CN [Day - 4]

UID: 24MCI10204

Name: Rahul Saxena

Branch: 24MCA – AI & ML

Question 1: A user on a web browser (Client) makes an HTTPS request to access a webpage from a remote server. The DNS server is queried to resolve the domain name. Once the IP is resolved, a secure connection is established between the client and server using RSA. The following sequence occurs:

1. Client sends a DNS query to resolve www.securedata.com.
2. DNS resolves the domain to 192.0.2.10.
3. The browser initiates an HTTPS GET request using HTTP over TLS.
4. During the handshake, the server sends its public key (e, n) = (7, 187).
5. The client generates a session key k = 45 and encrypts it using the RSA public key.
6. The encrypted session key is sent to the server.
7. Communication proceeds using this session key under symmetric encryption.

Questions
(a) Identify and explain the role of each Application Layer protocol used in this scenario.
(b)Calculate the encrypted session key using the RSA public key (7, 187).
(c) Explain why DNS is critical to the Application Layer and how it supports scalability.
(d) Describe the purpose of the HTTPS handshake and how it ensures secure communication.
(e) Suppose an attacker intercepts the DNS response and alters the IP address. What type of attack is this, and how can it be mitigated?

Answer:

A) Identify and explain the role of each Application Layer protocol used in this scenario

1. DNS (Domain Name System):
    - Resolves the human-readable domain (www.securedata.com) to an IP address (192.0.2.10).
    - Operates over UDP (port 53) for queries and is essential for hostname resolution.
2. HTTP (HyperText Transfer Protocol):
    - Application-layer protocol used by the browser to request web content.
    - In this case, HTTP is wrapped inside TLS, making it HTTPS.
3. HTTPS (HTTP Secure):
    - HTTP + TLS/SSL ensures data confidentiality and integrity.
    - Establishes a secure channel using public key cryptography (RSA) and symmetric encryption for the session.

B) Calculate the encrypted session key using the RSA public key (e, n) = (7, 187)

RSA Encryption formula:

cipher = k^e \mod n

Given:

- Session key k=45

- Public key $(e,n)=(7,187)$
- $cipher=45^7 mod\ 187$

Let's compute step by step:

- $45^2=2025$
- $45^3=45×2025=91125$
- $45^4=45×91125 = 4100625$
- Instead, we use modular exponentiation for efficiency:

Fast Modular Exponentiation:

We need to compute:

45^7 mod 187

Using:

$45^7 = ((45^2)^3) X 45$

Step-by-step:

- $45^1 mod\ 187=45$
- $45^2 = 2025 \Rightarrow 2025 \bmod 187 = 2025 - (187×10) = 2025 – 1870 = 155$
- $45^4 = (452)2=1552=24025 \Rightarrow 24025 mod\ 187=24025-(187×128)=24025-23936=89$
- $45^7=45^4×45^2×45$ mod $187=89×155×45$ mod 187

Now calculate:

- $89×155=13795$
- $13795×45=620775$
- 620775 mod $187=620775-(3320×187=620840)=-65+187=122$

Encrypted session key = 122

C) Why is DNS critical to the Application Layer, and how does it support scalability?

Role of DNS:

- Translates **human-readable domain names** into **IP addresses** required by the transport and network layers.
- Acts as the **"phonebook" of the internet.**

Scalability:

- DNS is **hierarchical** (root → TLD → authoritative DNS).
- Uses **caching**, **replication**, and **distributed servers** to handle millions of queries efficiently.
- Enables the **internet to grow** without being bottlenecked by a central server.

D) Purpose of the HTTPS handshake and how it ensures secure communication

Purpose:

- The **TLS/SSL handshake** ensures that the client and server:
  - Authenticate each other (server always, client optionally)
  - **Establish a shared secret key** for symmetric encryption
  - Prevent **eavesdropping**, **tampering**, and **forgery**

Process Overview:

1. Server sends its **digital certificate** containing its **RSA public key**.
2. Client verifies the certificate, then **encrypts a session key** using the public key.
3. Only the server can decrypt it using its **private key**.
4. All further communication is **encrypted symmetrically** using this shared key.

Ensures **confidentiality**, **integrity**, and **authentication**.

E) If an attacker intercepts the DNS response and alters the IP address...

This is known as a **DNS Spoofing (DNS Cache Poisoning)** attack.

- The attacker sends a **fake IP address** for the domain name.
- The user unknowingly connects to a **malicious server** instead of the legitimate one.

Type of attack:

- **Man-in-the-Middle (MITM)**
- Specifically: **DNS Spoofing**

Mitigation Strategies:

1. **DNSSEC (Domain Name System Security Extensions):**
   - Adds **digital signatures** to DNS responses to verify authenticity.
2. **HTTPS with valid certificates:**
   - Even if DNS is spoofed, the browser warns if the **certificate doesn't match the domain**.
3. **Encrypted DNS (DoH/DoT):**
   - DNS over HTTPS or TLS to prevent tampering by intermediaries.
4. **Browser-based protection:**
   - Browsers validate certificates using **CA chains** and show warnings.

Question 2: Problem Statement

Consider the following scenario:

A university student is working remotely and performs these tasks:

1. Logs into the university server remotely using TELNET to submit an assignment.
2. Opens a browser and visits www.university-portal.edu using HTTP.
3. The DNS is queried to resolve the domain name to IP address 203.0.113.8.
4. The server responds with a webpage that includes a form submission via POST.
5. The server uses RSA encryption for login authentication. The public key is (e = 5, n = 221).
6. The user sends an email via SMTP, and then downloads emails via POP3.

Questions

(a) Describe the client-server interaction in each of the above tasks.

(b) Explain how DNS functions in this scenario and what would happen if DNS fails.

(c) What port numbers are used by TELNET, HTTP, SMTP, and POP3?

(d) The password "42" is encrypted using RSA public key (5, 221). What is the ciphertext?

(e) Explain the security limitations of using TELNET and how SSH addresses them.

(f) Compare HTTP and E-mail protocols (SMTP, POP3) in terms of their communication pattern.

Answer:

A) Client-Server Interaction in Each Task:

1. **TELNET Login (Remote Access):**
   - The student (client) initiates a **TELNET** session to log into the university server.
   - TELNET allows terminal-based remote login; the server responds with a login prompt and executes commands sent by the client.
2. **Accessing Website via HTTP:**
   - The browser (client) sends an **HTTP GET** request to fetch the webpage.
   - The web server responds with the requested HTML page.
3. **DNS Query:**
   - The client's machine sends a **DNS query** to a DNS server to resolve www.university-portal.edu to IP 203.0.113.8.
   - This enables the browser to connect to the correct server.
4. **Form Submission (HTTP POST):**
   - The student fills a form (e.g., assignment upload).
   - The browser sends an **HTTP POST** request to the server with form data.
5. **RSA Login Authentication:**
   - The server sends its **public RSA key (e = 5, n = 221)**.

- The client encrypts the password and sends it securely. Only the server (holding the private key) can decrypt it.

6. **Sending and Receiving Email:**
   - **SMTP** is used to **send** email from client to mail server.
   - **POP3** is used to **download** or **retrieve** emails from the mail server to the client.

B) How DNS Functions and What if It Fails:

How DNS works:

- Translates **domain names** into **IP addresses**.
- Acts like the **internet's phonebook**.
- The client sends a query; DNS server responds with the IP.

If DNS fails:

- The client **cannot reach the server** using the domain name.
- Browser shows "Server not found" or "DNS lookup failed."
- The user would have to manually enter the **IP address** (if known), which is not practical.

C) Port Numbers Used:

Protocol	Port Number	Description
TELNET	23	Unencrypted terminal access
HTTP	80	Web traffic (unencrypted)
SMTP	25	Sending emails
POP3	110	Retrieving emails

Encrypt password "42" using RSA (e = 5, n = 221)

D) RSA Encryption Formula:

cipher=m^e mod n

Given:

- Message/password m = 42
- Public key (e = 5, n = 221)

cipher=42^5mod 221

Step-by-step (modular exponentiation):

RSA Encryption of Plaintext = 42 using Public Key (e = 5, n = 221)

We want to calculate:

Ciphertext = (42^5) mod 221

____

Step 1: Calculate 42^2 mod 221

$42^2 = 1764$

1764 mod 221 = 1764 − (221 × 7) = 1764 − 1547 = 217

____

Step 2: Calculate 42^4 mod 221

$42^4 = (42^2)^2 = 217^2 = 47089$

47089 mod 221 = 47089 − (221 × 213) = 47089 − 47073 = 16

____

Step 3: Calculate 42^5 mod 221

$42^5 = 42 \times 42^4 = 42 \times 16 = 672$

672 mod 221 = 672 − (221 × 3) = 672 − 663 = 9

____
Final Answer:
Encrypted Ciphertext = 9

(e) Security Limitations of TELNET & How SSH Solves Them

TELNET Limitations:

- Transmits **data in plain text** — including usernames and passwords.
- Vulnerable to **eavesdropping**, **packet sniffing**, and **MITM attacks**.
- No encryption, no authentication of remote server identity.

SSH (Secure Shell) Advantages:

- **Encrypts all communication**, including credentials.
- Uses **public-key cryptography** to verify the server.
- Provides **confidentiality, integrity, and authentication**.

SSH is the **secure replacement** for TELNET.

F) Compare HTTP and Email Protocols (SMTP, POP3)

Feature	HTTP	Email (SMTP, POP3)
Direction	**Client → Server**	SMTP: **Client → Server**POP3: **Server → Client**
Type	**Pull-based** (GET, POST)	SMTP: **Push**, POP3: **Pull**
Use Case	Browsing web pages	Sending/receiving emails
Connection	Stateless	Session-based (especially POP3)
Port	80	SMTP: 25, POP3: 110
Protocol Pattern	Request/Response	SMTP: Send → Queue → DeliverPOP3: Fetch