Question 1 ***DHCP and ARP*** **()**

Recall that DHCP (Dynamic Host Configuration Protocol) is responsible for setting up configurations when a computer first joins a local network.

Assume that Alice wants to connect to the network, but she's missing a configuration. You want to supply a forged configuration to Alice, causing packets that she intends to send to the network to be sent to you instead.

For this part, assume that your computer has IP address 10.10.10.66 and the network's router and DHCP server have IP address 10.10.10.5. There are no other machines on the network and there are no reserved or private IP addresses.

Q1.1 Alice broadcasts a request for a configuration to everyone.

What values of the assigned IP address and the router's IP address could you include in your response to Alice in order to cause packets that she intends to send to the network to be sent to you instead?

> **Solution:** Any IP address not already in use would work as the assigned IP address. Since there are no other machines on the network, and we are ignoring reserved or private IP addresses, any IP except 10.10.10.66 and 10.10.10.5 is correct.
>
> For the router's IP address, you want to make your own computer the gateway, so that Alice sends any outgoing messages to you first. Thus, the only correct answer is 10.10.10.66 (your IP address).

Q1.2 Imagine that you supply the correct router's IP address in your forged configuration offer to Alice. What other vulnerabilities exist in the DHCP protocol that would still allow you to become a man-in-the-middle attacker?

> **Solution:** Even if you supply the accurate router's IP address, you can still become a man-in-the-middle by manipulating the DNS server address. This lets the you supply malicious translations between human-readable host names (www.google.com) and IP addresses (6.6.6.6) for any website that Alice tries to visit.

Now recall that ARP, the Address Resolution Protocol, translates Layer 3 IP addresses into Layer 2 MAC addresses.

For this part, imagine that Alice has successfully obtained a configuration from the network's router and now she wants to communicate with Bob, whose computer is on the same LAN network.

Alice knows Bob's IP address but wants to learn his MAC address. You want to convince Alice that your MAC address (and not Bob's) corresponds to Bob's IP address, causing messages intended for Bob to be sent to you instead. You, Alice, and Bob are part of the same LAN network and, apart from the router, there are no other machines on the network.

Assume that your computer has IP address 10.10.10.66 and your MAC address is 66:66:66:66:66:66, Alice's IP address is 10.10.10.77 and her MAC address is 77:77:77:77:77:77, and Bob's IP address is 10.10.10.88 and his MAC address is 88:88:88:88:88:88. The network router's IP address is 10.10.10.5.

Q1.1 Alice broadcasts to everyone else on the LAN: "What is the MAC address of 10.10.10.88?" Recall that 10.10.10.88 corresponds to Bob's IP address.

What values for the IP and MAC address could you include in your response to Alice to cause her messages intended for Bob to be sent to you instead?

Solution: You want to convince Alice that your MAC address corresponds to Bob's IP address. Therefore, the only correct answer is 10.10.10.88 as the IP address (Bob's IP) and 66:66:66:66:66:66 as the MAC address (your MAC address).

Q1.2 How would your spoofed response to Alice change if Bob was outside the LAN that you and Alice are both part of?

Solution: If Bob is outside of the LAN, Alice would instead generate an ARP request for the gateway router. The router would respond with its MAC address and forward messages to some other LAN to get it closer to Bob.

Therefore, you are now instead trying to convince Alice that your MAC address corresponds to the router's IP address. For this reason, the only correct answer is 10.10.10.5 as the IP address (the router's IP) and 66:66:66:66:66:66 as the MAC address (your MAC address).

Q1.3 Which defenses exist against such an ARP-spoofing attack?

Solution: A simple defense against ARP spoofing is to use a tool like arpwatch, which tracks the IP address to MAC address pairings across the LAN and makes sure nothing suspicious happens.

Modern wired Ethernet networks defend against ARP spoofing by using switches rather than hubs. Switches have a MAC cache, which keeps track of the IP address to MAC address pairings. If the packet's IP address has a known MAC in the cache, the switch just sends it to the MAC. Otherwise, it broadcasts the packet to everyone. Smarter switches can filter requests so that not every request is broadcast to everyone.

Higher-quality switches include VLANs (Virtual Local Area Networks), which implement isolation by breaking the network into separate virtual networks.

Q1.4 You decide to use a network switch to prevent further ARP spoof attacks.

Explain how a network switch prevents such attacks in the following two cases: (1) the switch knows Bob's IP to MAC address mapping and (2) the switch does **not** know Bob's IP to MAC address mapping.

Solution: Switches have a MAC cache, which keeps track of the IP address to MAC address pairings.

(1) If the packet's IP address has a known MAC in the cache, the switch just sends the packet to the known MAC address.

(2) If the switch does not know the mapping, it simply broadcasts the packet to everyone.

Question 2 ***WPA2-PSK*** **()**

Recall that WPA2-PSK (Wi-Fi Protected Access: Pre-Shared Key) is a protocol that enables secure communications over a Wi-Fi network by encrypting messages with cryptography.

For each of the following questions, assume that Mallory is trying to attack a Wi-Fi network secured with the standard WPA2-PSK protocol.

Q2.1 How does the access point and the client derive the PSK (Pre-Shared Key)?

Solution: The access point derives the PSK by applying a password-based key derivation function on the SSID and the password.

When a computer (client) wants to connect to a network protected with WPA2-PSK, the user must first type in the Wi-Fi password. Then, the client uses the same key derivation function to generate the PSK.

Q2.2 Assume that Mallory can observe the unencrypted data sent as part of the 4-way WPA handshakes between the clients and the access point. Which values does Mallory need to perform a brute-force search for the Wi-Fi password? Select all that apply.

- ■ (G) ANonce
- ■ (H) SNonce
- ■ (I) PSK
- ■ (J) PTK
- □ (K) GTK
- ■ (L) The router's MAC address
- ■ (M) The client's MAC address
- ■ (N) The MICs

Solution:

In the WPA2 4-way handshake, the information dependency goes {SSID, password} $\rightarrow$ PSK + {ANonce, SNonce, Router MAC, Client MAC} $\rightarrow$ PTK $\rightarrow$ MIC. In other words, the SSID and password are used to derive the PSK. Then the PSK, the nonces, and the MAC addresses are used to derive the PTK. Finally, the PTK is used to generate the MIC (message integrity/authentication codes).

To generate a guess for the PTK, you guess a password. Then you use your guessed password and the SSID to generate a guessed PSK. Then you use the guessed PSK, the nonces, and the MAC addresses to guess the PTK. Finally, you generate a MIC with your guessed PTK and see if it matches the MICs in the observed handshake.

Thus we need the nonces, the MAC addresses, PSK, PTK, and the MICs to perform our brute-force attack.

Q2.3 Now imagine that Mallory is trying to masquerade as the access point in order to launch a Man-in-the-Middle (MITM) against Alice, who is trying to join the Wi-Fi network. How can Mallory trick Alice into thinking that she is the access point?

Solution: Mallory needs to complete the 4-way handshake with Alice and prevent the genuine access point from doing the same.

In order to complete the handshake, Mallory needs to generate and offer her own ANonce to Alice. Mallory also needs to know PSK or be able to derive it from the Wi-Fi password and the SSID. To complete the handshake, Mallory would use the PSK, the nonces, her and Alice's MAC addresses to derive the PTK. Using the PTK, Mallory can then generate and send the MIC and GTK, thus, successfully completing the handshake.

Q2.4 What can Mallory do after successfully brute-forcing the Wi-Fi password? Select all that apply.

■ (G) Perform on-path network attacks against victims in the same Wi-Fi network.

■ (H) Decrypt network traffic encrypted with the PTK of a user who joins the network after you.

■ (I) Decrypt network traffic encrypted with the GTK

☐ (J) Decrypt TLS network traffic.

☐ (K) None of the above.

☐ (L) ——

Solution: Mallory is on the local network, so she can enable sniffing mode and see packets of other people in the same Wi-Fi network. This makes her an on-path attacker.

With her brute-force attack, she now knows the Wi-Fi password. As mentioned above, Mallory is an on-path attacker, so when someone joins the network after her, she can observe their handshake. Mallory uses the SSID and her brute-forced password to derive the PSK. She uses the nonces and the MAC addresses (sent in plaintext during the handshake, so Mallory knows their values), along with the PSK to derive the PTK. Now she can decrypt the victim's messages with their PTK.

During the 4-way handshake, the access point sends the GTK encrypted with the PTK. Mallory has the PTK, so she can decrypt the GTK and then use it to decrypt any network traffic encrypted with the GTK. Alternatively, since Mallory knows the password, she can just initiate a WPA2 connection herself and get the GTK value from the access point.

TLS is end-to-end secure, so being an on-path attacker in the local network won't help Mallory decrypt TLS traffic.

Q2.5 Assume that Mallory can brute-force the Wi-Fi password in roughly an hour. Which defenses would stop this attack? Select all that apply.

☐ (A) Changing the Wi-Fi password every day.

■ (B) Using WPA2-Enterprise.

☐ (C) A modern NIDS system.

☐ (D) None of the above.

☐ (E) ——

☐ (F) ——

Solution: Changing the password each day won't work if the attacker can brute-force the Wi-Fi password in less than a day.

A NIDS system protects a local network from external attacks, but does not stop attacks from local attackers.

WPA2-Enterprise is a good defense against this attack, because the attacker wouldn't be able to authenticate itself to the third-party server.

Q2.6 Does WPA-Enterprise offer forward secrecy? If it does, explain how forward secrecy is achieved. If it does not, explain how Mallory can recover the shared PTK (Pairwise Transport Key) after the handshake is complete?

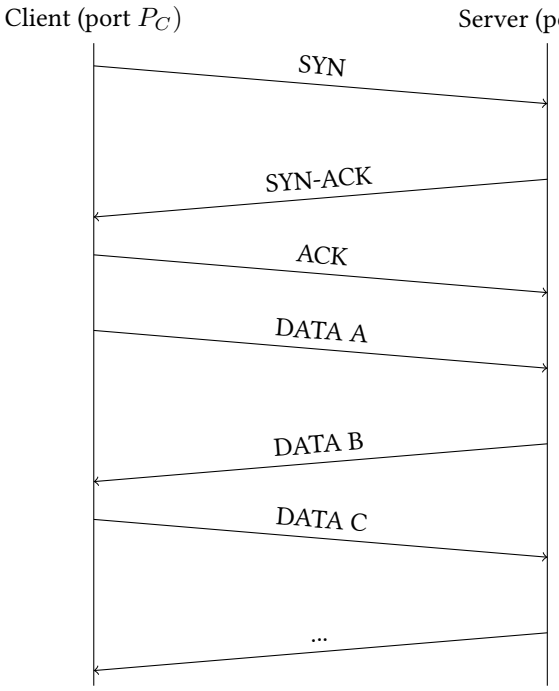
Solution: WPA-Enterprise does offer forward secrecy.

In the standard WPA2-PSK, an eavesdropper who records the values of ANonce and SNonce can derive the key if they later learn the password or PSK.

On the other hand, in WPA-Enterprise the generated PSK replacement, known as the PMK (Pairwise Master Key), is used once and then discarded, so no information is retained that allows the PTK to be recovered later.

Question 3 ***Attack on TCP*** ()

Suppose that a client connects to a server, and then performs the following TCP handshake and initial data transfer:



1. Client sends initial SYN with sequence number A (usually random).
2. Server sends SYN-ACK with sequence number B (also usually random) and ACK $A + 1$.
3. Client sends ACK with sequence number $A + 1$ and ACK $B + 1$.
4. Client sends DATA A of length L_A with sequence number $A + 1$ and ACK $B + 1$.
5. Server sends DATA B of length L_B with sequence number $B + 1$ and ACK $A + 1 + L_A$.
6. Client sends DATA C of length L_C with sequence number $A + 1 + L_A$ and ACK $B + 1 + L_B$.
7. Data exchange continues until both sides are done sending data.

Q3.1 Assume that the next transmission in this connection will be DATA D from the server to the client. What will this packet look like?

Sequence number:	$B + 1 + L_B$	ACK:	$A + 1 + L_A + L_C$
Source port:	P_S	Destination port:	P_C
Length:	L_D	Flags:	ACK

Q3.2 You should be familiar with the concept and capabilities of a *man-in-the-middle* as an attacker who **can observe** and **can modify** traffic. There are two other types of relevant attackers in this scenario:

1. *On-path* attacker: **can observe** traffic but **cannot modify** it.
2. *Off-path* attacker: **cannot observe** traffic and **cannot modify** it.

Carol is an *on-path* attacker. Can Carol do anything malicious to the connection? If so, what can she do?

Solution: Yes, Carol can leverage the information she learns from your traffic to hijack the session.

In part (a), we identified the values of all of the fields of concern expected in the next data transmission in the connection. Say Carol wants to spoof a packet from the server to the client; Carol can create a packet with the source IP as the server's IP, the destination IP as the client's IP, and the payload as whatever she wants. To spoof traffic in the other direction, she swaps the sequence number/ACK, source port/destination port, and source IP/destination IP. The recipient of this data cannot distinguish it from legitimate traffic, so she has effectively hijacked the session from her victim, allowing her to inject arbitrary data.

Q3.3 David is an *off-path* attacker. Can David do anything malicious to the connection? If so, what can he do?

Solution: No, there isn't much he can do.

In part (b), we demonstrated that we are effectively defenseless against an attacker that knows the *sequence numbers* and the *port numbers* of the connection. An off-path attacker, however, does not have the power to observe the traffic and find these parameters.

Even without prior knowledge of these parameters, though, an *off-path* attacker may attempt to guess them. In a typical TCP client-server connection, the client's port is an *ephemeral* port, with a maximum potential range of $[0, 2^{16} - 1]$ (this varies, so we make an overestimation). The server's port is usually a *well-known* port for a specific service, such as port 80 for HTTP, which makes it much easier to guess. The sequence number and acknowledgement numbers are in the range $[0, 2^{32} - 1]$. However, an attacker can just ignore the acknowledgement number or use a random number. The TCP connection will not be reset with an invalid acknowledgement number as long as it's within the window (out of scope). Thus, an attacker has a rough $\frac{1}{2^{48}}$ chance of successfully brute forcing the correct parameters.

If, for some reason, the initial sequence numbers are not properly randomized, David may be able to make educated guesses on the sequence numbers and significantly decrease the range of possibilities. However, assuming that they are properly randomized, this attack is theoretically possible but largely improbable.

We call this attack *blind hijacking*, as David has no concrete information when attempting to hijack the session.

Q3.4 The client starts getting responses from the server that don't make any sense. Inferring that David is attempting to hijack the connection, the client then immediately sends the server a **RST** packet, which terminates the ongoing connection. David wants to impersonate the client by establishing a new connection. How would he go about doing this?

Solution: If David attempts to start a new connection, he can *choose* the source port (the *ephemeral* port) and the source sequence number to be whatever values he wants. The values of these parameters for any subsequent transmissions in the connection will then be predictable. The server's port remains a well-known port; the only remaining unknown is the server's sequence number. Because David is still an off-path attacker, he still has to guess this field, with an overall probability of $\frac{1}{2^{32}}$ of success, which we note is higher than the *blind hijacking* approach.

Note that there's now a time constraint on David's attack: if the client receives a response from the server based on his spoofed *SYN*, it will send a **RST** and terminate the connection, putting David back at step 1.

We call this attack *blind spoofing*, as David has no concrete information when attempting to spoof a new session.