Class Activity 6

Question # 1:

DHCP:

DHCP (Dynamic Host Configuration Protocol) is a network management protocol used to dynamically assign an Internet Protocol (IP) address to any device, or node, on a network so they can communicate using IP. DHCP automates and centrally manages these configurations rather than requiring network administrators to manually assign IP addresses to all network devices. DHCP can be implemented on small local networks, as well as large enterprise networks.

DHCP will assign new IP addresses in each location when devices are moved from place to place, which means network administrators do not have to manually configure each device with a valid IP address or reconfigure the device with a new IP address if it moves to a new location on the network. Versions of DHCP are available for use in IP version 4 (IPv4) and IP version 6 (IPv6). IPv6 became an industry standard in 2017 -- nearly 20 years after its specifications were first published. While the adoption rate of IPv6 was slow, by July 2019, more than 29% of Google users were making inquiries using IPv6.

DNS:

The Domain Name System (DNS) is a central part of the internet, providing a way to match names (a website you're seeking) to numbers (the address for the website). Anything connected to the internet - laptops, tablets, mobile phones, websites - has an Internet Protocol (IP) address made up of numbers. Your favorite website might have an IP address like 64.202.189.170, but this is obviously not easy to remember. However a domain name such as bestdomainnameever.com is something people can recognize and remember. DNS syncs up domain names with IP addresses enabling humans to use memorable domain names while computers on the internet can use IP addresses.

Types of Attacks:

1. **Denial of service (DoS)**
   An attack where the attacker renders a computer useless (inaccessible) to the user by making a resource unavailable or by flooding the system with traffic.

2. **Distributed denial of service (DDoS)**
   The attacker controls an overwhelming amount of computers (hundreds or thousands) in order to spread malware and flood the victim's computer with unnecessary and overloading traffic. Eventually, unable to harness the power necessary to handle the intensive processing, the systems will overload and crash.

3. **DNS spoofing (also known as DNS cache poisoning)**
   Attackers will drive the traffic away from real DNS servers and redirect them to a “pirate” server, unbeknownst to the users. This may cause the corruption/theft of a user’s personal data.

4. **Fast flux**
   An attacker will typically spoof his IP address while performing an attack. Fast flux is a technique to constantly change location-based data in order to hide where exactly the attack is coming from. This will mask the attacker’s real location, giving him the time needed to exploit the attack. Flux can be single or double or of any other variant. A single flux changes the address of the web server while double flux changes both the address of the web server and names of DNS servers.

5. **Reflected attacks**
   Attackers will send thousands of queries while spoofing their own IP address and using the victim’s source address. When these queries are answered, they will all be redirected to the victim himself.

6. **Reflective amplification DoS**
   When the size of the answer is considerably larger than the query itself, a flux is triggered, causing an amplification effect. This generally uses the same method as a reflected attack, but this attack will overwhelm the user’s system’s infrastructure further.

Measures against DNS attacks:

1. Use digital signatures and certificates to authenticate sessions in order to protect private data.
2. Update regularly and use the latest software versions, such as BIND. BIND is an open source software that resolves DNS queries for users. It is widely used by a good majority of the DNS servers on the Internet.
3. Install appropriate patches and fix faulty bugs regularly.
4. Replicate data in a few other servers, so that if data is corrupted/lost in one server, it can be recovered from the others. This could also prevent single point failure.
5. Block redundant queries in order to prevent spoofing.
6. Limit the number of possible queries.

Question # 2:

Imagine that, as a senior-year prank, high school seniors change out all the room numbers on their high school campus, so that the new students who don't know the campus layout yet will spend the next day getting lost and showing up in the wrong classrooms. Now imagine that the

mismatched room numbers get recorded in a campus directory, and students keep heading to the wrong rooms until someone finally notices and corrects the directory.

DNS cache poisoning is the act of entering false information into a DNS cache, so that DNS queries return an incorrect response and users are directed to the wrong websites. DNS cache poisoning is also known as 'DNS spoofing.' IP addresses are the 'room numbers' of the Internet, enabling web traffic to arrive in the right places. DNS resolver caches are the 'campus directory,' and when they store faulty information, traffic goes to the wrong places until the cached information is corrected. (Note that this does not actually disconnect the real websites from their real IP addresses.)

DNS resolvers provide clients with the IP address that is associated with a domain name. In other words, they take human-readable website addresses like 'cloudflare.com' and translate them into machine-readable IP addresses. When a user attempts to navigate to a website, their operating system sends a request to a DNS resolver. The DNS resolver responds with the IP address, and the web browser takes this address and initiates loading the website.

A DNS resolver will save responses to IP address queries for a certain amount of time. In this way, the resolver can respond to future queries much more quickly, without needing to communicate with the many servers involved in the typical DNS resolution process. DNS resolvers save responses in their cache for as long as the designated time to live (TTL) associated with that IP address allows them to.

Question # 3:

ARP (Address Resolution Protocol) is the protocol that bridges Layer 2 and Layer 3 of the OSI model, which in the typical TCP/IP stack is effectively gluing together the Ethernet and Internet Protocol layers. This critical function allows for the discovery of a devices' MAC (media access control) address based on its known IP address.

By extension, an ARP table is simply the method for storing the information discovered through ARP. It's used to record the discovered MAC and IP address pairs of devices connected to a network. Each device that's connected to a network has its own ARP table, responsible for storing the address pairs that a specific device has communicated with.

ARP is critical network communication, so pairs of MAC and IP addresses don't need to be discovered (and rediscovered) for every data packet sent. Once a MAC and IP address pair is learned, it's kept in the ARP table for a specified period of time. If there's no record on the ARP table for a specific IP address destination, ARP will need to send out a broadcast message to all devices in that specific subnet to determine what the receiver MAC address should be.