CIS/CSE644: quizz1a student name: Anthony Redamonti

1(10): How can you get the address of a variable in C language? Get the address of a normal integer variable.

Int i =10;
Address of i= ?

If you want to use a pointer to hold the value of this address, first try to declare a pointer(what type?) and use it to save the address of i. Then print out the value of (int i) using this pointer you just declared. (do not need a compiling code, just the major lines).

Answer: To get the address of a variable in C, use the "&" operator. Store the address in a pointer to the variable. The pointer must be of the same type as the variable.

int* intPntr = &i; // address is stored in the pointer
printf("Value of integer i is: %d", *intPntr); // print the integer stored at this address

2:(10) what does "(int*) &a" mean: ("a" can be any type except integer type).

Answer: "(int*) &a" means cast the pointer to variable "a" to an integer type. For example, if variable "a" is a character type, we would use "int* aToIntPntr = (int*) &a;" to cast the pointer to "a" as an integer type stored in the pointer "aToIntPntr".

3: (10)

```
char buffer[LENGTH];
struct ipheader *ip = (struct ipheader *) buffer;
struct udpheader *udp = (struct udpheader *) (buffer + sizeof(struct ipheader);
Char *data = buffer + sizeof(struct ipheader) + sizeof(struct udpheader);
```

Explain the above code, what is it doing? Also what is type casting in C language?

Answer: The above code is allocating memory for a packet (stored in a buffer) and dividing the packet into its sections: IP header, UDP header, and data.

The first line in the above code is allocating a character array of length "LENGTH". The next line creates a struct pointer of type "ipheader" and points it at the beginning of the buffer in memory.
The next line creates a struct pointer of type "udpheader" and points it in the buffer at the address immediately after the ipheader portion.
The last line of code creates an array of characters called "data" and points the array in the buffer at the address immediately after the udpheader portion.

Type casting in C is when a variable of a particular type is changed to another type. It can be done explicitly using the casting operator.

3b(5): What does sizeof() return; in what unit?

Answer: The sizeof() function in C returns the number of bytes that a data type occupies in memory.

4:(10) Mention the steps you need to think about in order to write a sniffer? Give the reasons for each step.(briefly)

Answer: Each machine on the network has a NIC card that interfaces with the physical network. Each packet received by the NIC card contains a MAC address for the source machine and destination machine.

1. As the superuser, turn on promiscuous mode on the NIC card.
2. Use socket programming to receive the packets. Use a RAW socket.
3. Filter the incoming messages to receive only packets of a certain type.
4. Create an infinite loop to receive packets (recvfrom blocking function).
5. Create a function that processes the packet. It should provide information like “From IP address:” and “To IP address:” for each packet.

4b:(10) What is promiscuous mode? What is normal mode? How do these two modes make difference in sniff tool? What is necessary for sniffing and why?

Answer: If a NIC card is in normal mode, it will discard packets whose destination MAC address do not match the MAC address on the machine. Most NIC cards have a promiscuous mode that allow the monitoring of all packets on the network. If a card is in promiscuous mode, it will not discard a packet whose MAC address does not match the MAC address of the machine. Sniffing depends on the promiscuous mode.

5:(10) What is raw socket? What is special about raw socket?

Answer: A raw socket is a useful socket for attack purposes. By using a raw socket, the user is informing the system to not format any of the fields in the packet. Then the attacker can customize each field in the packet.

5b(5): What is critical in the function: send_raw_ip_packet()? (without it, the packet will not be sent), and explain the reason?

Answer: sendto(sock, ip, ntohs(ip->iph_len), 0, (struct sockaddr *)&dest_info, sizeof(dest_info)); The sendto function will send the packet to the destination IP address. Without it, the packet would not be sent on the network. Other steps before this are important, like creating a raw socket and providing the destination information.

6:(10) What is the relationship between raw socket programming and pcap library programming? Why do we need the root privilege to run?

Answer: The PCAP library has useful methods for raw socket programming. Instead of implementing methods to handle the low-level packet transmission and filtering, use the PCAP API to perform these functions.

We need to be the superuser to turn on promiscuous mode on the NIC card. Only the “root” or superuser will have this ability.

7:(20) Describe the packet flow through the Internet in high level picture. What is major function of each connection? how does Network data travel through Internet in detail layers? Explain what is in each layer, what's it's function?

Answer: A computer uses an application to generate data. The data will be sent through the computer's operating system onto the TCP/IP stack. The TCP protocol will prepare/format the packet and eventually reach the network interface card (NIC). The card is a piece of hardware connected to the network. The data is transmitted on the Data-Link Layer (Layer 2) from the NIC card to a router, also connected to the network. The Data-Link Layer is a protocol designed for computers that are on the same network to communicate with each other.

The router will send the data to other routers connected to the internet through the Network/IP Protocol (Layer 3). The routers use the Routing Protocol to determine which router to send information. Eventually the data will reach the router that is connected to the server. The router will send the data to the server through the server's NIC. Once the data reaches the server's NIC, the server sends it to the correct application using the Transport Layer TCP/UDP (Layer 4). The transport layer's function is to deliver the data from one application to another.