Final

111061610 謝崇志

1. True or False? Explain why? (50 points)

(a) Suppose an opponent captures an unexpired service granting ticket and modifies its network address to match that of the valid user. Is it correct that the opponent will be granted access to the corresponding service.

Ans:

False. If an opponent captures a service granting ticket and modifies its network address to match that of the valid user, the opponent will not be granted access. Most authentication systems use more than just the network address to verify the identity of a user.

(b) If the lifetime stamped on a ticket is very short (e.g., minutes) an opponent has a greater opportunity for replay.

Ans:

True. If the lifetime stamped on a ticket is very short, it provides a smaller window of opportunity for an opponent to capture and replay the ticket within that time frame.

(c) User certificates generated by a CA need special efforts made by the directory to protect them from being forged.

Ans:

False. User certificates generated by a Certificate Authority (CA) are inherently designed to be resistant to forgery. The CA's role is to verify the identity of the entity requesting the certificate, making it difficult to forge.

(d) In IPsec, authentication must be applied to the sections of the original IP packet. (e) The principal feature of IPsec is that it can encrypt and/or authenticate all traffic at the IP level

Ans:

True. In IPsec, authentication is applied to the sections of the original IP packet, ensuring the integrity and authenticity of the transmitted data.

(e) The principal feature of IPsec is that it can encrypt and/or authenticate all traffic at the IP level.

Ans:

True. The principal feature of IPsec is that it can encrypt and/or authenticate all traffic at the IP level, providing security for communication at the network layer.

(f) A stateful packet inspection firewall reviews the same packet information as a packet filtering firewall but also records information about TCP connections.

Ans:

True. A stateful packet inspection firewall reviews the same packet information as a packet filtering firewall but goes beyond by keeping track of the state of active connections, allowing it to make more informed decisions about which packets to allow or block.

(g) Packet filter firewalls examine upper layer data therefore they can prevent attacks that employ application specific vulnerabilities or functions.

Ans:

False. Packet filter firewalls operate at the network layer and examine lower-layer data, such as IP addresses and port numbers. They are not designed to inspect the content of the upper layer (application layer) data, making them less effective against attacks exploiting application-specific vulnerabilities.

(h) The change cipher spec protocol exists to signal the transition in ciphering strategies

Ans:

True. The Change Cipher Spec protocol in SSL/TLS is used to signal the transition in ciphering strategies during the secure communication establishment.

(i) Transport mode in IPSec provides security to the entire IP packet.

Ans:

False. In IPSec, transport mode provides security for the upper-layer protocols and the payload but does not protect the entire IP packet. Tunnel mode is the one that encapsulates the entire original IP packet for secure transmission.

(j) In tint fragment attack, it is possible for an intermediate fragment to pass through filter before the initial fragment is rejected.

Ans:

True. In a "tint" fragment attack (assuming a typo and meant to be "tiny" fragment attack), it is possible for an intermediate fragment to pass through a

filter before the initial fragment is rejected, potentially leading to security vulnerabilities.

2. Suppose Alice has a secret a and Bob has a secret b. Alice and Bob also have access to a secure communication channel. The goal is to let a server s compute $sum = a \oplus b$. However, they don't want s learn a or b. How can they achieve this. (10 points)

Ans:

First:

- Alice and Bob privately generate random values. Let's call Alice's random value r_a and Bob's random value r_b
- They both send their random values to the server s

Computation on Server:

- The server s computes $r_a \oplus r_b$ to get a random value r_{ab}.

Communication:

- The server s sends r_{ab} back to both Alice and Bob.

Final result:

- Alice computes $a \oplus r_a$ to get a', and Bob computes $b \oplus r_b$ to get b'.
- They both send their results a' and b' to the server s.

Server's Computation:

- The server s computes $a' \oplus b'$ to get the final result, which is $a \oplus b$,without ever knowing a or b individually.

The idea here is that the XOR operation is commutative and associative, meaning that $a \oplus b = b \oplus a$ and $(a \oplus b) \oplus c = a \oplus (b \oplus c)$. By introducing random values r_a and r_b, the server s can perform the XOR operation without learning the actual values of a or b.

3. Consider the following protocol:

$$A \rightarrow KDC : ID_A||ID_B||N_1$$
$$KDC \rightarrow A : E(k_a, [k_s||ID_B||N_1||E(k_b, [k_s||ID_A]))$$
$$A \rightarrow B : E(k_b, [k_s||ID_A])$$
$$B \rightarrow A : E(k_s, N_2)$$
$$A \rightarrow B : E(k_s, f(N_2))$$

(a) Explain the protocol. (5 points)

Ans:

A requests session key:

- Alice (A) initiates communication by requesting a session key from the Key Distribution Center (KDC).

- Message: $A \rightarrow KDC : ID_A||ID_B||N_1$

KDC Generates Session Key:

- The KDC generates a session key (k_s) and encrypts it using Alice's key (k_a), along with Bob's ID (ID_B), a nonce (N_1), and a nested encryption using Bob's key (k_b) with Alice's ID (ID_A).
- Message: $KDC \rightarrow A : E(k_a, [k_s||ID_B||N_1||E(k_b, [k_s||ID_A]))$

A Sends Encrypted Message to B:

- Alice sends an encrypted message to Bob containing the session key (k_s) encrypted with Bob's key (k_b) and her ID (ID_A).
- Message: $A \rightarrow B : E(k_b, [k_s||ID_A])$

B Responds with Encrypted Nonce:

- Bob responds to Alice with an encrypted nonce (N_2) using the session key (k_s).
- Message: $B \rightarrow A : E(k_s, N_2)$

A Sends Encrypted Function to B:

- Alice sends an encrypted function ($f(N_2)$) to Bob using the session key (k_s).
- Message: $A \rightarrow B : E(k_s, f(N_2))$

(b) Can you think of a possible attack on this protocol? Explain how it can be done. (5 points)

Ans:

Yes this protocol is vulnerable to attacks. One possible attack is a man-in-the-middle (MITM) attack during the initial communication between A and the KDC. An attacker intercepts the request from A to the KDC and replaces it with their own request, initiating a session with the KDC using their own identity. The attacker then relays the messages between A and B, capturing the session key.

(c) How can we prevent this attack? just write the basics of the idea. (5 points)

Ans:

To prevent the MITM attack, public key cryptography can be used. The KDC's public key can be distributed securely to all parties. When A requests a session key, it can encrypt the request with the KDC's public key. Only the KDC, possessing the corresponding private key, can decrypt and respond. This ensures that A is communicating directly with the legitimate KDC and prevents a malicious party from intercepting and modifying the communication.

4. **IPSec documentation allows combining the security associations. Discuss why is it recommend to perform ESP protocol before AH protocol? (10 points)**

Ans:

When combining security associations in IPsec, it is generally recommended to perform ESP (Encapsulating Security Payload) protocol before AH (Authentication Header) protocol. Here's why:

Confidentiality and Integrity Protection:

- **ESP provides both confidentiality (encryption) and integrity protection, making it suitable for securing the payload of the IP packet.**
- **AH, on the other hand, primarily provides integrity protection but does not offer confidentiality. It ensures the integrity of the entire packet, including the IP header**

Traffic Confidentiality:

- **If confidentiality is a primary requirement, ESP is the preferred choice. It encrypts the payload, ensuring that the actual data remains confidential.**
- **AH, being focused on integrity protection, does not encrypt the payload and, therefore, does not provide traffic confidentiality.**

Order of Operations:

- **ESP operates on the payload of the packet and can encapsulate the entire packet or just the payload. This makes it suitable for encrypting and authenticating the actual data.**
- **AH operates on the entire packet, including the IP header, and may not work correctly if the packet has been encrypted or modified by ESP. Therefore, it is recommended to apply ESP first for confidentiality, and then AH for integrity on the encapsulated payload.**

In summary, while both ESP and AH serve important roles in IPsec, performing ESP before AH is recommended when confidentiality and integrity protection for the payload are the primary goals. This approach aligns with common security practices and ensures that the data is first encrypted before applying additional integrity checks.

5. **What are the different types of port forwarding supported by SSH? Give an example for each type. (15 points)**

Ans:

Local Port Forwarding:

- **Local port forwarding allows you to forward a local port on your client machine to a specified destination host and port on the server side.**

- **Example:**

  ```
  ssh -L 8080:localhost:80 user@remote-server
  ```

  This forwards the local port 8080 on your machine to port 80 on the remote server. You can then access a web service on the remote server by navigating to http://localhost:8080 in your local browser.

Remote Port Forwarding:

- **Remote port forwarding enables you to forward a port from the remote server to your local machine.**
- **Example:**

  ```
  ssh -R 2222:localhost:22 user@remote-server
  ```

  This forwards the remote port 2222 on the server to port 22 on your local machine. Now, you can SSH into your local machine from the remote server using ssh -p 2222 user@localhost.

Dynamic Port Forwarding (SOCKS Proxy):

- **Dynamic port forwarding sets up a local SOCKS proxy that allows your SSH client to act as a proxy server, dynamically forwarding traffic to different destinations.**
- **Example:**

  ```
  ssh -D 1080 user@remote-server
  ```

  This establishes a dynamic port forwarding on your local machine, using port 1080. You can then configure your browser or other applications to use the SOCKS proxy at localhost:1080. All the traffic will be securely forwarded through the SSH tunnel to the remote server.

These port forwarding techniques provide flexibility and security for various networking scenarios, allowing users to securely access services on remote servers or redirect traffic from the server to their local machine.