Wireshark Exercises

Use the input file **wireshark.pcap** for the following questions:

1. What 3 protocols can be found in this pcap?
2. How many different IP addresses were involved in the conversations in this pcap?
3. What is the largest number of bytes exchanged in an IPv4 conversation?
4. What is the duration (in seconds) of the longest conversation?
5. What is the hexadecimal value of the Ethernet Type in the first packet?
6. What is the IP TTL value in the fourth packet?
7. What Transport layer protocol is being used in the fourth packet?
8. What is the hexadecimal byte value of the first byte in the TCP header? What does this represent, and what is the decimal value?
9. Follow the MySQL TCP conversation. What version of Ubuntu is being used (Ubuntu.???)

Use the input file **link.pcap** for the following questions:

1. In the first packet, what is IP address 192.168.11.11 trying to find?
2. What MAC address is this request sent to? Why is that address used?
3. What is the Ethernet Type for a ARP request?
4. What is the MAC address for 192.168.11.1?
5. Look at packet 3 through 5, what do you think is happening?
6. Look at packet 6 through 55, what are the first 3 source MAC addresses?
7. Look at packet 6 through 55, what are the first 3 source IP addresses?
8. What is the problem with the MAC and IP addresses you examined in question 6 and 7? What might this be an indication of?

Use the input file **ipv4.pcap** for the following questions:

1. There are two problems with the IP header in the second packet. What are they and what will happen to the packet at its destination?
2. What conflicting fields are in the IP header of the third packet?
3. Packet 5-7 are related fragments. These packets are sent over Ethernet (Which has a MTU of 1500). What two abnormalities are in all of the packets?

Wireshark Answers

wireshark.pcap Questions

1. IRC, SSH, and MySQL
2. 4, they are 192.168.88.73, 192.168.88.56, 192.168.88.46, 192.168.88.78
3. 31071 bytes
4. 776.1629 seconds, can be seen on the Conversations screen if you scroll to the right.
5. 0x08000
6. TTL = 64
7. TCP
8. 0x9792, this represents the source port of 38802
9. Ubuntu 5.8

link.pcap Questions

1. The first packet is an ARP request. 192.168.11.11 is trying to find the MAC address for 192.168.11.1
2. The destination MAC address is ff:ff:ff:ff:ff:ff which is the ARP broadcast address.
3. 0x0806
4. The MAC address for 192.168.11.1 is 00:0c:29:03:23:19
5. In packet 3, 192.168.11.44 is looking for the MAC address for 192.168.11.111. In packet 4 192.168.11.111 responds with the MAC address 00:0c:29:0c:23:19. In packet 5 another host with IP 192.168.11.111 responds with IP aa:bb:cc:ee:dd:ff . This is an example of attempted ARP cache poisoning where the attacked spoofed this frame to poison the cache.
6. The first three source MAC addresses are:

   Packet 6: 67:aa:17:2f:ba:02

   Packet 7: ac:1d:9d:2a:7c:71

   Packet 8: c6:58:a2:5e:02:49

7. The source IP address for packets 6 through 55 is 10.10.10.5
8. All frames with the same source IP address should have the same MAC address. This is not the case for packet 6-55. This would be an indication that these Ethernet frames were spoofed.

Ipv4.pcap Questions

1. The IP version in the header is listed as ‘8’ which is invalid. The other problem in an invalid checksum which Wireshark highlighted for us. Both of these conditions would cause the packet to be dropped at the destination.
2. The Do Not Fragment (DF) flag is set, and yet there is a non-zero fragment offset value which is an indication of a fragmented packet.
3. All three packets have a fragment offset of 16, yet they have different payloads. This would result in overlapping fragments which is not normal. Additionally, each legitimate fragment should have a unique offset value. The second problem is that the IP header shows a total length of 1516 bytes per packet on a network that only supports 1500 byte packets.