CSCI 530 - Lab-5

Rishit Saiya (rsaiya@usc.edu)

1. The number of bytes in the echo protocol exchange in section 4 above, according to the "Follow TCP Stream" window, is 14.

2. The number of bytes in the echo protocol exchange in section 4 above, according to the Statistics/Conversation List/TCP window, is 888 bytes.

3. The iptables command syntax to create a firewall rule prohibiting use of the standard echo protocol (Section 6 above) is:

# Netfilter (iptables) rules for tcp.pcap, packet 4. Change eth0 to a valid interface if needed.

# IPv4 source address.

iptables --append INPUT --in-interface eth0 --source 192.168.1.2/32 --jump DROP

# IPv4 destination address.

iptables --append INPUT --in-interface eth0 --source 192.168.1.1/32 --jump DROP

# Source port.

iptables --append INPUT --in-interface eth0 --protocol tcp --source-port 38096 --jump DROP

# Destination port.

iptables --append INPUT --in-interface eth0 --protocol tcp --source-port 7 --jump DROP

# IPv4 source address and port.

iptables --append INPUT --in-interface eth0 --protocol tcp --source 192.168.1.2/32 --source-port 38096 --jump DROP

# IPv4 destination address and port.

iptables --append INPUT --in-interface eth0 --protocol tcp --source 192.168.1.1/32 --source-port 7 --jump DROP

# MAC source address.

iptables --append INPUT --in-interface eth0 --mac-source 08:00:27:ab:7e:1c --jump DROP

# MAC destination address.

iptables --append INPUT --in-interface eth0 --mac-source 08:00:27:26:a4:44 --jump DROP

4. OMITTED – Do not answer this question

5. The number of frames in section 9's datastream was 1350.

6. The average length/size (in bytes) of the frames in section 9's datastream was 1012.45 bytes.

7. The most common frame size among the frames in section 9's datastream was 1514 bytes.

8. The maximum frame size among the frames in section 9's datastream was 1514 bytes.

9. For any of those max-sized frames, the size of its ethernet payload portion was:

Assuming payload consists of FTP data and headers as well, then 1500 bytes which is calculated with 1460 bytes of FTP Data and 20 bytes each of IP and TCP. If payload is assumed as only the FTP data, then it would be 1460 bytes.

10. For that frame, the size of the remainder of the packet (ie, its header) was 14 bytes (MAC Header).

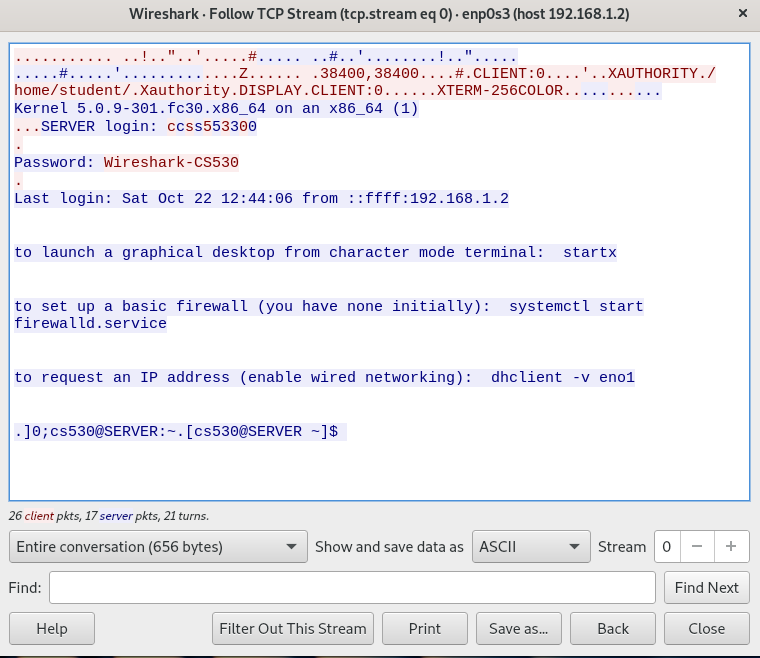
11. For that frame (and all the others like it) Wireshark names its highest-level payload (see the packet details pane). It's FTP stream data.

12. The observed value of the maximum frame size is interesting. It could not be any larger because (consider the reference graphic that follows):

The observed value of maximum size is 1514. In this case MAC header size is 14 bytes. FTP data size is 1460 bytes, IP and TCP have 20 bytes each. So it reaches 514 bytes (14 + 1460 + 20 +20). Further there are 4 bytes reserved for CRC checksum. So now the total size has become 518 (514 + 4) bytes. This is done mainly so that inside the network, a single node is not occupied for a longer duration. As per RFC, Ethernet type 2 frame maximum size could not be larger than 518 bytes. In our case it has already reached maximum size. Therefore maximum size could not be any larger.

13.

**Section-7 (telnet login showing cleartext password):**



**Section-10 ("snakeoil" tls decrypt):**

