Weaver
Spring 2022

CS 161
Computer Security

Exam Prep 8

Q1 *Coffee-Shop Attacks* (17 points)

Dr. Yang comes to MoonBucks and tries to connect to the network in the coffee shop. Dr. Yang and `http://www.piazza.com` are communicating through TCP. Mallory is an on-path attacker.

Q1.1 (5 points) Which of the following protocols are used when Dr. Yang first connects to the Wi-Fi network and visits `http://www.piazza.com`? Assume any caches are empty. Select all that apply.

□ (A) CSRF

■ (B) IP

■ (C) DNS (or DNSSEC)

■ (D) HTTP

■ (E) DHCP

□ (F) None of the above

Solution:

A: False. CSRF is not a protocol, but a web attack.

B: True. IP is used to send messages across the internet and is used by TCP, which is used by TLS, which is used by HTTPS.

C: True. DNS is used to look up the IP address of `www.piazza.com`.

D: True. HTTP is the application protocol being used.

E: True. DHCP is used to receive the initial network configuration for the client.

Q1.2 (3 points) Suppose Mallory spoofs a packet with a valid, upcoming sequence number to inject the malicious message into the connection. Would this affect other messages in the connection?

● (G) Yes, because the malicious message replaces some legitimate message

○ (H) Yes, because future messages will arrive out of order

○ (I) No, because on-path attackers cannot inject packets into a TCP connection

○ (J) No, because TCP connections are encrypted

○ (K) ——

○ (L) ——

Solution: When the server receives the original TCP packet whose sequence number was used by Mallory, the server will ignore it, thinking that it has already received its data and that it was retransmitted.

Q1.3 (3 points) To establish a TCP connection, Dr. Yang first sends a SYN packet with Seq $= 980$ to the server and receives a SYN-ACK packet with Seq $= 603$; Ack $= 981$. What packet should Dr. Yang include in the next packet to complete the TCP handshake?

○ (A) SYN-ACK packet with Seq $= 981$; Ack $= 604$

○ (B) SYN-ACK packet with Seq $= 604$; Ack $= 981$

● (C) ACK packet with Seq $= 981$; Ack $= 604$

○ (D) ACK packet with Seq $= 604$; Ack $= 981$

○ (E) Nothing to send, because the TCP handshake is already finished.

○ (F) ——

Solution: This is the third step of the 3-way handshake, when the client sends an ACK packet to acknowledge the server's SYN-ACK packet.

Q1.4 (3 points) Immediately after the TCP handshake, Mallory injects a valid RST packet to the server. Next, Mallory spoofs a SYN packet from Dr. Yang to the server with headers Seq $= X$. The server responds with a SYN-ACK packet with Seq $= Y$; Ack $= X + 1$. What is the destination of this packet?

● (G) Dr. Yang

○ (H) The server

○ (I) Mallory

○ (J) None of the above

○ (K) ——

○ (L) ——

Solution: The server uses the source as the destination for the SYN-ACK packet. Because Mallory spoofed the packet from the client, the response is sent to the client.

Q1.5 (3 points) Which of the following network attackers would be able to perform the same attacks as Mallory?

Clarification during exam: By “perform the same attacks,” we mean “reliably perform the same attacks.”

● (A) A MITM attacker between Dr. Yang and the server

○ (B) An off-path attacker

○ (C) All of the above

○ (D) None of the above

○ (E) ——

○ (F) ——

Solution: A MITM attacker has all the capabilities of an on-path attacker, so it would be able to perform Mallory’s attacks. An off-path attacker would be unable to guess the sequence numbers and would be unable to perform Mallory’s attacks.

Q2 *Pancake Query Protocol* (19 points)

EvanBot is already prepared for the winter break but realizes that there are no more pancakes and needs to order more! To speed up the ordering process, EvanBot crafts a custom Pancake Query Protocol (PQP) and needs to ensure that it is secure.

PQP runs directly over IP, and a PQP packet contains the following information:

- A packet type
- The pancake query data (either a request for an order, or the order itself)

For now, assume that the only packet type supported by PQP is the ORDER type. For example, EvanBot might send the following PQP packet:

- EvanBot ⟶ Restaurant: {Type: ORDER; Data: "I want 1 stack of blueberry pancakes!"}

For all parts, assume that EvanBot knows the IP address of the restaurant. All subparts of this question are independent.

Q2.1 (5 points) Which of the following statements are true about PQP? Select all that apply.

☐ (A) An off-path attacker can learn EvanBot's order

■ (B) An off-path attacker can trick the restaurant into cooking unwanted pancakes for EvanBot

☐ (C) An on-path attacker can conduct a RST injection attack

■ (D) An on-path attacker can learn EvanBot's order

☐ (E) EvanBot can be sure that the restaurant received the order

☐ (F) None of the above

Solution: A: False. EvanBot will always send their order to the restaurant (i.e. not visible to the off-path attacker), since Bot knows the restaurant's IP address.

B: True. An off-path attacker can forge an IP packet as if it were coming from EvanBot and place an malicious order inside the forged PQP packet.

C: False. RST injection attacks don't exist without TCP, since a RST packet is a construct of TCP.

D: True. An on-path attacker can see the contents of the PQP packet and learn the order.

E: False. Raw IP does not provide reliability.

Q2.2 (3 points) EvanBot adds an ACK packet type to PQP packets. After a restaurant receives an order, the restaurant sends an ACK packet acknowledging the order. If EvanBot does not receive the ACK, EvanBot re-sends the order until an ACK is received.

EvanBot tries to order 1 stack of pancakes from the restaurant and eventually receives an ACK. Assume that **no** network attackers are present. How many orders could the restaurant receive?

○ (G) Exactly 0, because IP is unreliable.

○ (H) Either 0 or 1, because IP is unreliable.

○ (I) 0 or more, because IP is unreliable.

○ (J) Exactly 1, because the restaurant ACKs any order it receives.

● (K) 1 or more, because EvanBot might try more than once.

○ (L) 2 or more, because the restaurant may send multiple ACKs.

Solution: EvanBot succeeds in receiving an ACK, sot he restaurant must have received at least one order. However, it is possible that the restaurant received more than one order: One possible sequence of events is that EvanBot tried sending their order multiple times without receiving an ACK. However, the restaurant received all of these orders, but all of the ACK packets they sent back were lost, because IP is unreliable. Eventually, one of the ACK packets got through to EvanBot.

Q2.3 (4 points) Consider the following modification to PQP: To order, the client generates a random order ID and sends it in the PQP ORDER packet along with the order. The server sends back the order ID in the PQP ACK packet.

Can an off-path attacker trick the restaurant into accepting a spoofed order appearing to come from EvanBot? Briefly justify your answer (1–2 sentences).

● (A) Yes

○ (B) No

○ (C) ——

○ (D) ——

○ (E) ——

○ (F) ——

Solution: While including the random order ID makes this seem secure, notice that the restaurant takes action immediately when the order is received. An off-path attacker can thus forge an ORDER packet containing a malicious order and random order ID.

Q2.4 (4 points) Which of the following modifications to PQP, if made individually, would prevent an off-path attacker from tricking the restaurant into accepting spoofed orders? Select all that apply.

■ (G) The restaurant generates a random order ID and sends it back in the PQP ACK. The restaurant must receive a PQP ACK-ACK packet from EvanBot containing the order ID to confirm the order.

□ (H) The restaurant sends a fixed time-to-live (TTL) to EvanBot in the PQP ACK. The restaurant must receive a final, empty PQP ACK packet from EvanBot within the TTL to confirm the order.

□ (I) PQP runs over UDP instead of IP, and EvanBot chooses a random source port.

■ (J) PQP runs over TCP instead of IP.

□ (K) None of the above

□ (L) ——

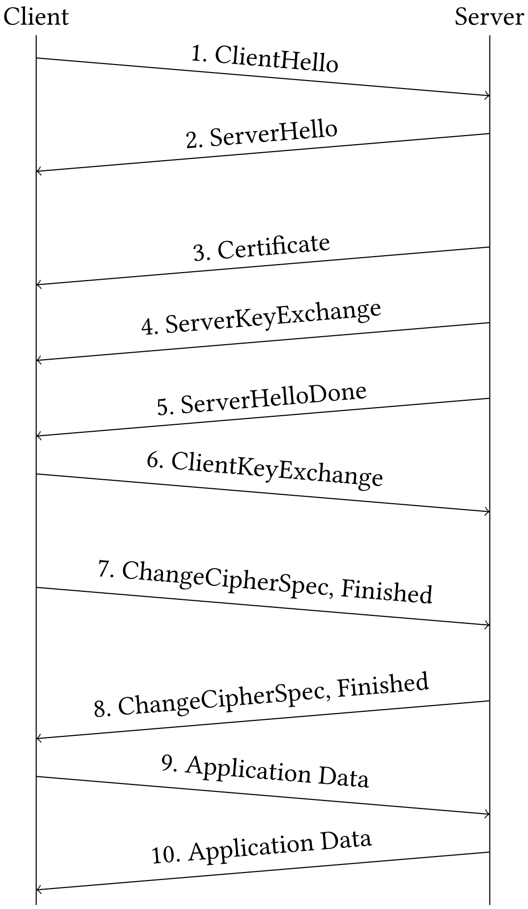
Solution: Having the *restaurant* generate a random order ID, sending it back to the client, and then having the client send that order ID back to the server to confirm the order would work. This ensures that an off-path attacker must guess information that it cannot see, since it can't see the random order ID generated by the restaurant.

Q2.5 (3 points) EvanBot proposes an additional packet types for PQP: LISTORDERS. When a restaurant receives an LISTORDERS packet, it responds with a list of all orders that it has ever received from any customer. Name one security issue with this proposal and describe the steps an attacker should take to exploit this issue (1–2 sentence).

Solution: This creates a classic amplified DoS attack: An attacker can spoof an ORDER_LIST packet appearing to come from a victim IP address, and the restaurant will send a very large response to the victim. This is similar to DNSSEC response packets and the deprecated NTP MONLIST command.

Recall the TLS handshake:

Q3 ***Mutuality*** **(18 points)**



1. Client sends 256-bit random number R_b and supported ciphers

2. Server sends 256-bit random number R_s and chosen cipher

3. Server sends certificate

4. DH: Server sends $\{g, p, g^a \bmod p\}_{K_{\text{server}}^{-1}}$

5. Server signals end of handshake

6. DH: Client sends $g^b \bmod p$
   RSA: Client sends $\{PS\}_{K_{\text{server}}}$

Client and server derive cipher keys C_b, C_s and integrity keys I_b, I_s from R_b, R_s, PS

7. Client sends MAC(dialog, I_b)

8. Server sends MAC(dialog, I_s)

9. Client data takes the form $\{M_1, \text{MAC}(M_1, I_b)\}_{C_b}$

10. Server data takes the form $\{M_2, \text{MAC}(M_2, I_s)\}_{C_s}$

In TLS, we verify the identity of the server, but not the client. How would we modify TLS to also verify the identity of the client?

Clarification during exam: All parts of this question refer to a modified TLS scheme designed to verify the identity of the client.

Q3.1 (3 points) Which of these additional values should the client send to the server?

○ (A) A certificate with the client's public key, signed by the client's private key

○ (B) A certificate with the client's public key, signed by the server's private key

○ (C) A certificate with the client's private key, signed by a certificate authority's private key

● (D) A certificate with the client's public key, signed by a certificate authority's private key

○ (E) ——

○ (F) ——

Solution: This is analogous to the server sending its certificate, which has the server's public key, signed by the certificate authority's private key.

Q3.2 (3 points) How should the client send the premaster secret in RSA TLS?

● (G) Encrypted with the server's public key, signed by the client's private key

○ (H) Encrypted with the client's public key, signed by the server's private key

○ (I) Encrypted with the server's public key, signed by a certificate authority's private key

○ (J) Encrypted with the client's public key, signed by a certificate authority's private key

○ (K) ——

○ (L) ——

Solution: The client should encrypt the premaster secret with the server's public key so that the server can decrypt it (just like in regular TLS).

However, the client should additionally sign the premaster secret, so that the server can validate the signature and confirm that the server is talking to the correct client. The client should sign the premaster secret with their own private key. (The client doesn't know the certificate authority's private key, and the CA's private key is only used to sign certificates anyway.)

Q3.3 (3 points) EvanBot argues that the key exchange protocol in Diffie-Hellman TLS doesn't need to be changed to support client validation. Is EvanBot right?

○ (A) Yes, because only the client knows the secret a, so the server can be sure it's talking to the legitimate client

○ (B) Yes, because the server has already received and verified the client's certificate

● (C) No, the client must additionally sign their part of the Diffie-Hellman exchange with the client's private key

○ (D) No, the client must additionally sign their part of the Diffie-Hellman exchange with the certificate authority's private key

○ (E) ——

○ (F) ——

Solution: Diffie-Hellman on its own doesn't provide any authenticity. We also need the client to sign their Diffie-Hellman message.

Q3.4 (2 points) TRUE or FALSE: The server can be sure that they're talking to the client (and not an attacker impersonating the client) immediately after the client and server exchange certificates.

○ (G) True ● (H) False ○ (I) —— ○ (J) —— ○ (K) —— ○ (L) ——

Solution: False. Remember that certificates are public, and attackers can present a certificate for anyone. The ClientHello and ServerHello messages only contain random nonces and an agreement on what algorithms to use, so they also do not give the client and server any guarantees about who they're talking to.

The client and the server need to wait at least until the signatures are exchanged to verify that they're talking to the correct person. If an attacker tampers with the handshake, the client and the server may even have to wait until the MACs are exchanged.

Q3.5 (3 points) At what step in the TLS handshake can both the client and server be sure that they have derived the same symmetric keys?

○ (A) Immediately after the TCP handshake, before the TLS handshake starts

○ (B) Immediately after the ClientHello and ServerHello are sent

○ (C) Immediately after the client and server exchange certificates

○ (D) Immediately after the client and server verify signatures

● (E) Immediately after the MACs are exchanged and verified

○ (F) ——

Solution: The reasoning here is the same as in regular TLS. A MITM could tamper with messages, and the client and server will only detect this once they verify the MAC on the entire handshake.

Q3.6 (4 points) Which of these keys, if stolen individually, would allow the attacker to impersonate the client? Select all that apply.

■ (G) Private key of a certificate authority

■ (H) Private key of the client

□ (I) Private key of the server

□ (J) Public key of a certificate authority

□ (K) None of the above

□ (L) ——

Solution: If the attacker steals the private key of a trusted CA, they can sign a fake certificate claiming that the attacker's public key belongs to the client.

If the attacker steals the private key of the client, they can sign messages as the client.

Stealing the public key of the server doesn't help the attacker impersonate the client.