Quizz1b student name: Evan Smith

1(10): without option part, normally what will be the size of the header of IP packet? how to get to the data part in IP packet giving a pointer to the beginning of a IP packet? (you need ip header graph.3.2 IP protocol).

Normal size of the IP header is 20 bytes. You can use the total length field to determine where the data starts, since we know the size of the header.

2(5) what is TTL in IP header? what is the uses of TTL and how to use it ?

This is the Time To Live, which is the maximum number of routers that the packet can pass through before giving up. This prevents a continuous loop where a packet is moving in circles and never reaches a destination.

3: (5) is there limit length of IP packet? What is that and the reason?

The length limit of IP Packets is 65535 bytes because the Max Length field in the header is only 16 bits, so that is the largest value it can represent.

4(5) What is IP fragmentations? What cause that?

IP fragmentation is when the next routes maximum transmission unit is smaller than the input packet size. If possible (the flags are aligned), the router splits the data into smaller fragments for the receiver to reassemble. This is caused when the payload is too large for the downstream router.

5(10): among IP fragments for one big IP packet, what part is really fragmented? Among all those ip fragments, what do they have in common?

The data is what is actually fragmented, but the header is basically the same, with the exception of labelling which fragment the associated data is.

6(10) how to design attacks according to the method of fragmentation( 2-3 examples)?

* Can make DOS-style attack that sends packets that are known to be larger than the target network and are set up such that the host can’t actually reassemble any of them. This can overwhelm the network. You can do this with incomplete packets or misleading headers.
* You can send smaller packets than would be expected for fragments with the intent to deny service.

7(10) what is ICMP stands for? How many purpose of ICMP? What is ping’s purpose?

ICMP is Internet Control Message Protocol. It is used to provide informational messages about IP packets, such as success or failure messages. Ping is mostly used to test connections to remote hosts and determine whether the host is alive and if there is data loss when sending information over the linkage.

8: (10)what are the types of ping ICMP in our lab1 (lecture code)? What are they? How to set it ?

There are many settings for ping, but I most commonly used the -c flag to control how many packets are sent, and the -a flag which needs the host to be resolved. You can set these flags in the shell when executing a ping.

9: (10) explain what is Spoof ICMP ECHO request in our lab1? what you need to do to achieve it(pseudo code should work here) and how to test your work?

To spoof an ICMP echo, you need to create an ICMP packet, set the destination IP to some address, and set the type to ECHO\_REQUESTPACKET. You can test that change by sniffing on the destination IP to see if a request comes in from the source IP, and you should see a response from the destination, provided it is alive.

10: (10): what is our snoof in lab1? what do you need to do in order to achieve snoof(action sequence)? what is the difference between spoof in this task and spoof in the previous task?(spoof ICMP Echo request). How to test your spoof?

Snoofing is both sniffing and spoofing. The difference here is that the destination IP, source IP, and timing are no longer arbitrary, and we determine those values by sniffing on the network. Additionally, in the lab we are spoofing an Echo response rather than a request.

The general steps would be to have a sniffer on one machine listening for echo requests. When it finds one, it must then spoof an Echo response using the source and destination IP addresses (swapped, of course), and send that immediately back to the host. This will look as if the intended recipient responded, when that is not the case.

11: (5) What are those lines of codes doing below? In sniff or spoof do you need access source or destination address, why?

char Buffer[length];

struct ipheader \* ip = (struct ipheader\*)buffer ;

ip->iph\_sourceip.s\_addr = inet.addr(SRC\_IP);

ip->iph.destip.s\_addr = inet.addr(DES\_IP);

These lines of code are setting the source and destination IP address of an IP header. In sniffing you may need to know the source address, if you are trying to sniff specific hosts, while you will need access to both addresses for spoofing, since you need to set the target as well as fake your own identity.

12(10): how to get TCP header length?? How to move \*tcp(beginning of the tcp) to the data part?

The TCP header has a field Data Offset which is equivalent to the header length. To move the \*tcp to data you need to add the data offset (in bytes) to the current pointer.