**CS4238 Lab: Network Attacks   
and Detection**

The **goal** of this lab is to get familiar with **network attacks** (including sniffing, spoofing, session resetting & hijacking), as well as **their detection** using network packet analysis.

**Lab Set-Up**

You will work at your own Kali and Ubuntu VMs in your local system.

**A. Packet Sniffing**

The goal of this task is to **sniff packets** sent in clear by a telnet session.

1. Connect from your Kali machine to your Ubuntu machine.

(If you can’t connect to your Ubuntu machine, do double-check the firewall/iptables rules at both machines.)

1. Install telnet server at Ubuntu:

sudo apt-get update

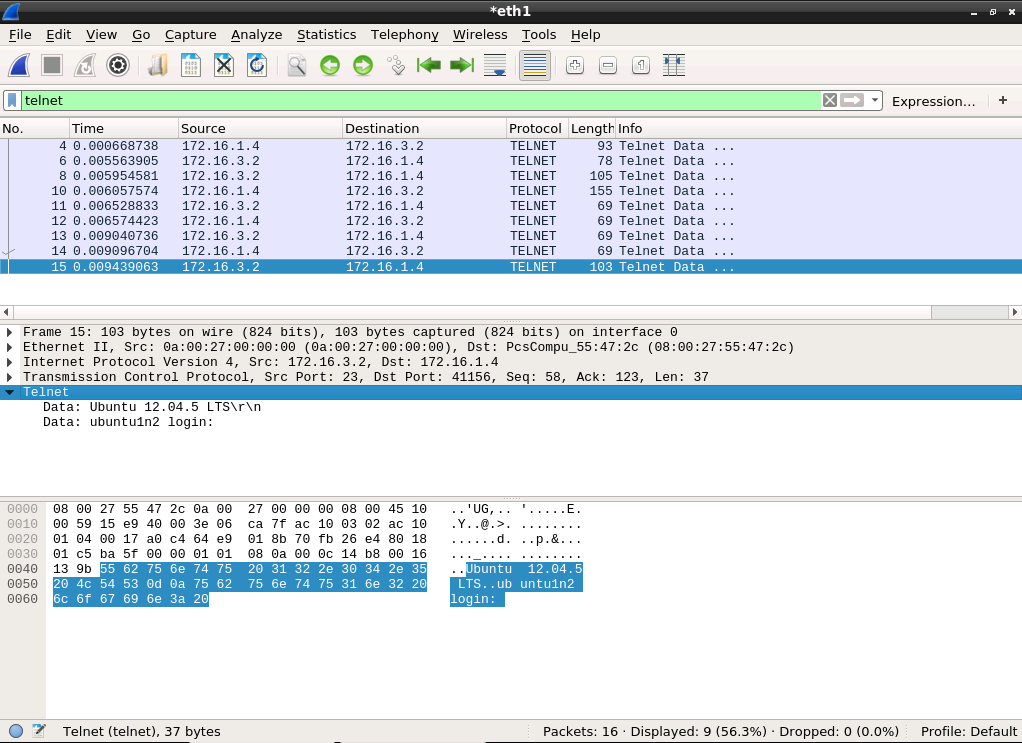
sudo apt-get install telnetd

1. Check whether you can connect to the telnet server from your Kali:

telnet <*Ubuntu-IP*>

If you can’t connect, do double-check that you can connect to the telnet server from your Ubuntu itself using localhost   
(i.e. telnet localhost).

Terminate your telnet session first.

1. At your Kali, run Wireshark and start capturing all traffic.
2. Connect to the telnet server running on Ubuntu again, and log in.
3. You can terminate your Wireshark’s packet capturing.
4. Enter telnet to the display-filter field to see only telnet data (and not TCP 3-way handshake packets, for instance). Inspect the telnet traffic.
5. (Optional) Start another Wireshark’s packet capture session.   
   Connect to your Ubuntu from Kali again, but now using SSH.   
   Inspect the SSH traffic.   
   Can you still see the shell commands sent in the SSH session?

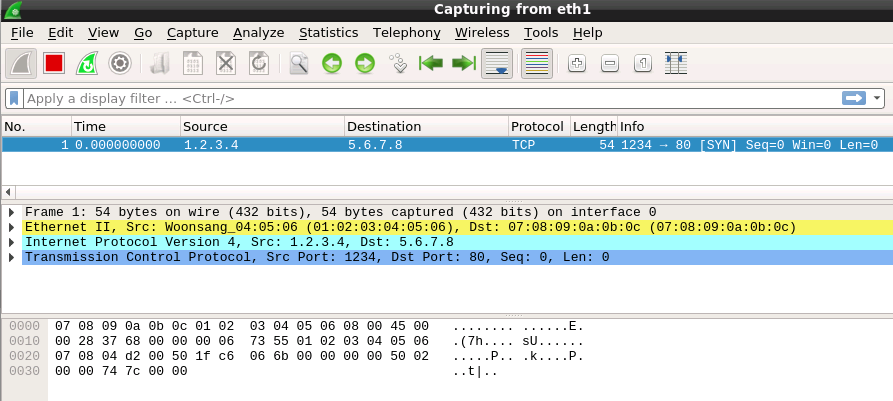
**B. Packet Spoofing**

The goal of this task is to perform a **packet spoofing** using netwox tool.

1. Spoof a SYN packet using netwox tool number **36**:

sudo netwox 36 -d enp0s3 -a a:b:c:d:e:f -b f:e:d:c:b:a -l 1.2.3.4 -m 5.6.7.8 -o 1234 -p 80 -C

Inspect the output of the tool.

1. Check Wireshark to see that the crafted packet is indeed sent out.

Does this packet reach its destination?

1. (Optional) Try hping3 and nping, other packet generators available in Kali, for the same task and compare their capabilities.

**C. Telnet Session Resetting**

The goal of this task is to **reset** a telnet session using netwox.

1. Run your Wireshark at Kali for another packet capturing session.
2. Establish a telnet session from your Kali to your Ubuntu. Here, we use 192.168.56.101 (Kali) and 192.168.56.102 (Ubuntu) with device named enp0s8 as our example.
3. At Kali, run the following command, which utilizes netwox’s tool 78 to reset every TCP packet:

sudo netwox 78 -f "dst host 192.168.56.102" -d enp0s8 -s best

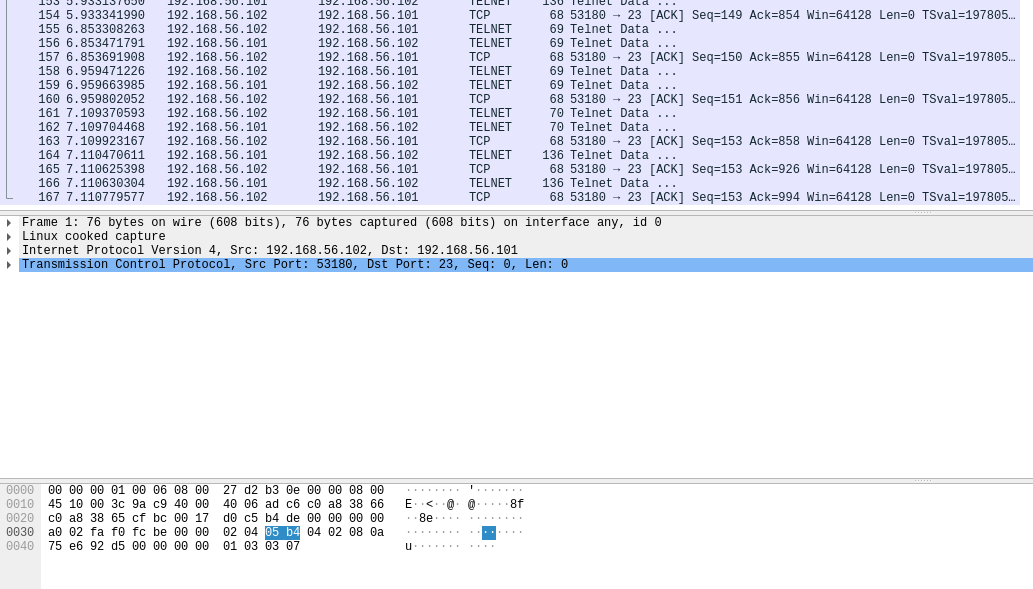
Check the telnet client whether the connection session is now lost.

1. Inspect the captured traffic in Wireshark.
2. Don’t forget to kill netwox tool after you finish.

**D. Telnet Session Hijacking**

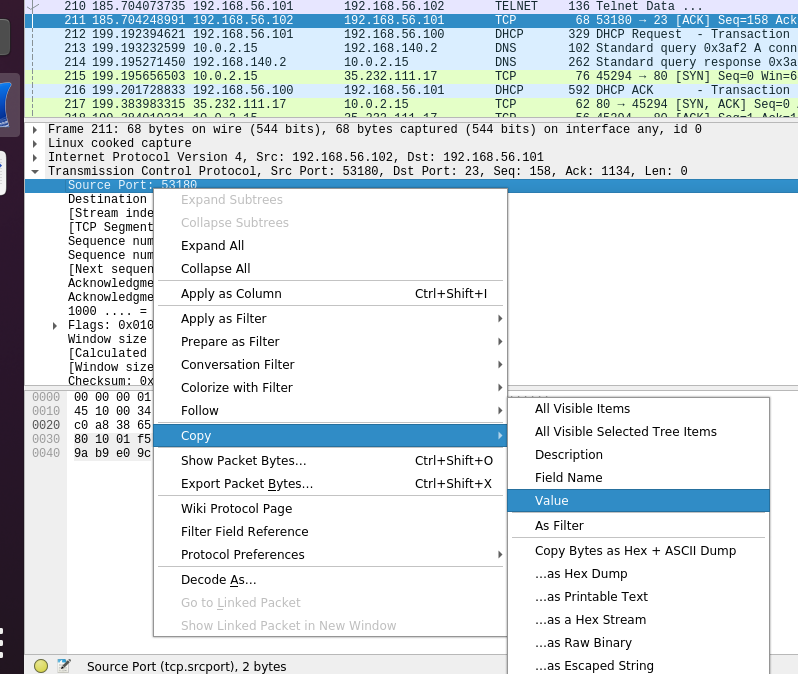
The goal of this task is to **inject a command** to a telnet session.

1. Start a new packet capturing using Wireshark at Kali.
2. Establish a telnet connection from your Kali to your Ubuntu.
3. Using Wireshark at your Kali, search for the last TCP ACK packet sent by your Kali to the telnet server. (Note: don’t use telnet filter   
   in Wireshark as the packet you want will be filtered out.)
4. Right-click on the TCP portion of the Wireshark pane displaying the decoded packet. Select a menu item "Protocol Preference". In its sub-menu, uncheck the "Relative Sequence Number" option. In this way, we get the **absolute** sequence and acknowledgement numbers.



1. netwox’s tool 40 can be used for spoofing an IPv4 packet.   
   Modify the following command according to your TCP session.

netwox 40 --ip4-dontfrag --ip4-offsetfrag 0   
--ip4-ttl 64 --ip4-protocol 6 --ip4-src *192.168.56.102* --ip4-dst 192.168.56.101 --tcp-src *53180* --tcp-dst 23 --tcp-seqnum *153*   
--tcp-acknum *994* --tcp-ack --tcp-psh   
--tcp-window *501* --tcp-data "'hostname'0d0a"   
--spoofip "best"

The values of the required numbers can be copied from Wireshark:   
just right-click on a selected field, and then choose Copy → Value.

1. Explore newly-added packets in Wireshark.

Can you find the spoofed packet and the reply from the server?   
What is the output of the command?

The telnet server will keep sending data with hostname output.   
How could the retransmission be stopped?

After the hijacking attack, the telnet client will be frozen without any response. If you type something on the client, you can find out that server is sending many TCP Dup ACK and TCP Retransmission packets to the client because the client still want to use a “new” sequence number (from the client’s viewpoint) which actually has already been used by the attacker.