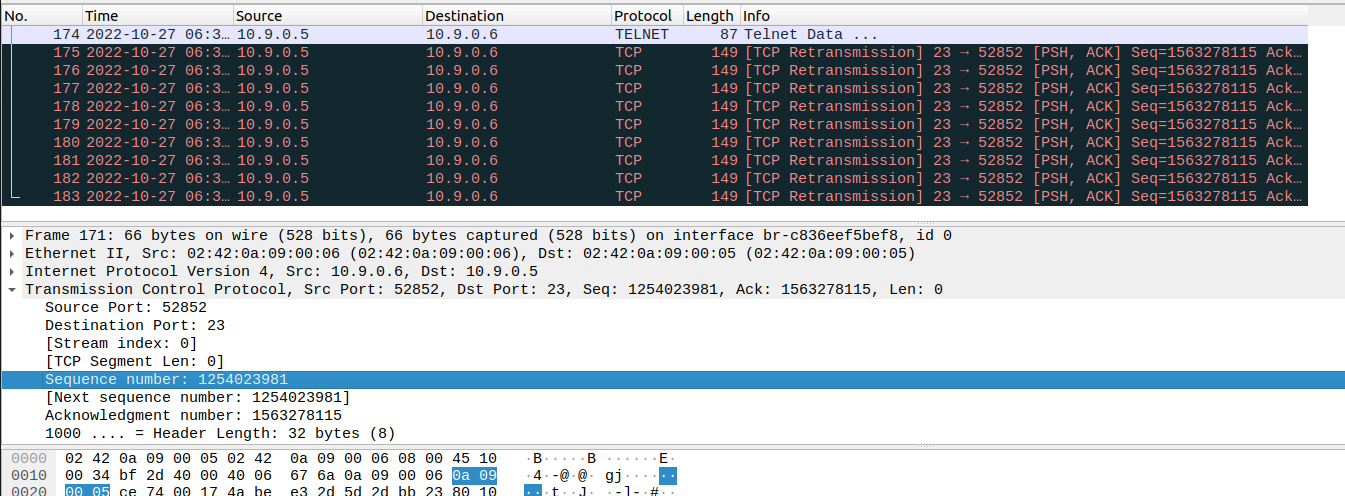
* Now I launched the attack from the python script



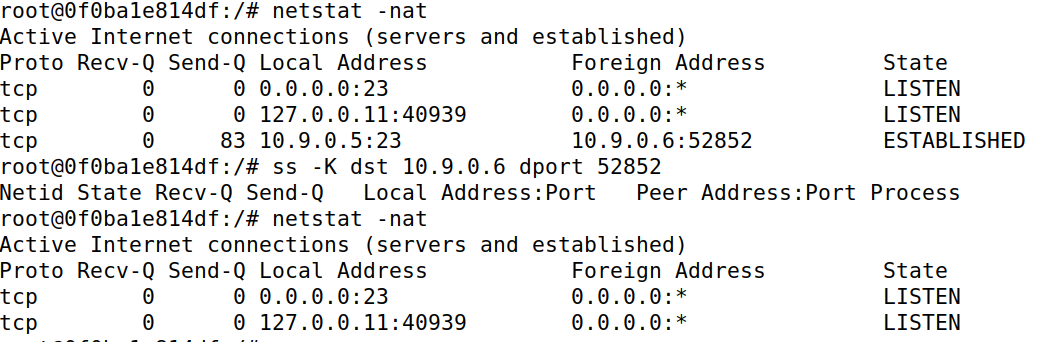
* And in the process we found the secret we were looking for as mentioned in the script.



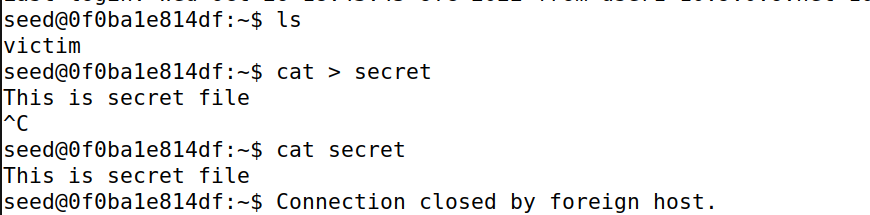
* Now on Wireshark we can observe Phishing Packets



* Now terminating the **user1** and **victim** connection from **victim**’s machine.

****

* Here we can notice the connection closed from **user1**’s terminal by a foreign host which is **victim**.

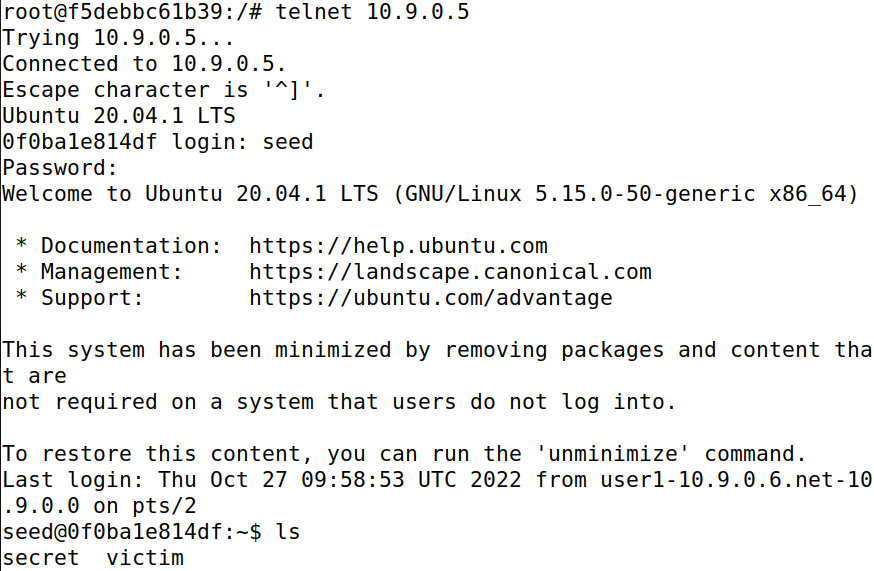


### Automatic Attack

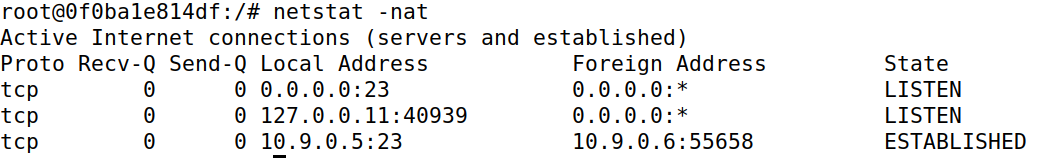
* I wrote another script involving contents from manual hijacking attack and the automatic reset attack. And modified them to requirements.



* Reconnecting **user1** with **victim**’s machine and verifying if the **user1**’s machine has the secret file.



* Link Established verified from **user1**’s terminal



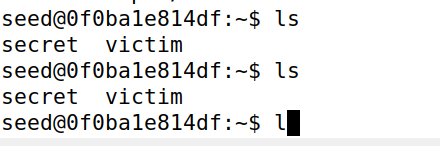
* Before we launch the attack we will run the command to listen to the port 9090 and put the process in background in order to launch the attack



* Launching the Attack



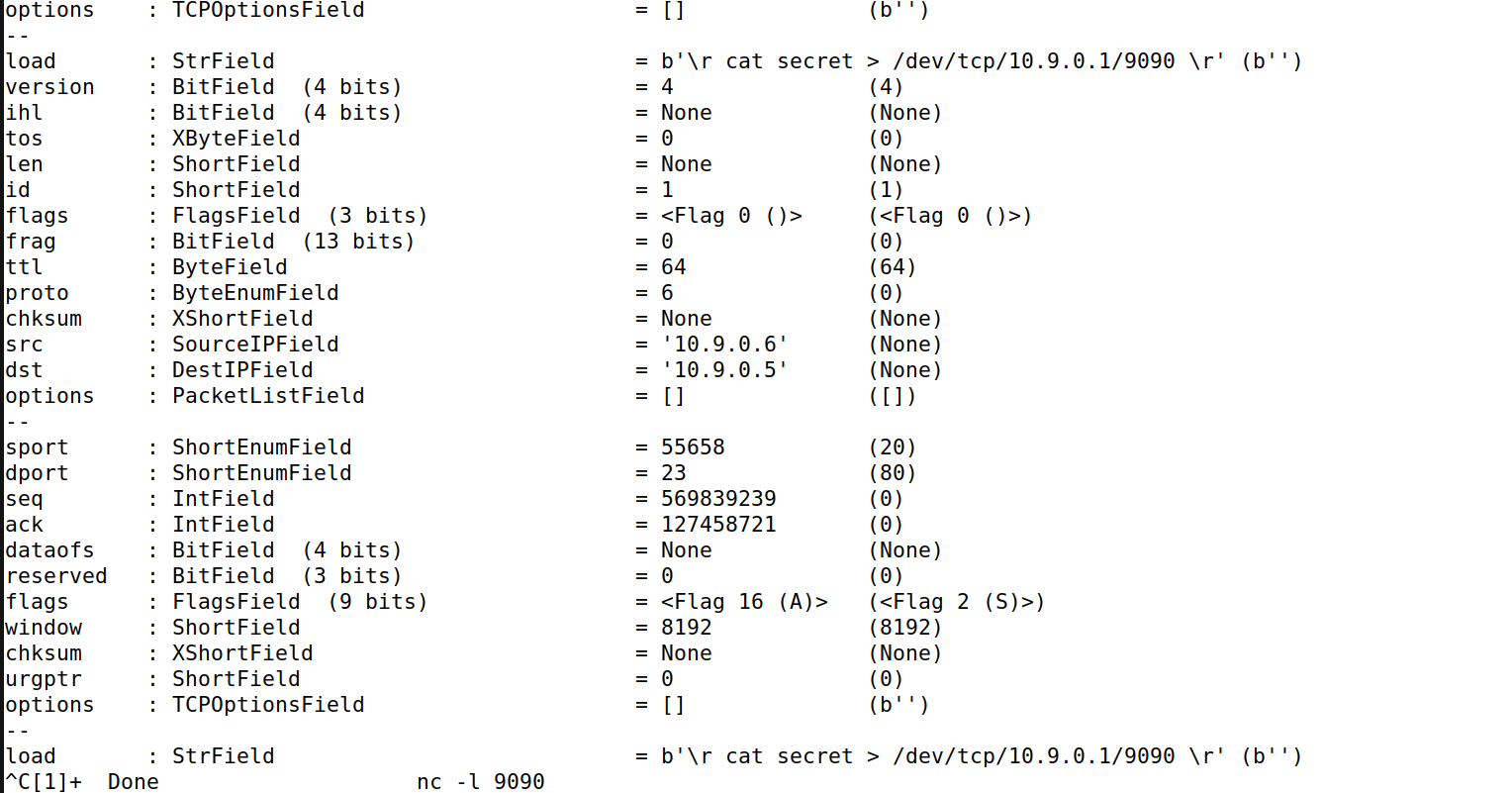
* After Launching the Attack I couldn’t type anything on **user1**’s terminal indicating that session has been hijacked.



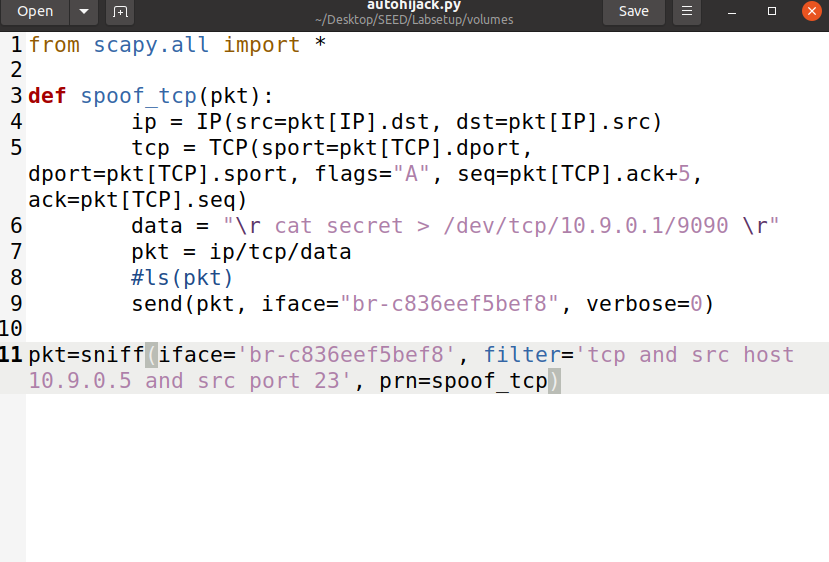
* It can be observed in Wireshark as well that the Phishing Packets have come in abundance and have continued to do so until the attack was stopped.



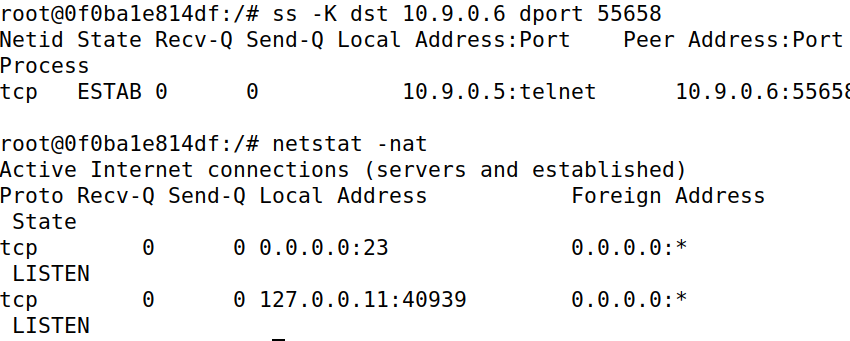
* It is a problem to find the secret in all this output from the attacks so.



* We will modify the code by stopping the sniffing when we obtain the secret. We have simply commented line 8 for that purpose.



* Killing the connection between **user1** and **victim**.



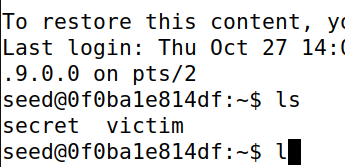
* Reconnecting **user1** with **victim**’s machine.



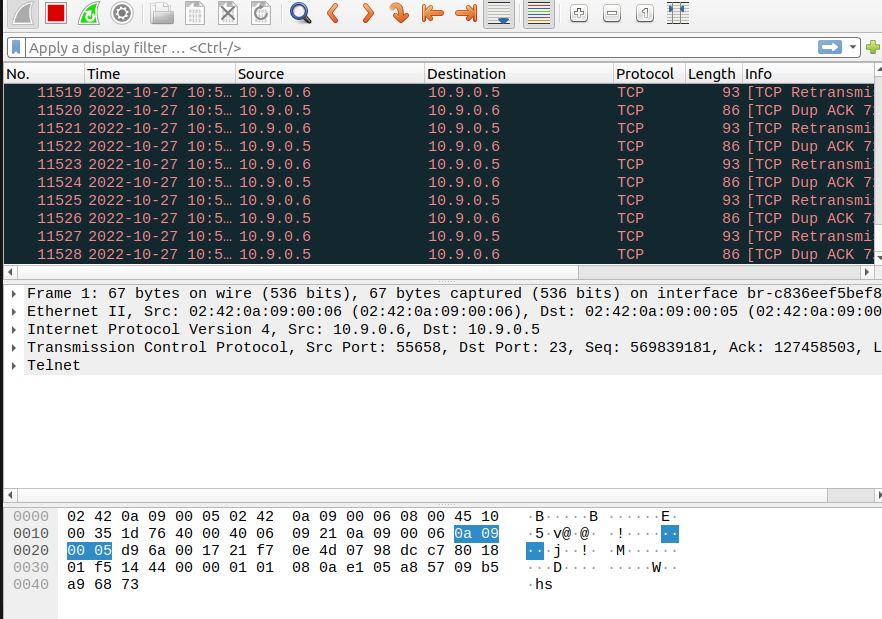
* Again launching the attack.



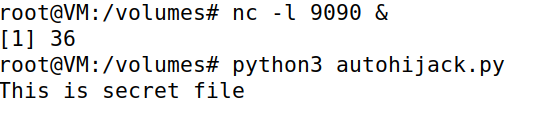
* Again we got stuck meaning the sniffing started and session is hijacked.



* While the traffic is moving in Wireshark

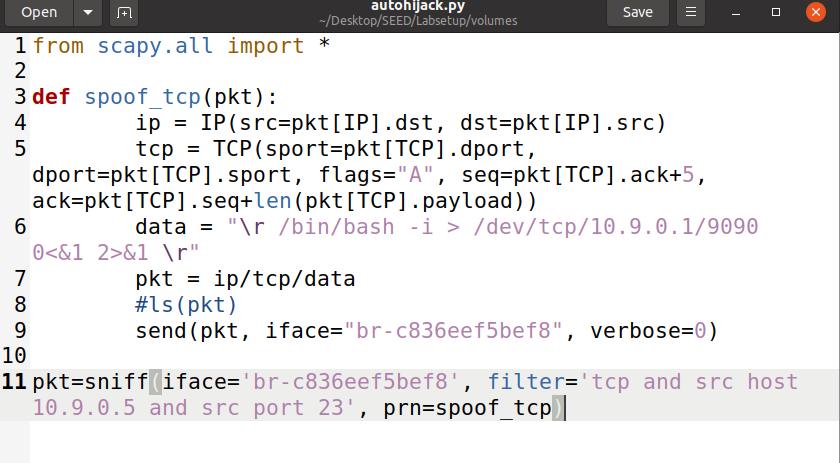


* And the **Attacker** found the secret

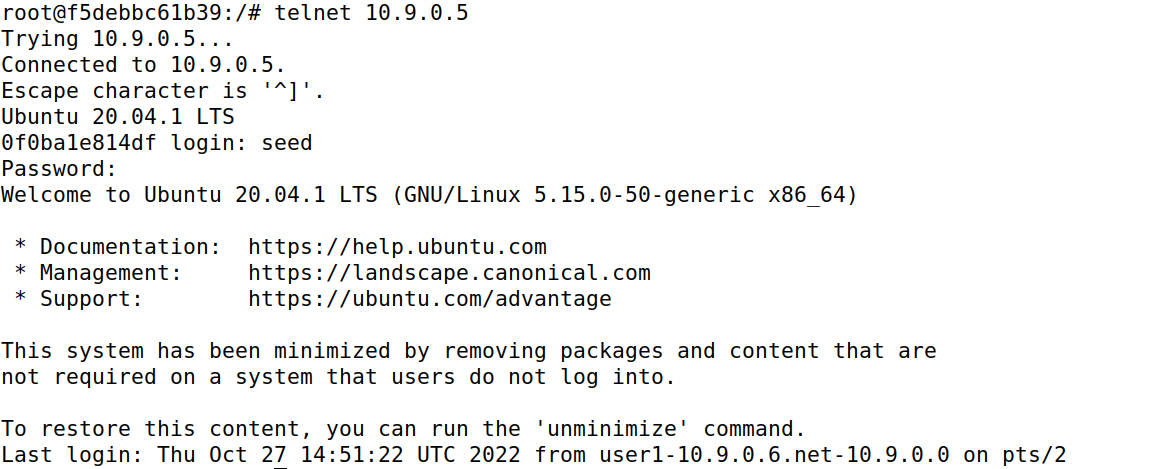


# Task 4

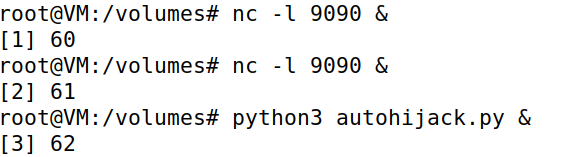
* For this task I have modified the code from Automatic Hijack Script.



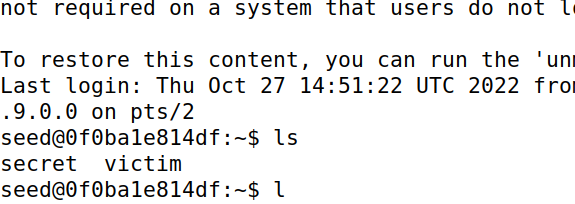
* Re-establishing link between **user1** and **victim**’s machine from **user1**’s terminal.



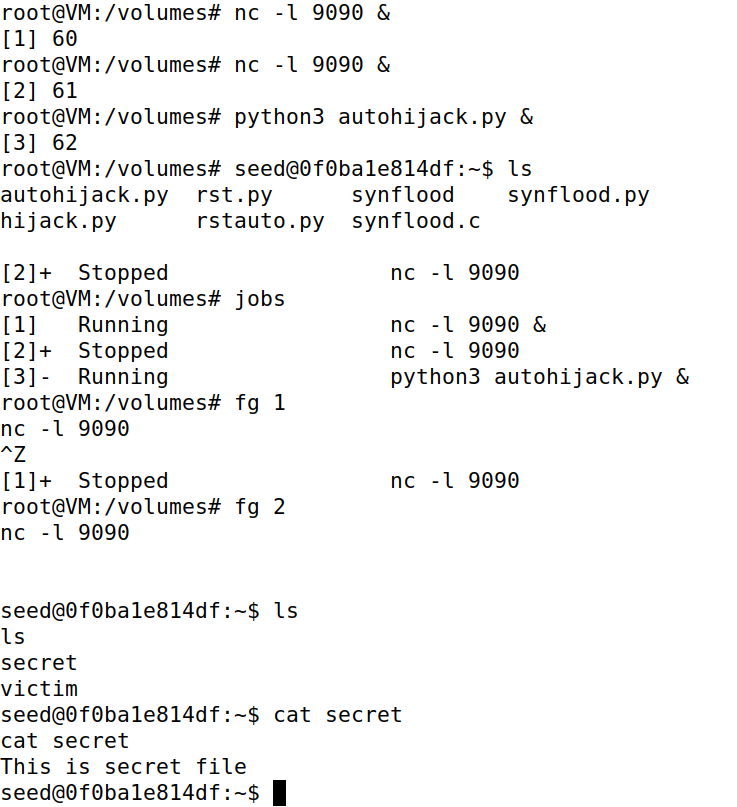
* Launching the Attack and listening on port 9090



* In **user1**’s terminal we got stuck while typing



* Now coming back to the **Attacker**’s terminal we just need to press simple enter or a command and then enter. After that we close one of the jobs which were listening on port 9090 and with the other one we will simply use forgo command and then press enter to get the reverse shell. Now we can use the commands to access the **user1**’s terminal and get the secret.



Hence, all the objectives have been successfully achieved.