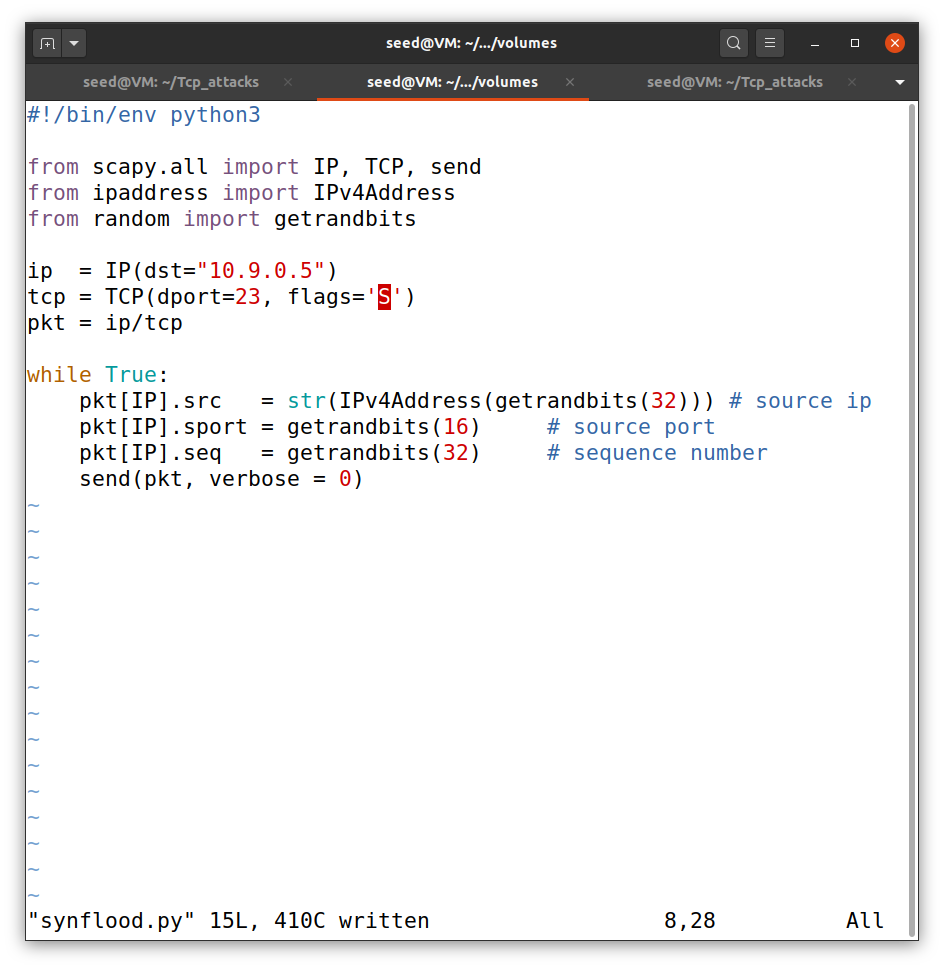
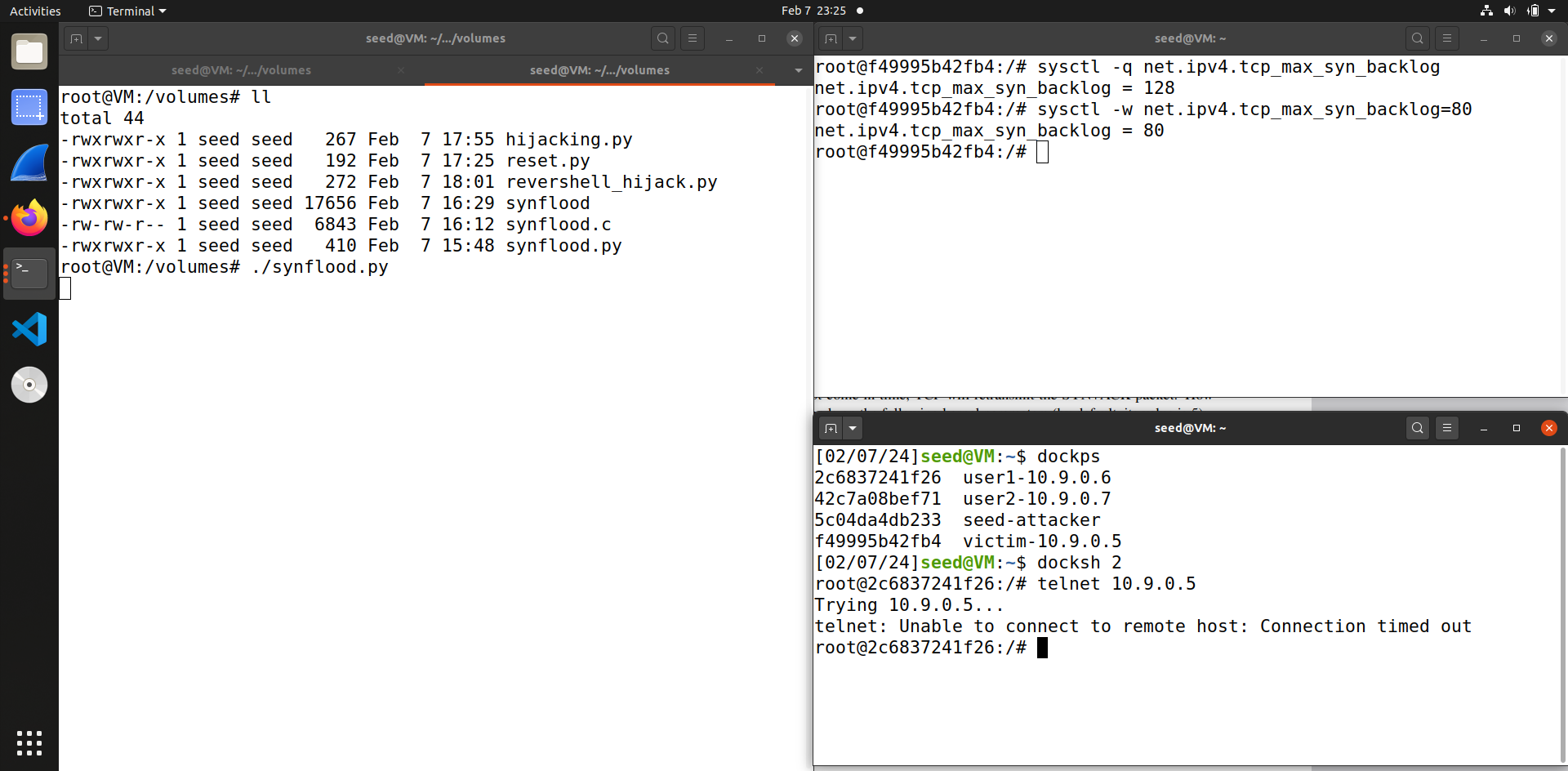
**Lab 4 TCP/IP Attacks**

**Task 1 : SYN Flooding Attack**

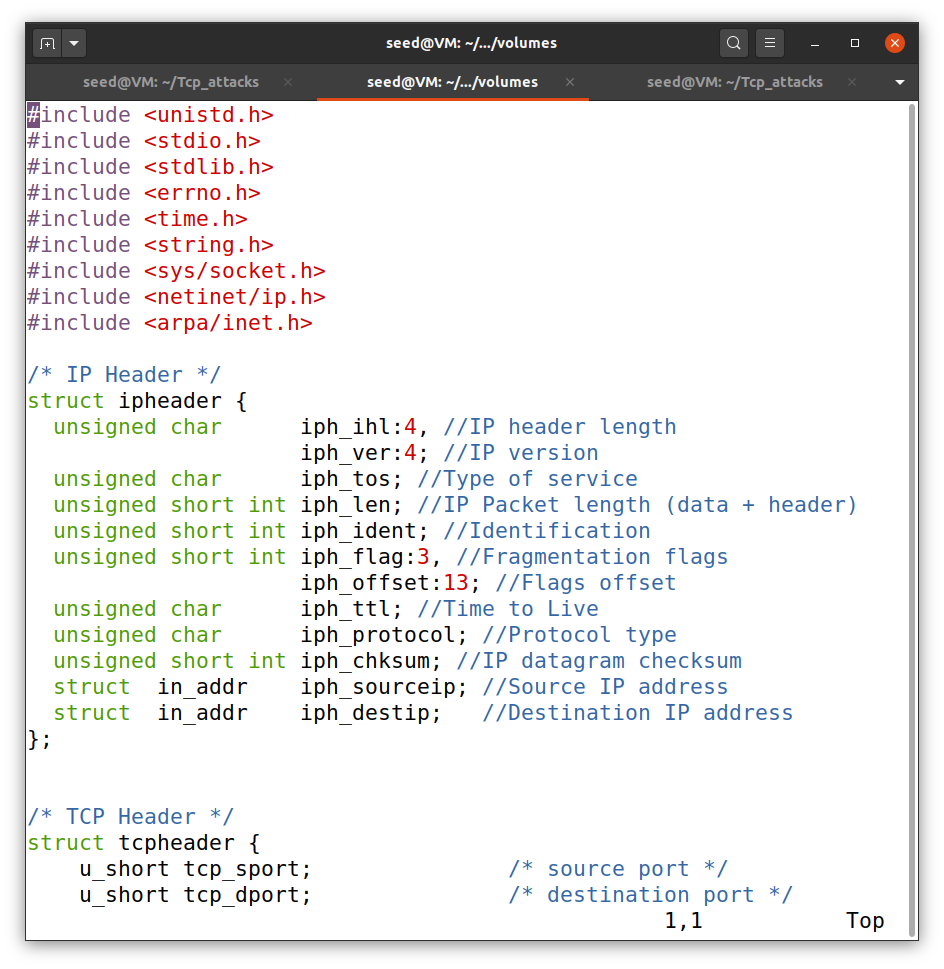
**Task 1.1 Launching the Attack using Python**

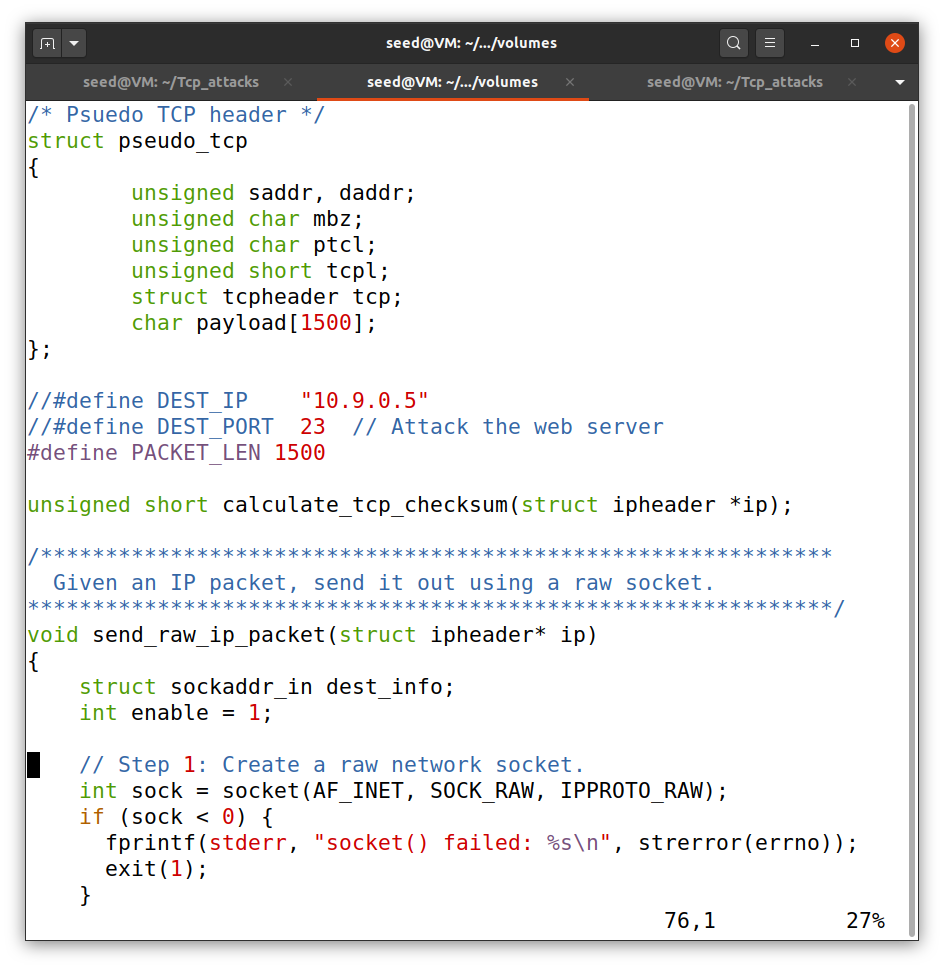
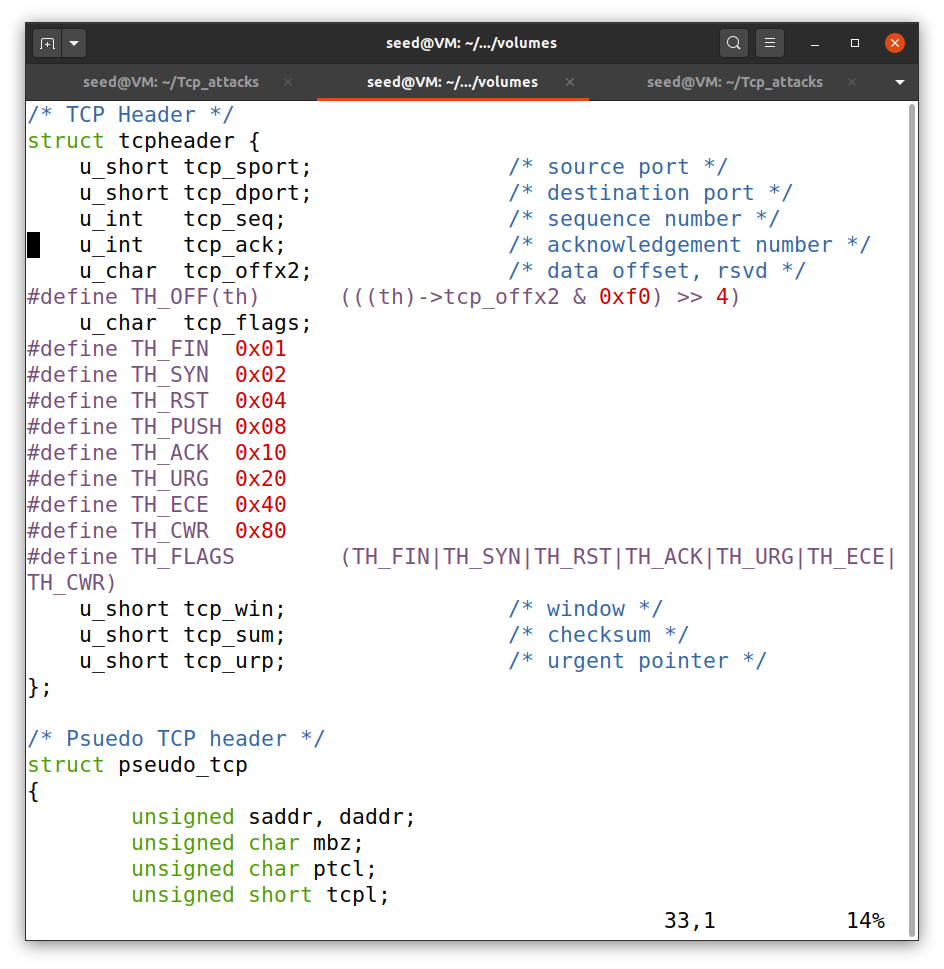


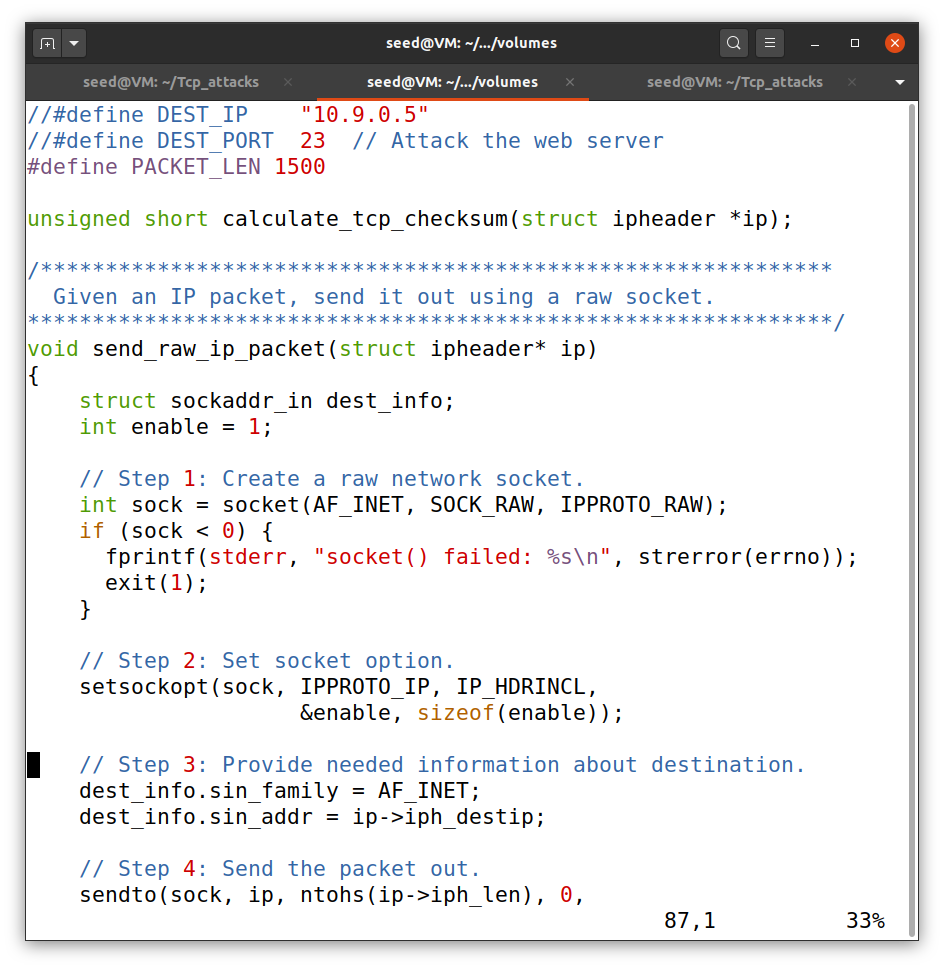
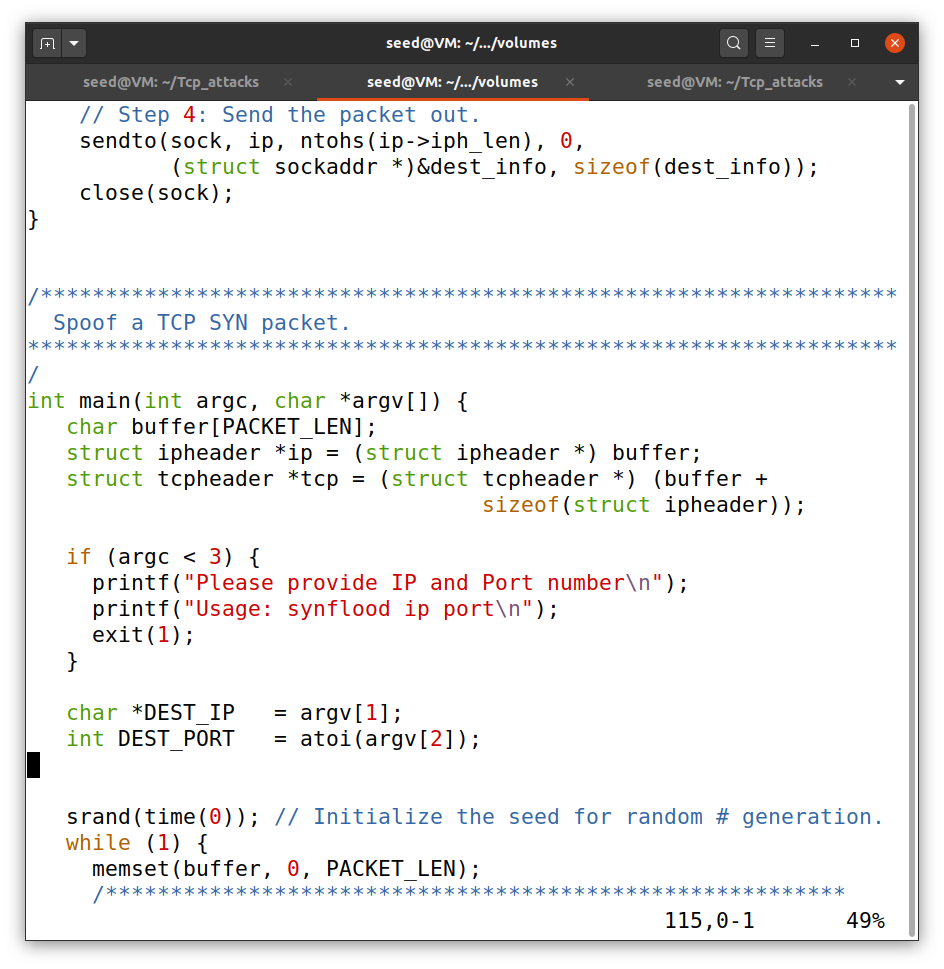
This is python code I have used for the synflood attach. I have entered destination address for attack which is 10.9.0.5 and then set the flag to S and 23 for telnet.

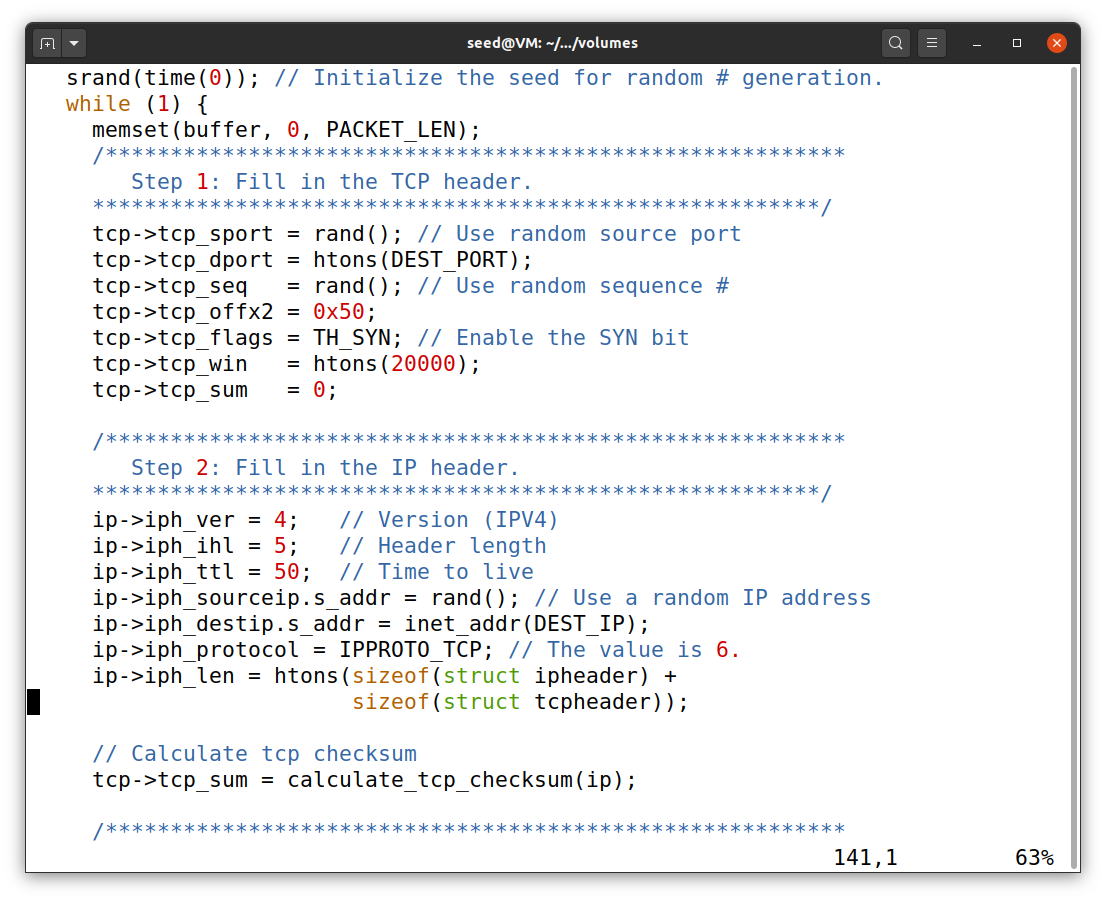


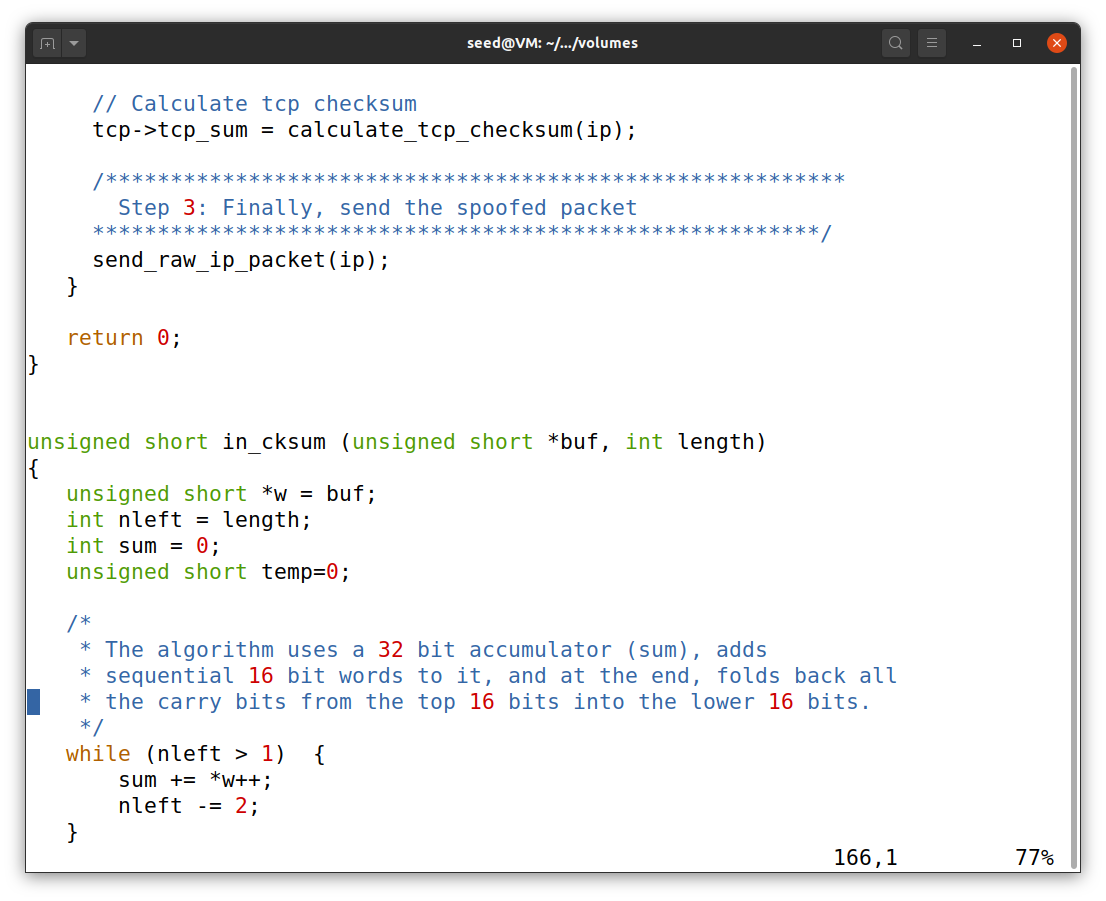
So in this code first I checked the queue size is 128. But as we are using python language for attack it is very much slower, so for the attack to be successful I have decreased the queue size to 80. So as you can see through the screenshot after we decrease the size and then launch attack, user 1 is not able to connect with the victim attack is successful.

**Task 1.2 Launch the Attack Using C**

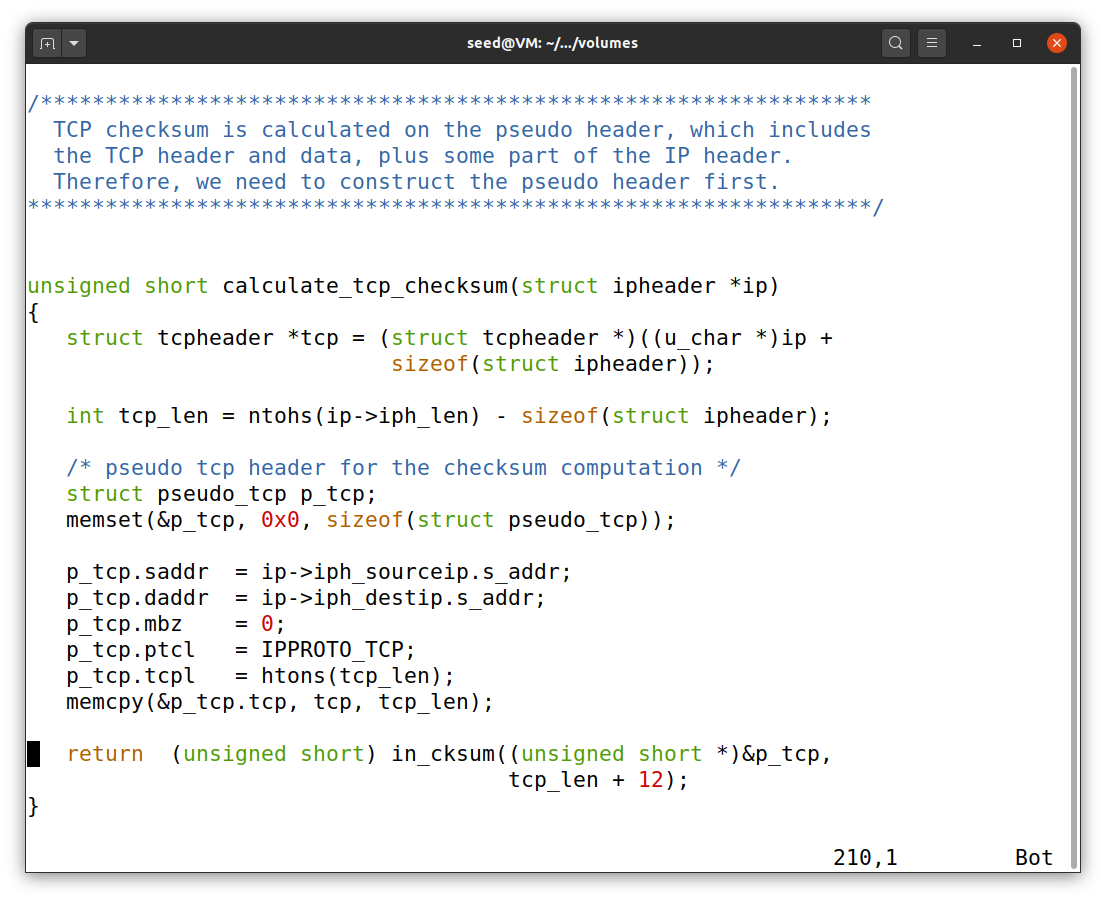
****

**** 

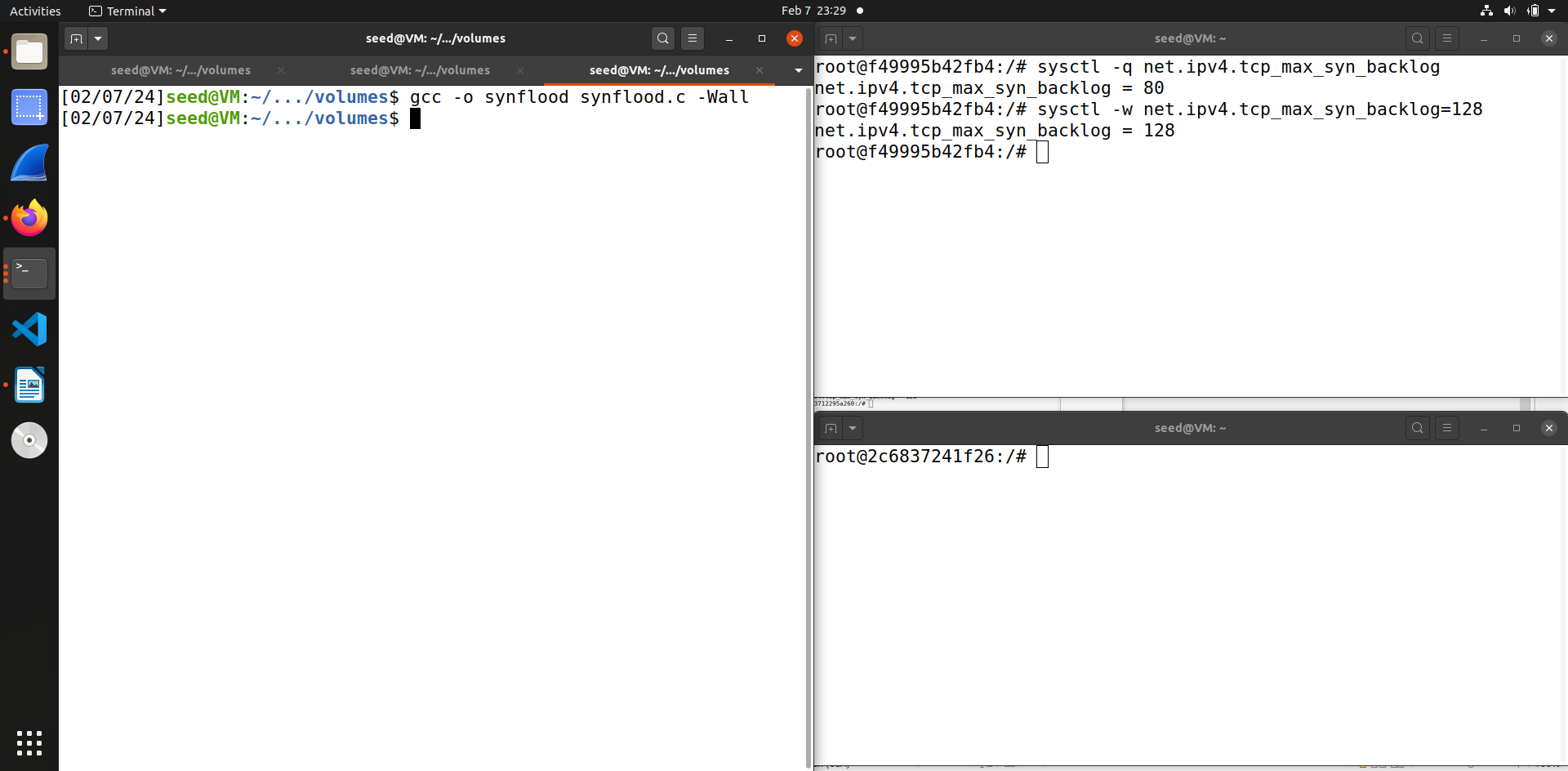
****

****

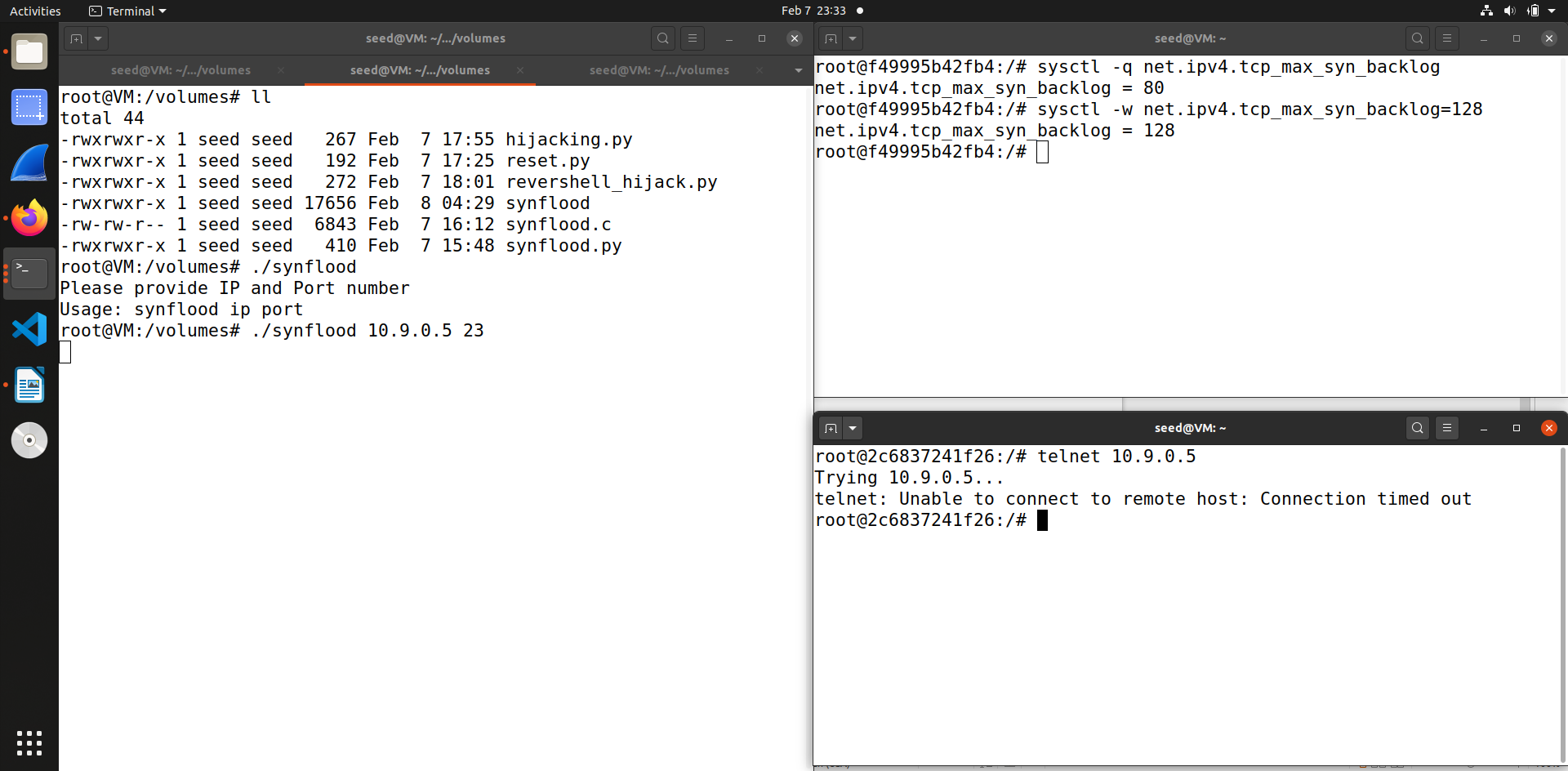




This is the C code I have used for synflood attack.



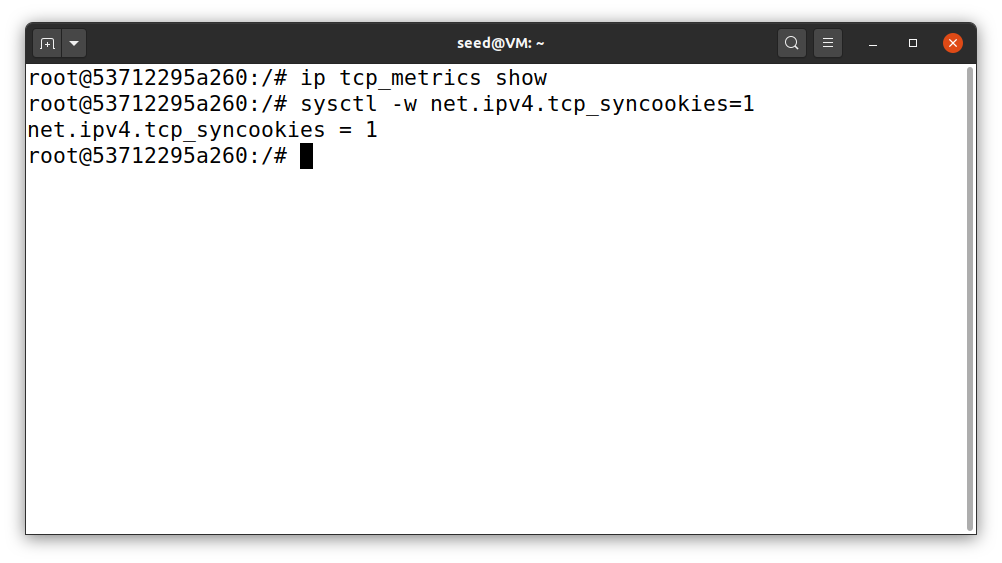
So in this step we first change the queue size to 128 once again. Then compile the code using gcc and I have also added -Wall to check if there are any warnings.



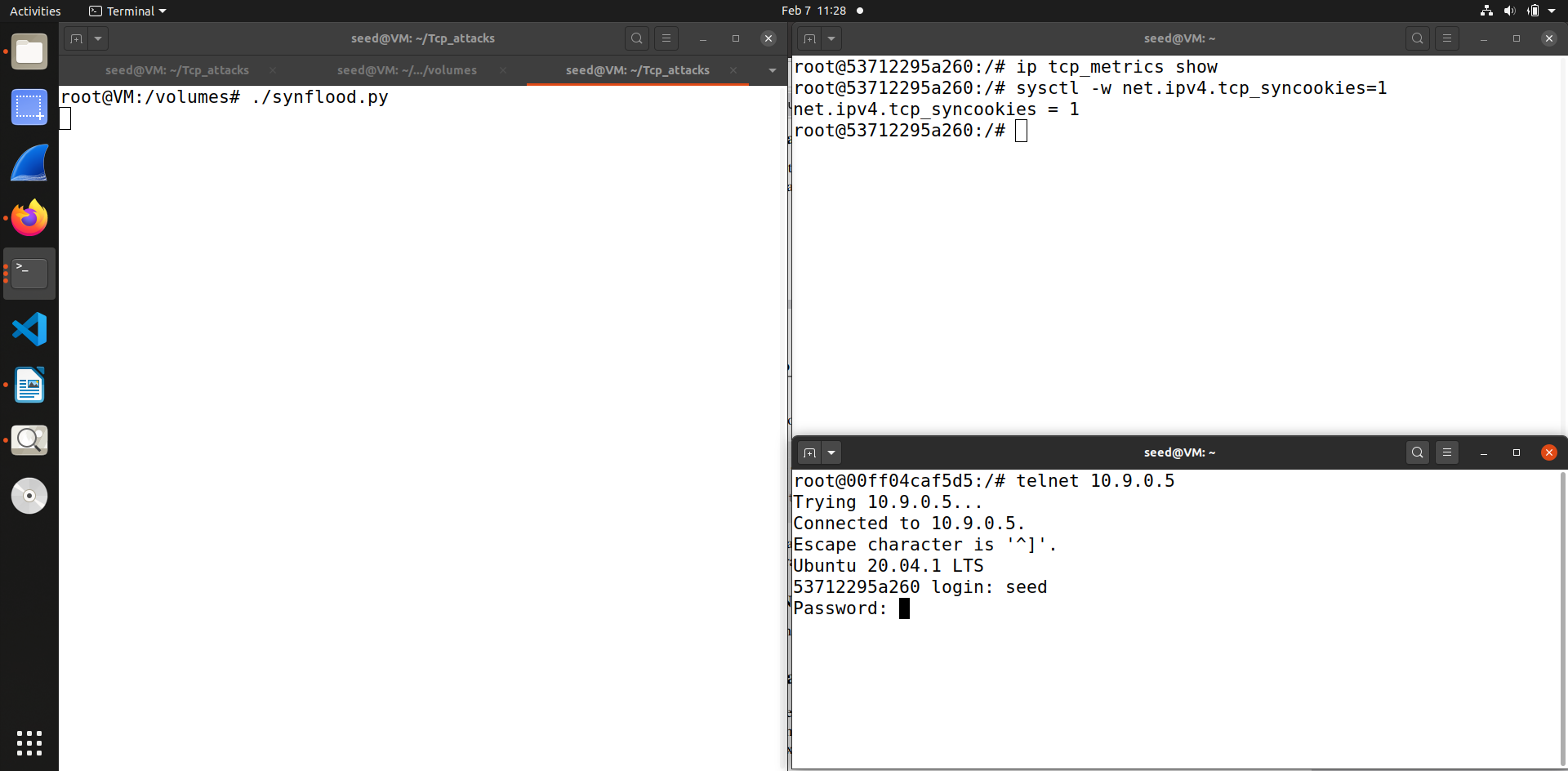
This is the screenshot that contains the output when when compiled and run the C code. For running the code we also have to add the command line arguments that is the victim IP address and telnet port number. In the other terminal you can see when I try to telnet the victim 10.9.0.5 the other user 10.9.0.6 is not able to connect and once again attack is successful.

**Task 1.3 Enable the SYN Cookie Countermeasure**

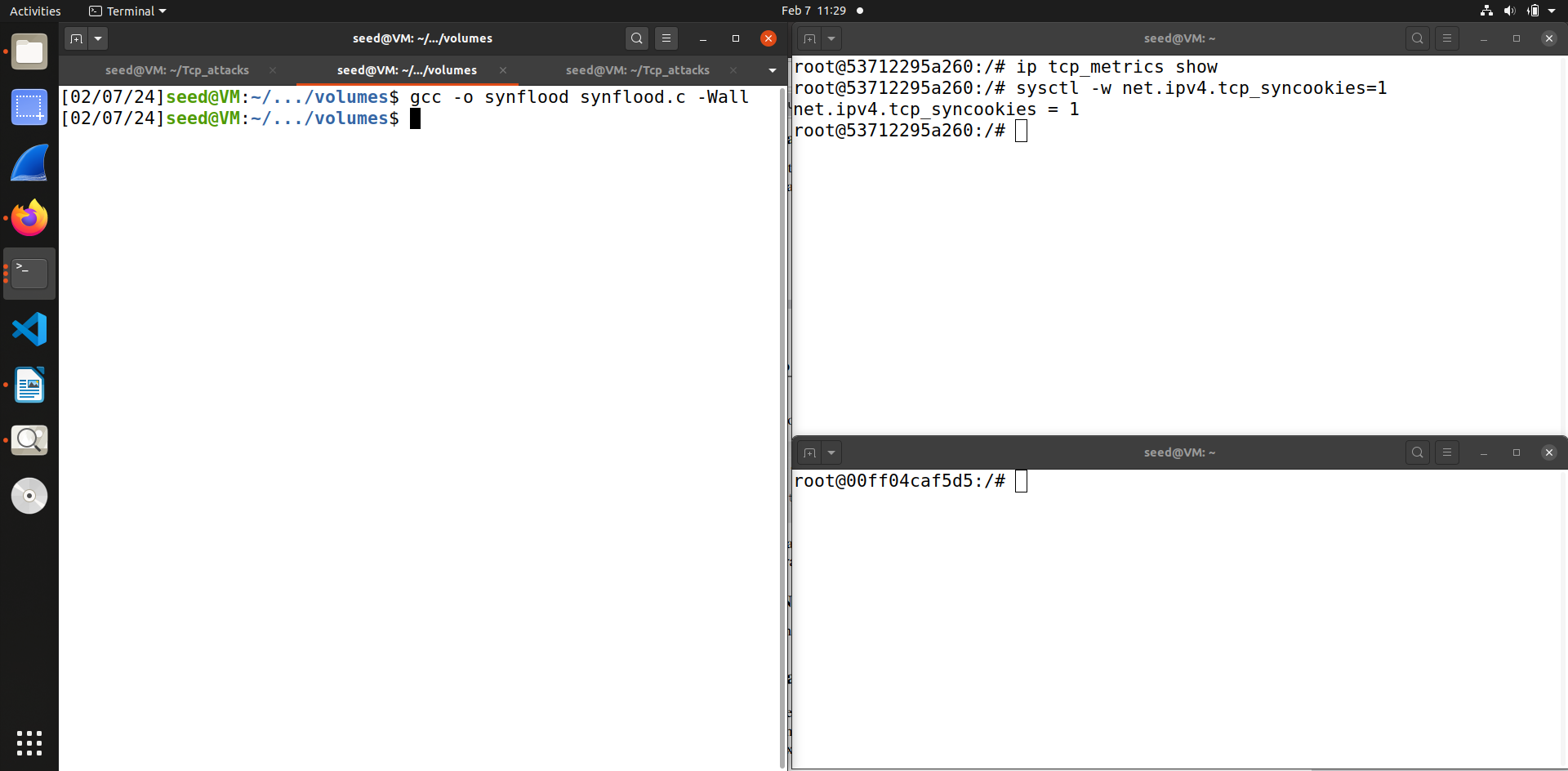
In this task we enable the syn\_cookie mechanism to defend through the synflood attack for both python and C.

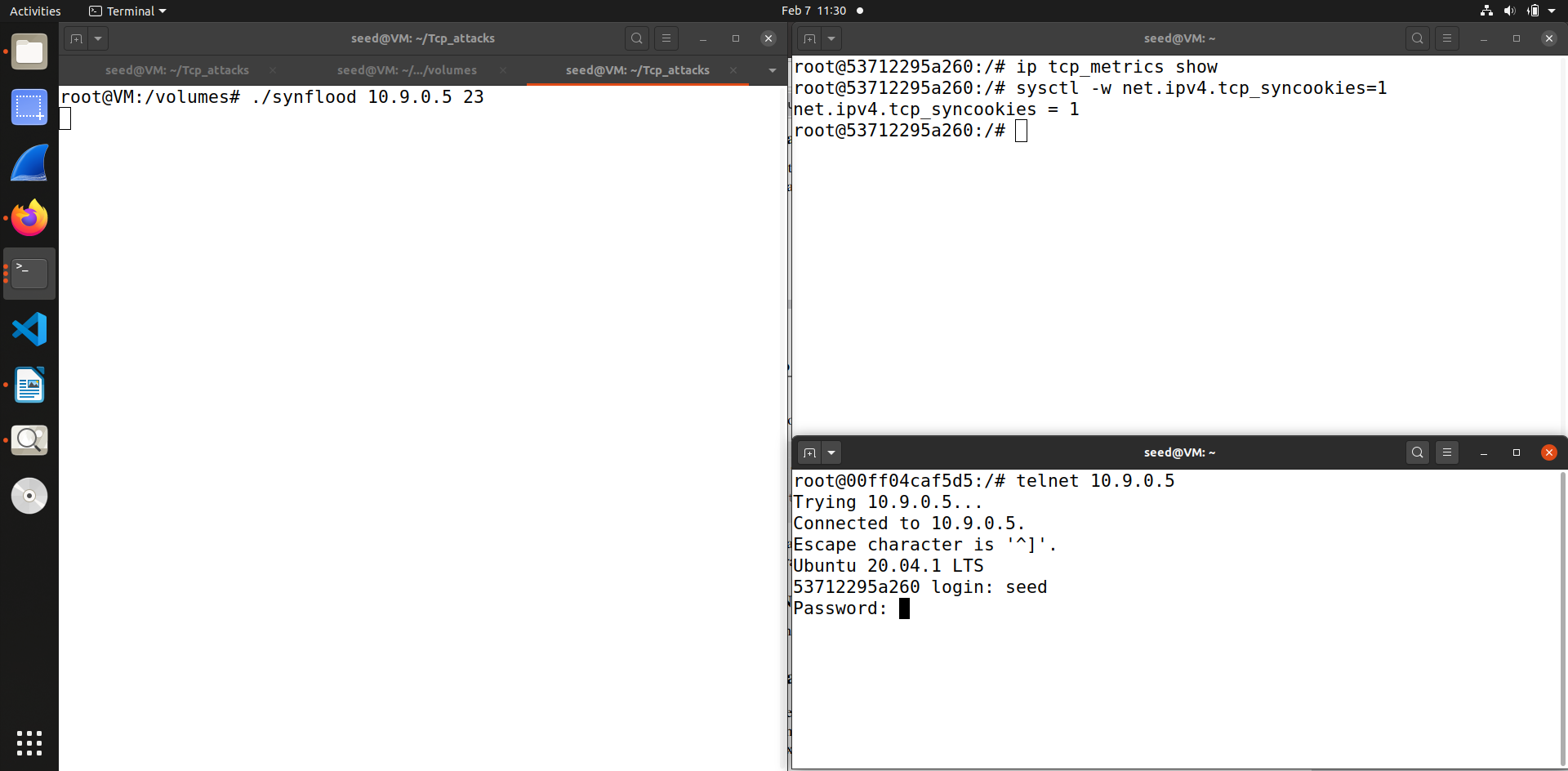


So first I checked if the user1 Ip address cache is been added or not using tcp\_metrics show and then I enabled the syn cookies to enable the defense.



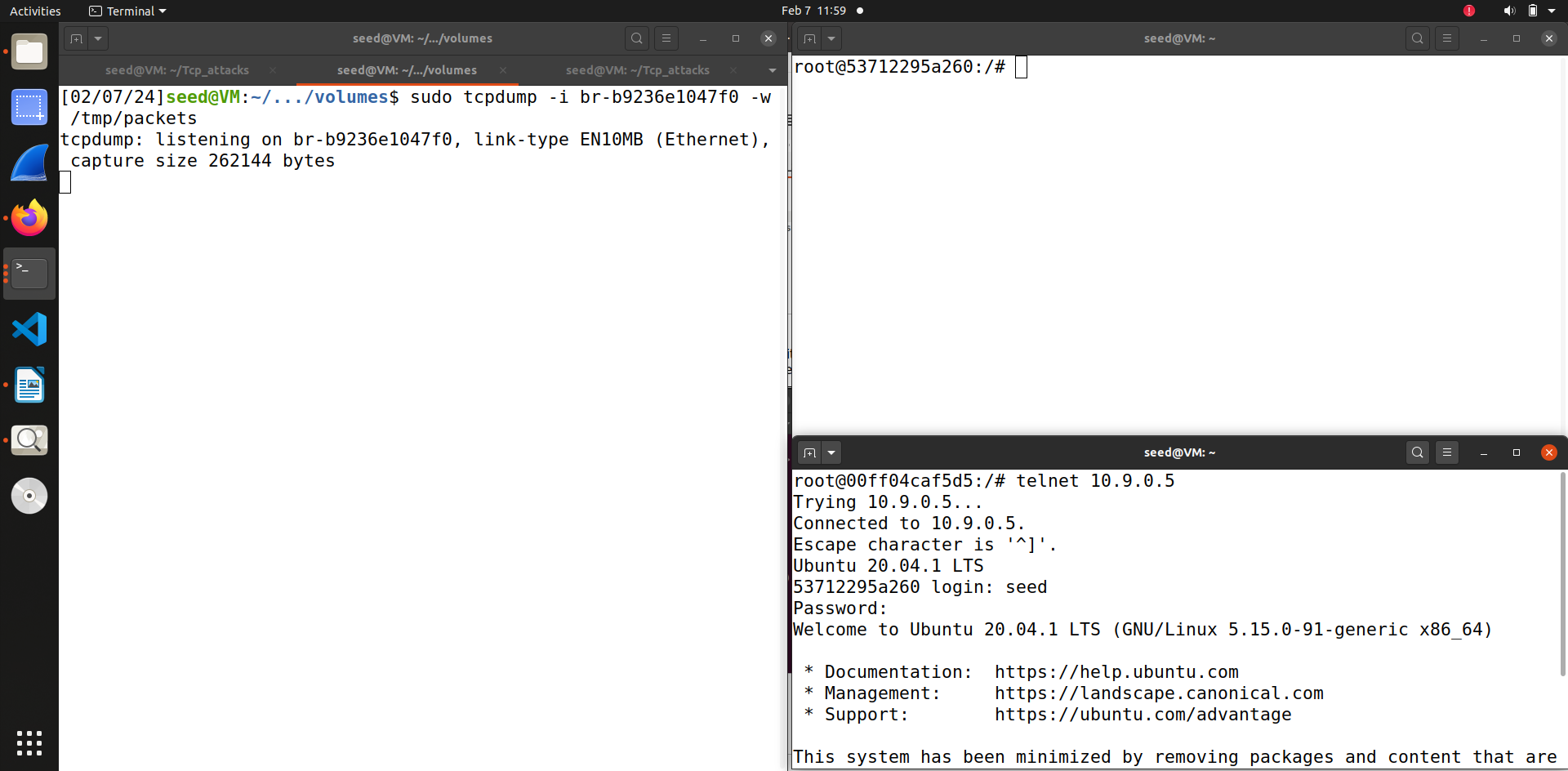
This screenshot shows that I have enabled the syn cookies and then try to run the synflood python code we see that the attack fails and the user1 is able to connect through telnet connection.

In this screenshot I have compiled the C code for synflood attack when we enable the syncookies.

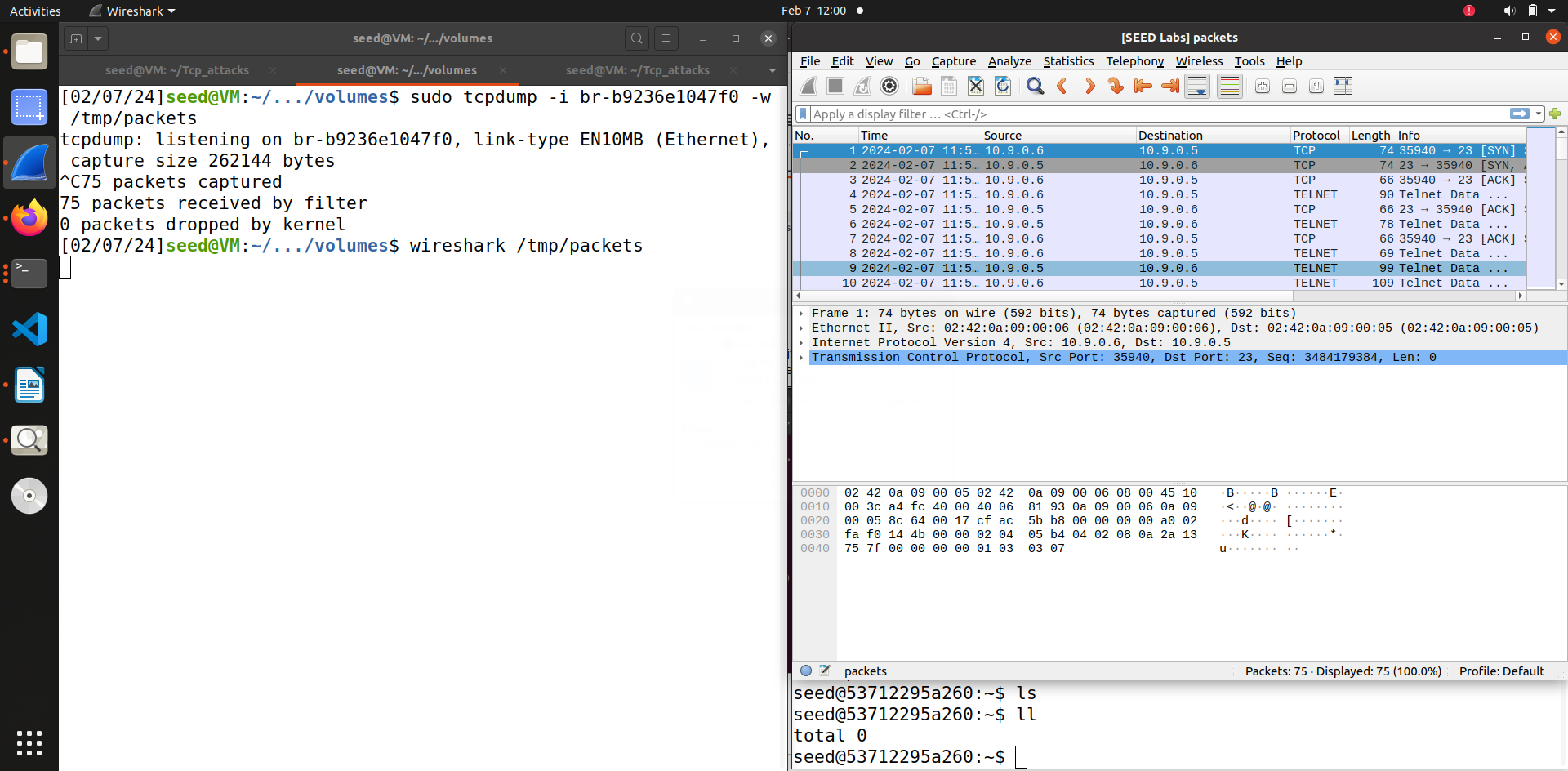


This is the screenshot contains output after compiling the C code for synflood we can see that the attack still fails the user is able to telnet to the victim machine.

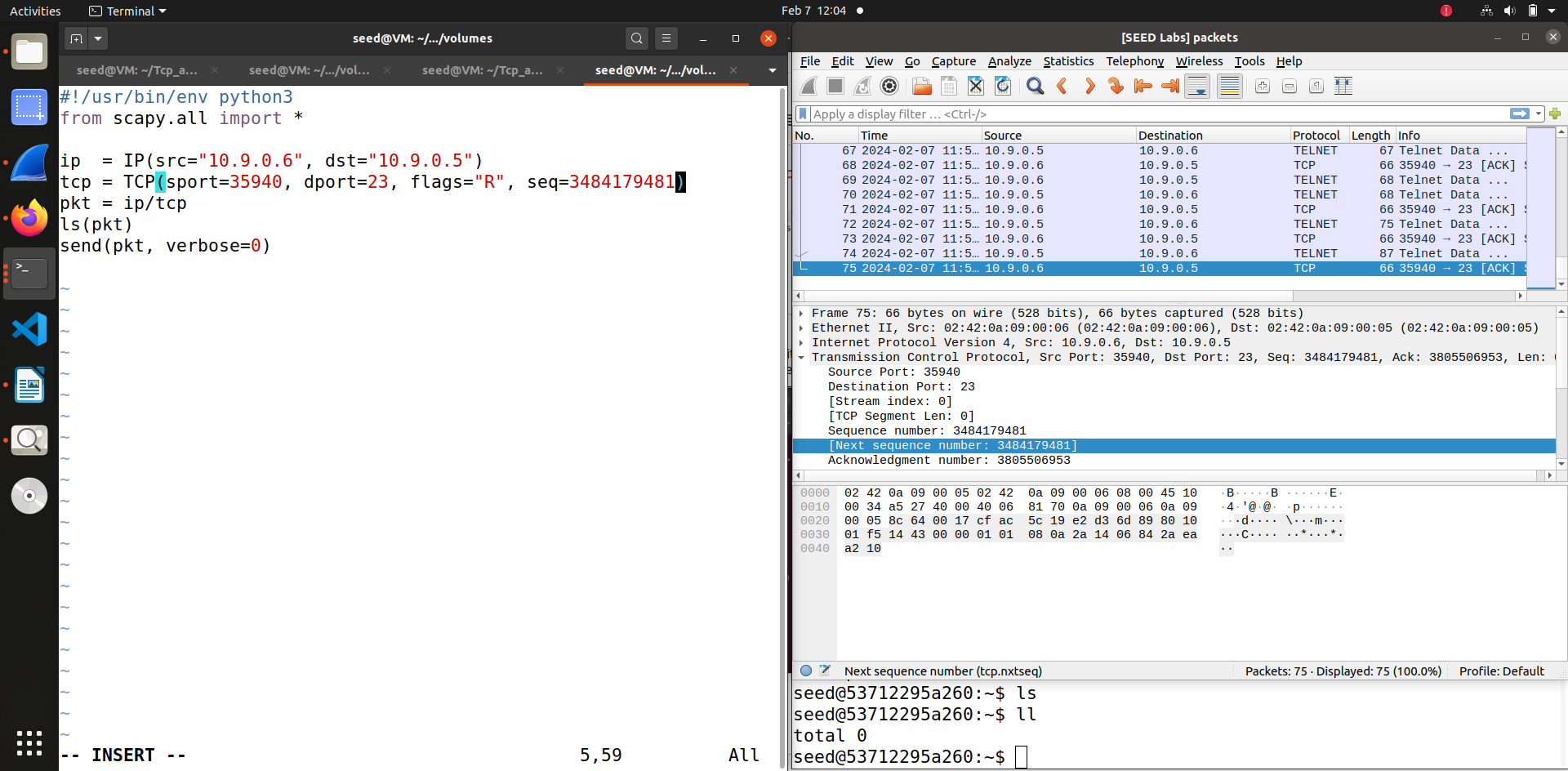
**Task 2 TCP RST Attacks on Telnet Connections**

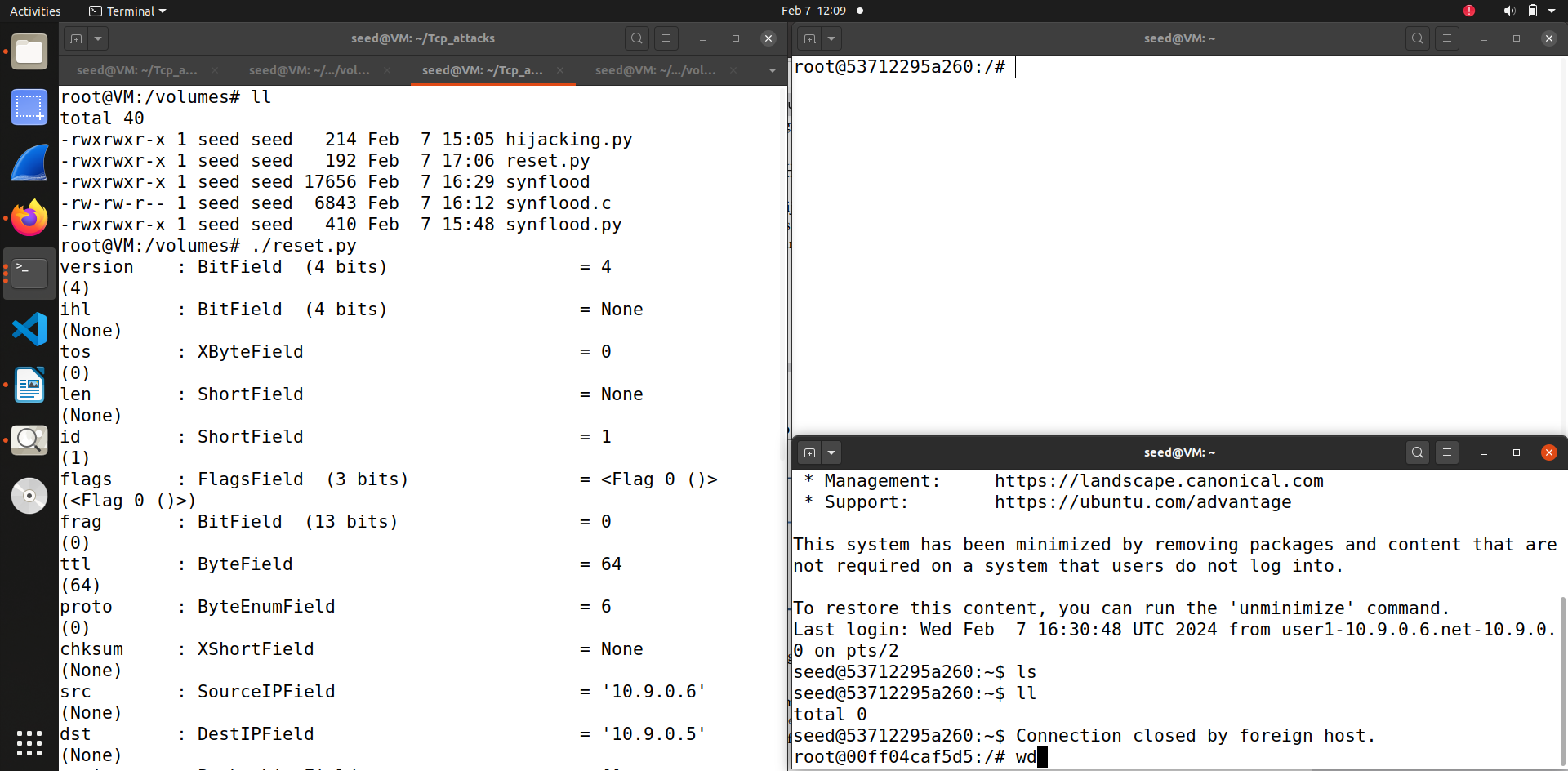


In this code first we telnet the victim user, and record the traffic using wireshark, as we need different parameters for the Reset attack such as the source port and the sequence number.



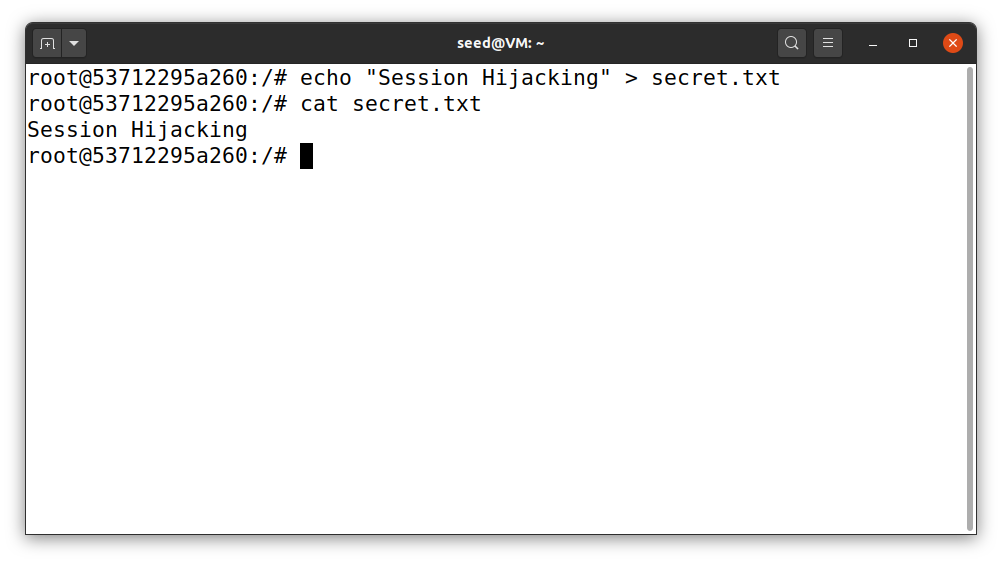
So this is the screenshot that contains that we are recorded the traffic between victim 10.9.0.5 and other user 10.9.0.6

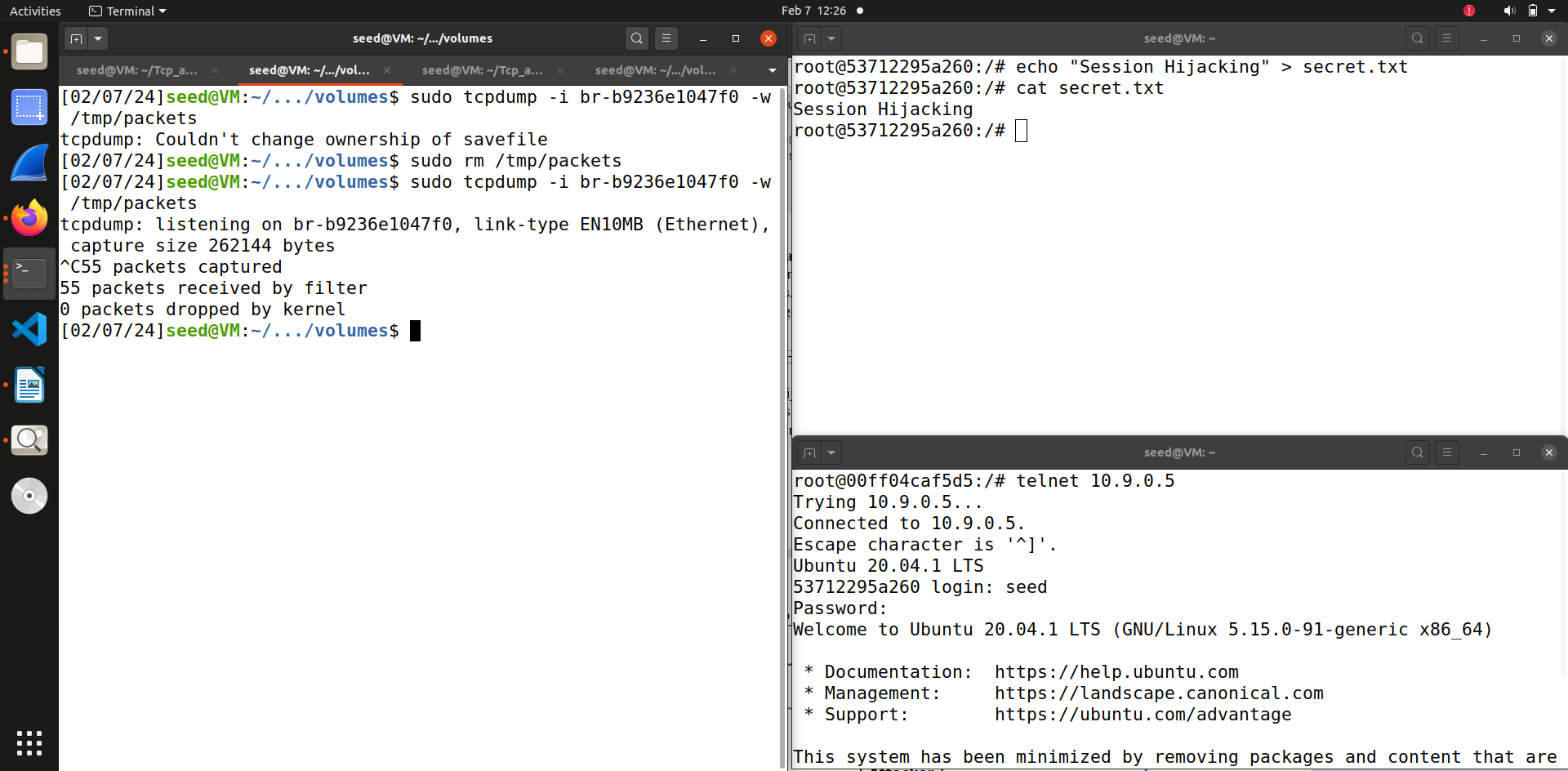
So after recording the traffic we will put the source Ip address, destination Ip address, source port, destination port, flag and sequence number. In this screenshot we can see the source Ip is 10.9.0.6 that the attacker will act like and destination IP is 10.9.0.5, the flag is R for reset and destination port is 23 for telnet connection, other parameters like source port and sequence number I added using wireshark .



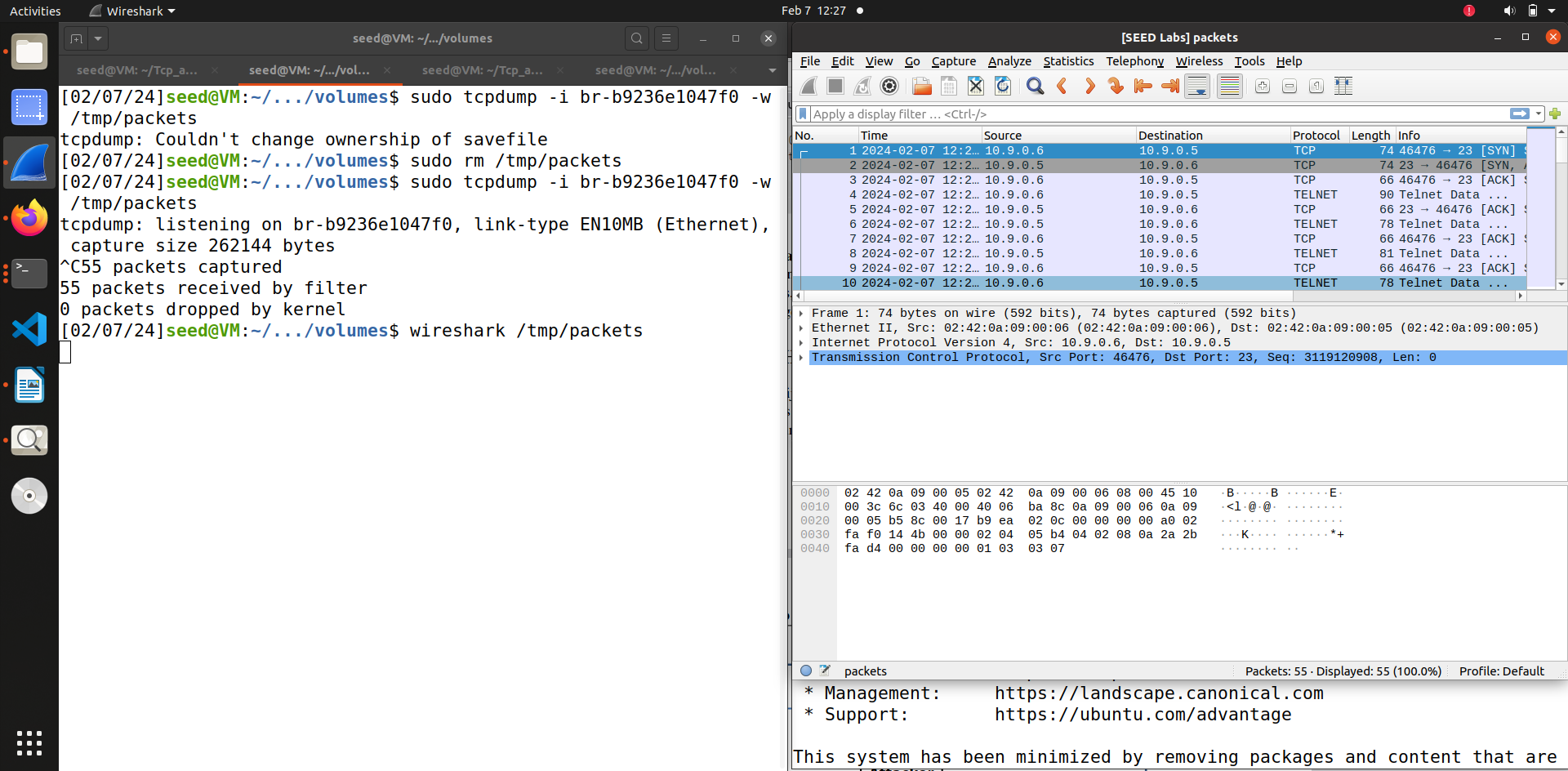
So when I run the code from attacker machine you can see that the telnet connection that the other user 10.9.0.6 had with the victim is broken. Thus we can see that Reset attack was successfully.

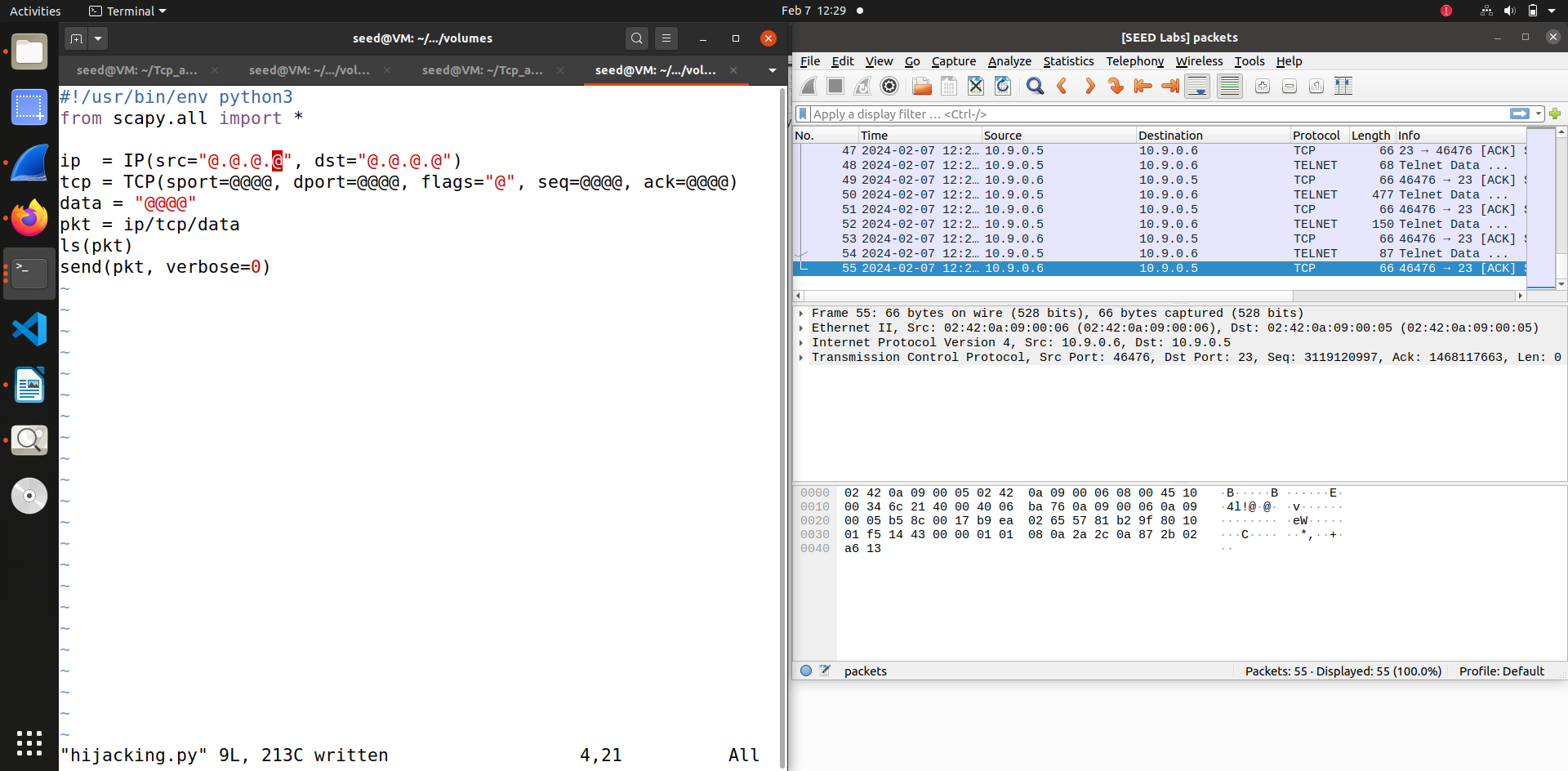
**Task 3 TCP Session Hijacking**

  
So in this attack first I created a text file in the victim’s machine.

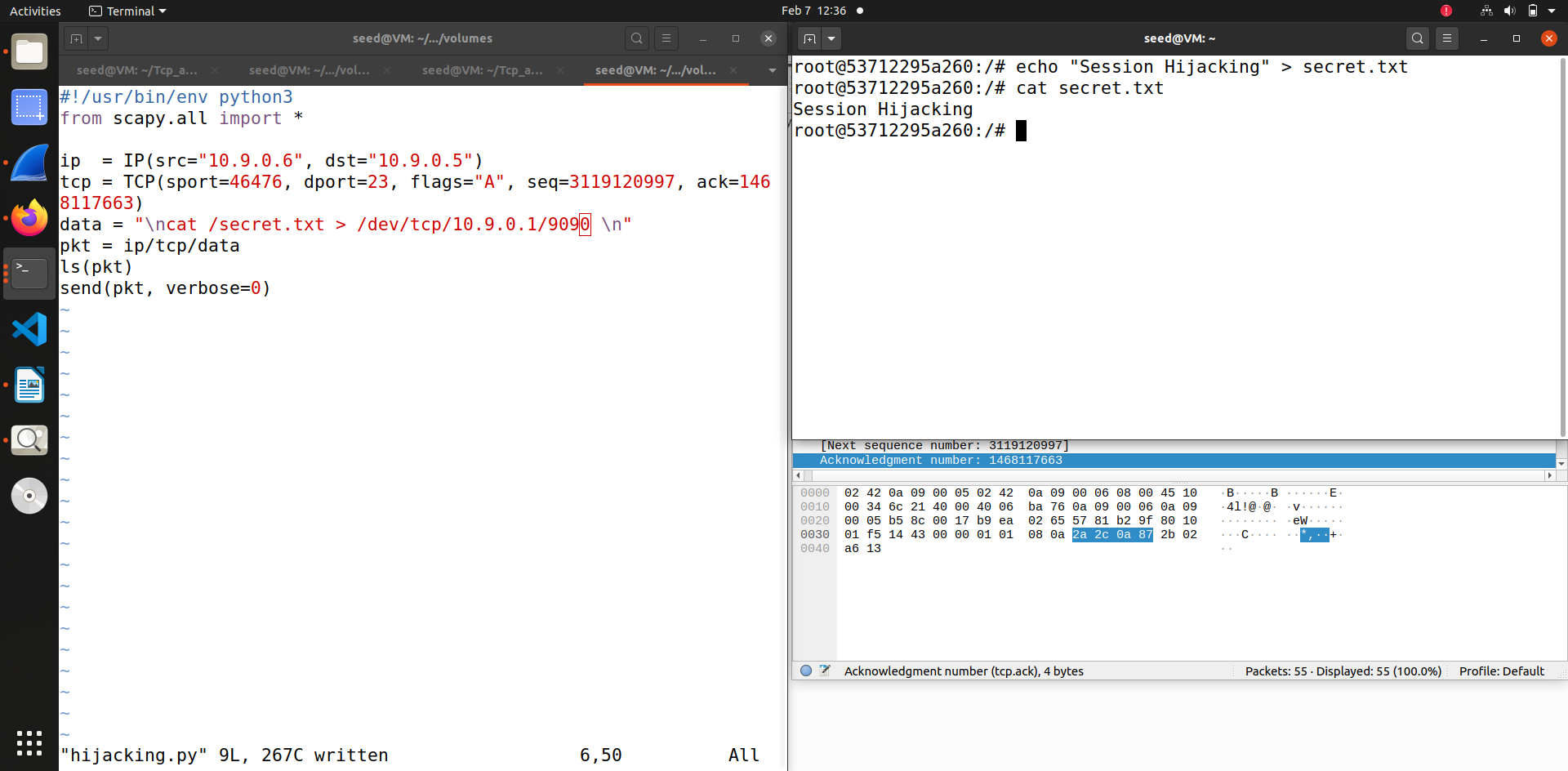


After that as I had done in the previous attack I recorded the traffic between the victim 10.9.0.5 and the other user 10.9.0.6 using wireshark.

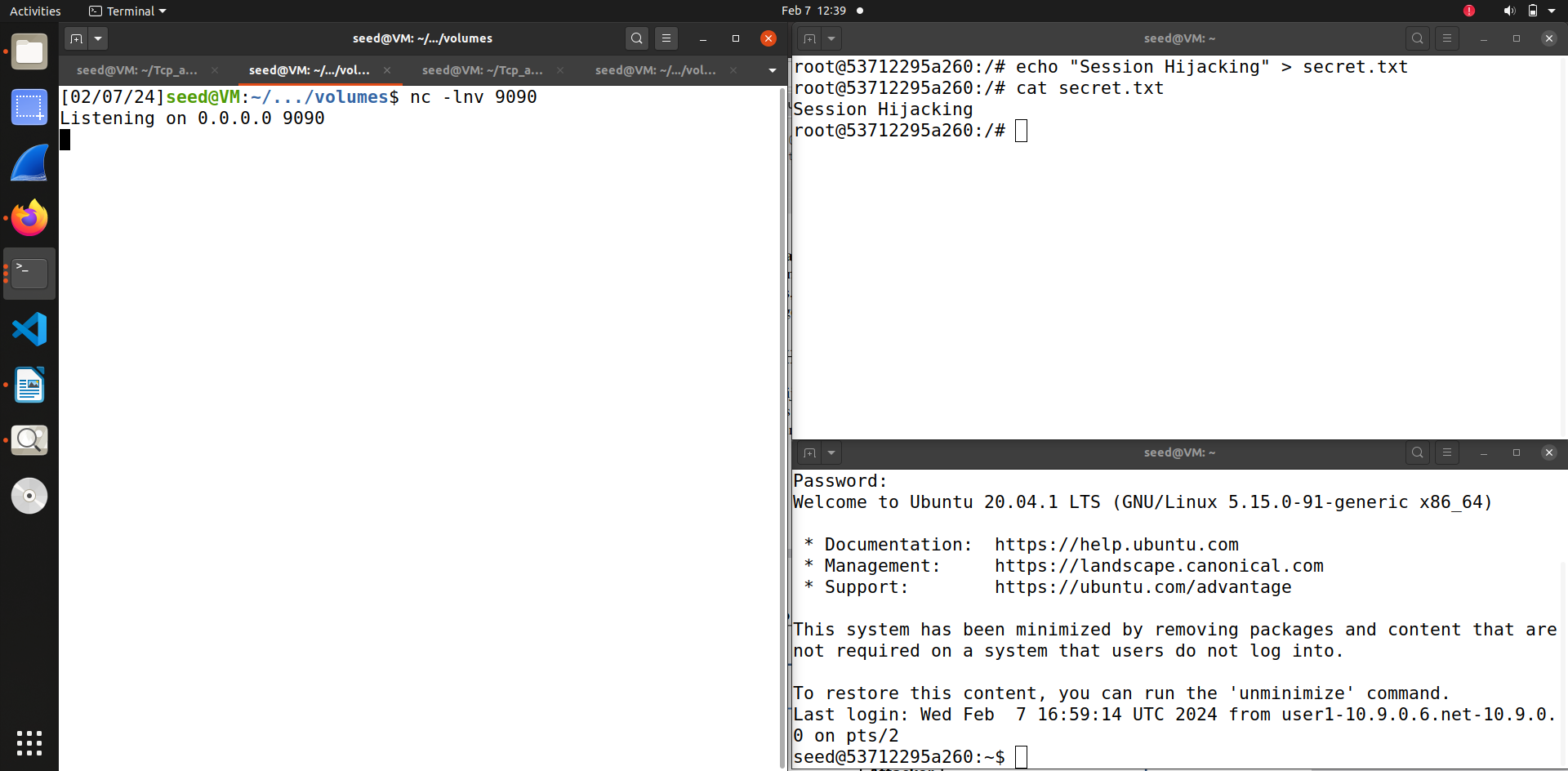




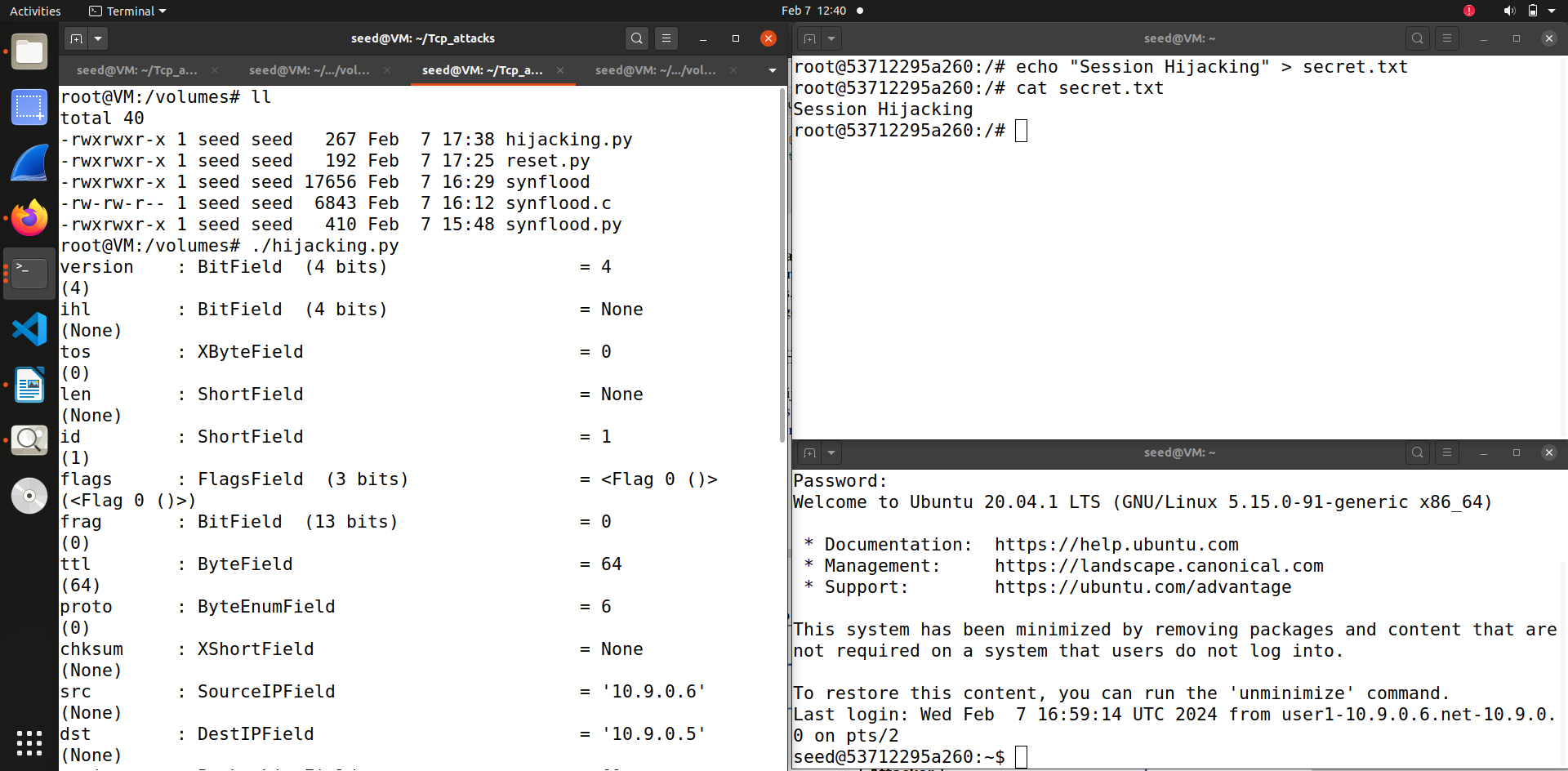
Then as you can through the both screenshot we first open wireshark to get the recorded data between the user and the victim and then fill out the parameters such as sequence port, sequence number and then acknowledge number using the data we recorded in wireshark and other parameters such as source IP and destination Ip and destination port and flag.



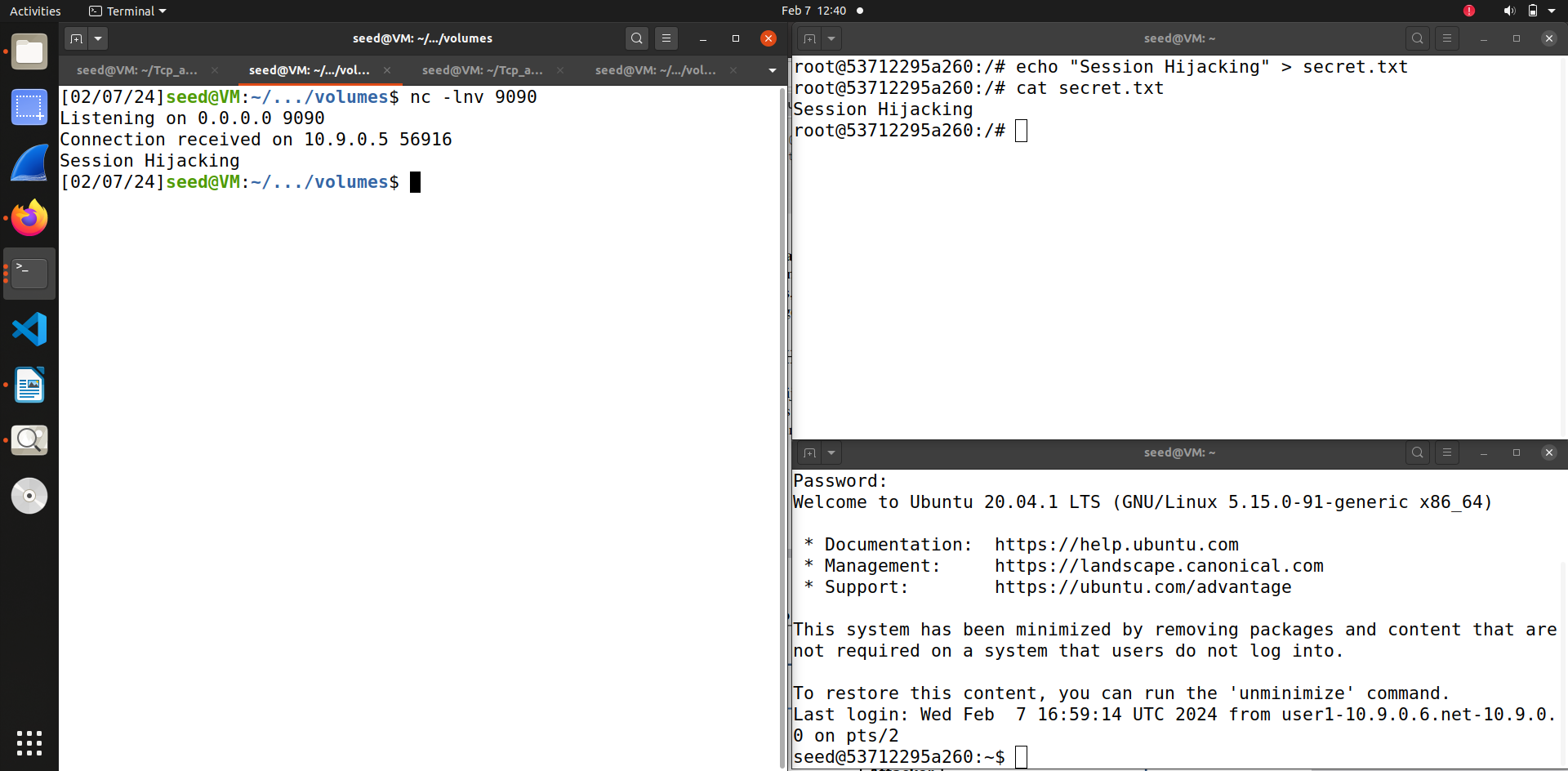
So this is the screenshot of code after filling the parameters and which I used for session hijacking. So first I have filled out the details using wireshark and then set flag to A for acknowledge and destination port is 23 for telnet. Then in data I have added the cat command to read the secret file that I created in the victim’s machine.



So for attack first we use the netcat command to listen on attacker machine.

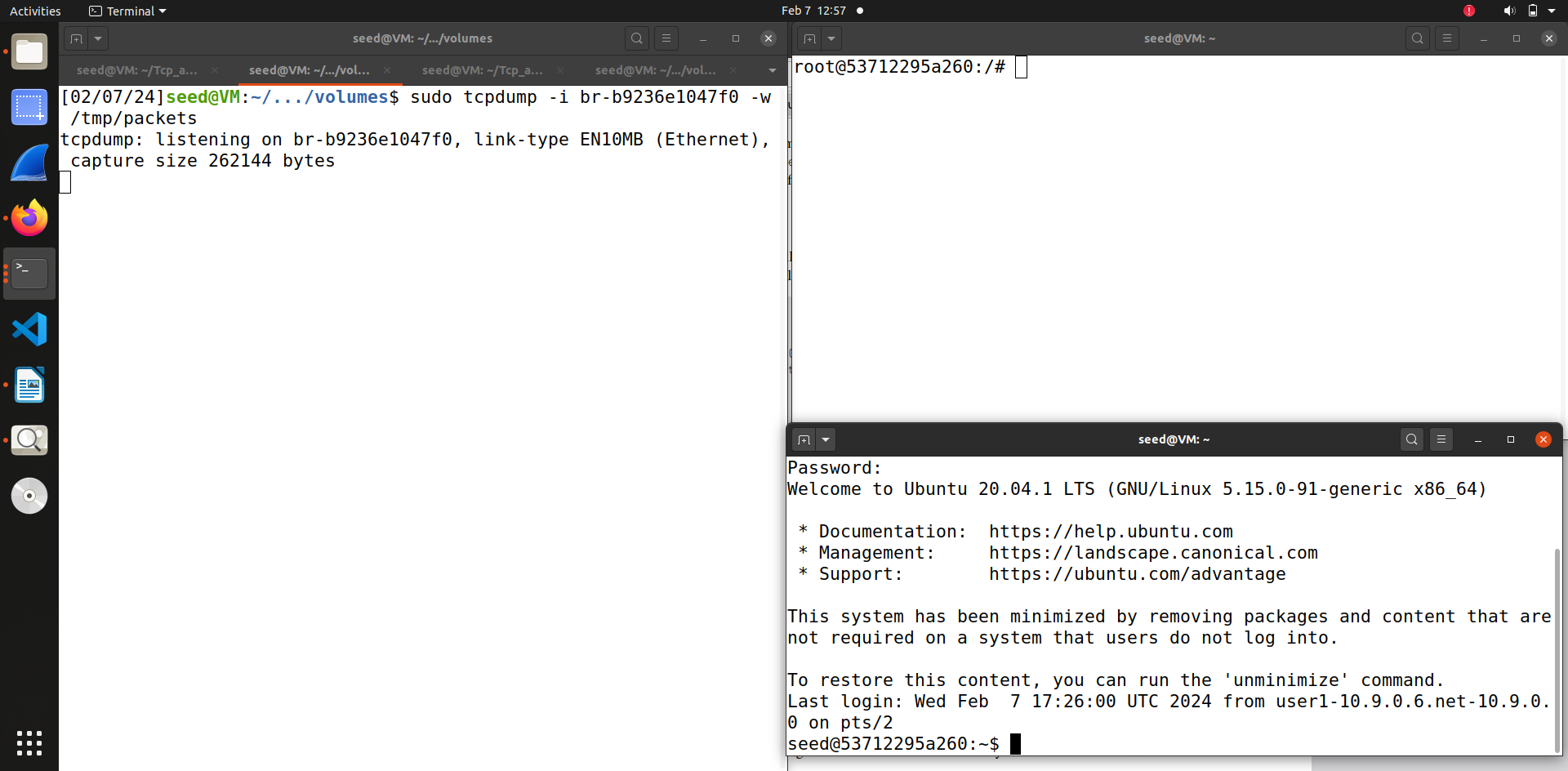


After running the netcat command and start listening we run the attack code.

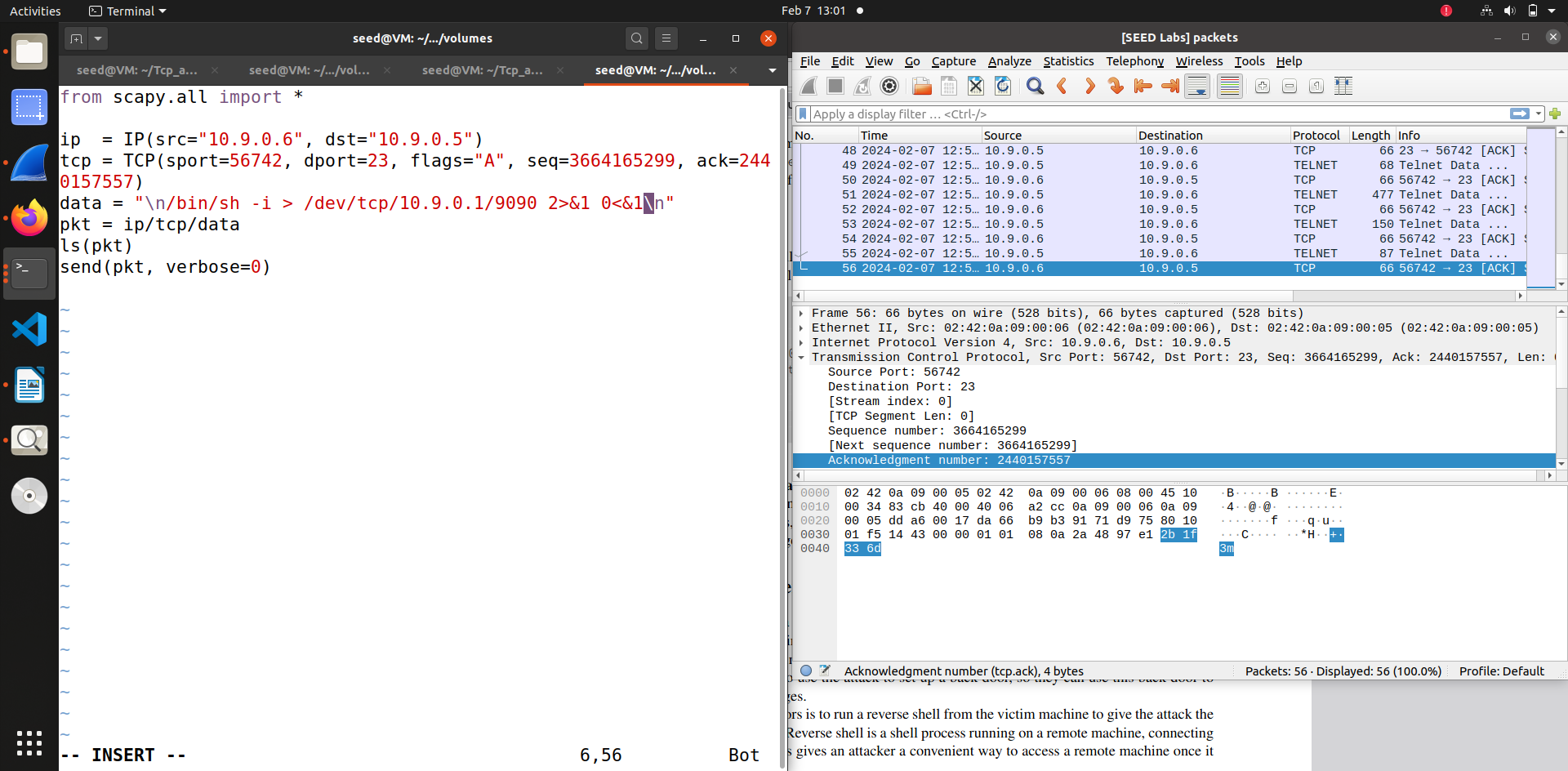


So as you can see the attack is successful the attacker is able to read the file contents that I had created in victim’s machine.

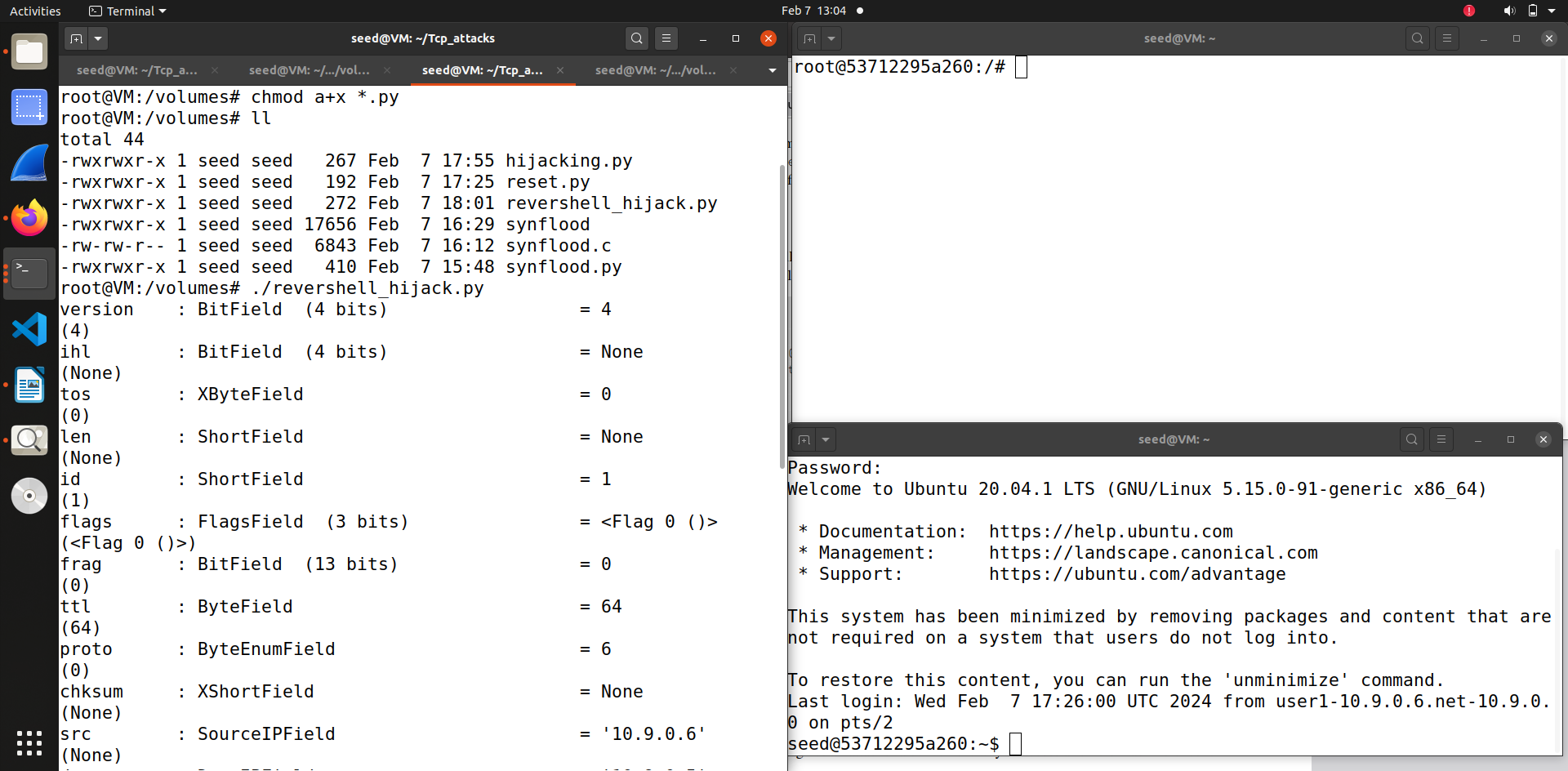
**Task 4 Creating Reverse Shell using TCP Session Hijacking**

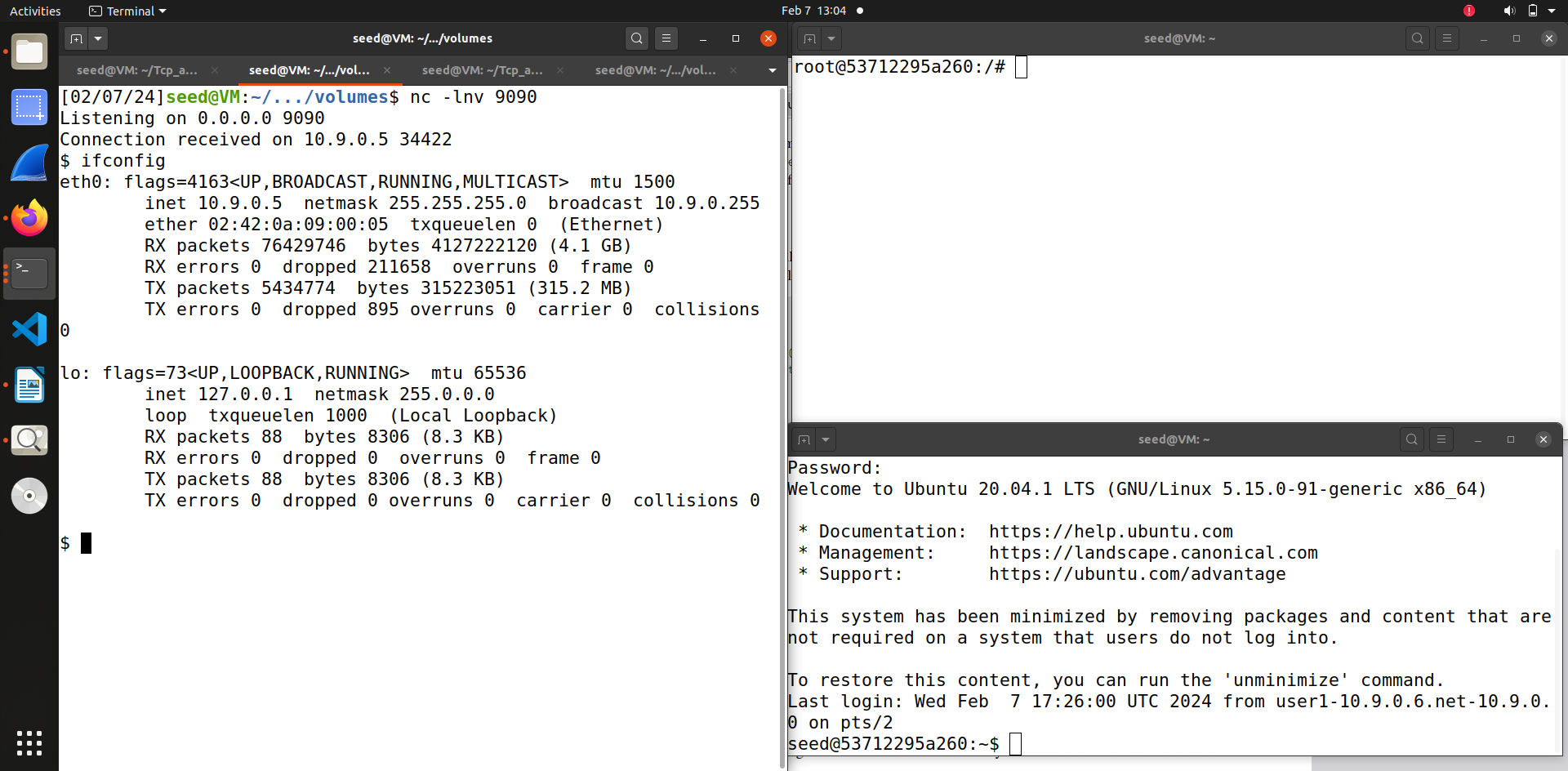


So for this task as I had done in previous task first we first establish a telnet connection between victim and user and record the traffic using wireshark.



So in this screenshot we first add the data that we had recorded through wireshark and then change the data parameter add the bash shell and the remaining reverse shell command given in SEED labs manual and execute the attack.





As you can see through both screenshot first we use the netcat command on attacker machine and start listening. Then after running the code for attack you can see that the attacker gets access to victim’s machine, I also did ifconfig and you can see that attacker has got access to victim’s machine and the attack is successful.