HTTPS/TLS

HTTPS & TLS Study Guide

What is HTTPS?

HTTPS stands for:

HyperText Transfer Protocol Secure

It is:

encrypted HTTP communication

Used to securely transmit:

- API traffic
- passwords
- tokens
- customer data
- banking information

HTTP vs HTTPS

HTTP	HTTPS
Not encrypted	Encrypted
Insecure	Secure
Data visible in transit	Data protected
Uses port 80	Uses port 443

Why HTTPS Matters

Without HTTPS:

- attackers could intercept traffic
- tokens/passwords could be stolen
- API data could be exposed

VERY important for:

- banking
- cloud platforms
- APIs
- OAuth authentication

What is TLS?

TLS stands for:

Transport Layer Security

TLS is:

the encryption protocol behind HTTPS

Meaning:

```
HTTPS = HTTP + TLS encryption
```

Simple Explanation

HTTP

Data travels in plain text.

HTTPS/TLS

Data is encrypted before transmission.

Example:

```
Customer → Encrypted Traffic → API Server
```

What TLS Protects

TLS provides:

Security Feature	Purpose

Encryption	Protects data confidentiality
Authentication	Verifies server identity
Integrity	Prevents data tampering

Encryption

Encryption converts readable data into:

unreadable encrypted data

Without decryption key: data cannot be understood.

Example

Without TLS:

```
password=MyPassword123
```

could be intercepted.

With TLS:

```
X93kL0sdP2mQ8...
```

encrypted and unreadable.

TLS Handshake (High-Level)

When browser/app connects securely:

Client connects

↓

Server presents certificate

↓

TLS session established

↓

Encrypted communication begins

SSL vs TLS

Older term:

SSL (Secure Sockets Layer)

Modern standard:

TLS

People still casually say:

- SSL certificate
- SSL encryption

But technically: TLS replaced SSL.

Certificates

HTTPS/TLS relies on:

digital certificates

Certificates verify:

- server identity
- trusted domain
- encryption validity

Example

When you open:

```
https://www.glia.com
```

browser checks:

- valid certificate?
- trusted authority?
- secure connection?

Common TLS Components

Component	Purpose
Certificate	Verifies server identity
Public Key	Encrypts data
Private Key	Decrypts data
Certificate Authority (CA)	Trusted issuer
TLS Handshake	Establishes secure session

Why TLS Is Critical For APIs

APIs often transmit:

- bearer tokens
- OAuth tokens

- customer data
- banking information

Without TLS: tokens could be stolen.

OAuth + HTTPS Relationship

OAuth tokens should ONLY travel over:

HTTPS/TLS encrypted connections

Example:

```
https://api.company.com/customers
Authorization: Bearer token
```

Common TLS/HTTPS Troubleshooting

Problem 1 — Certificate Expired

Symptoms:

- browser warning

- API connection failure
- TLS handshake errors

Troubleshooting:

- validate certificate expiration
- renew certificate

Problem 2 — Certificate Not Trusted

Symptoms:

```
Certificate not trusted
```

Causes:

- self-signed certificate
- missing CA chain
- invalid certificate

Problem 3 — TLS Version Mismatch

Example:

- client uses TLS 1.0
- server requires TLS 1.2+

Result: secure connection fails.

Problem 4 — Hostname Mismatch

Certificate issued for:

```
api.company.com
```

but request sent to:

```
test.company.com
```

Result: TLS validation failure.

Problem 5 — Firewall / Proxy Interference

Symptoms:

- HTTPS timeout
- TLS negotiation failure

Check:

- firewall
- proxy
- port 443 access

Common HTTPS/TLS Troubleshooting Flow

Step 1 — Validate URL

Verify:

```
https://
```

not:

```
http://
```

Step 2 — Validate Certificate

Check:

- expiration
- trusted CA
- hostname match

Step 3 — Validate TLS Version

Modern systems typically require:

- TLS 1.2
- TLS 1.3

Step 4 — Validate Port Connectivity

HTTPS typically uses:

port 443

Check:

- firewall
- load balancer
- proxy access

Step 5 — Review Logs

Check:

- TLS handshake errors
- certificate validation failures
- proxy logs

Common Interview Questions

“What is HTTPS?”

Good Answer:

> “HTTPS is secure HTTP communication that uses TLS encryption to protect data transmitted between systems.”

“What is TLS?”

Good Answer:

> “TLS is the encryption protocol that secures HTTPS communications by encrypting traffic, validating server identity, and protecting data integrity.”

“Difference between HTTP and HTTPS?”

HTTP	HTTPS
Unencrypted	Encrypted
Insecure	Secure
Port 80	Port 443

“Why is TLS important for APIs?”

Good Answer:

>

> “TLS protects sensitive API traffic such as OAuth tokens, credentials, and customer data by encrypting communication between systems.”

“What causes TLS failures?”

Common causes:

- expired certificates
- invalid certificates
- TLS version mismatch
- firewall/proxy issues
- hostname mismatch

Important Security Concepts

Never send:

- passwords
- bearer tokens
- OAuth credentials

over:

plain HTTP

Always use:

HTTPS/TLS encrypted communication

Easy Memory Trick

HTTPS = Secure Website/API

TLS = Encryption

Technology Behind HTTPS

Example:

```
HTTPS uses TLS to encrypt traffic
```

Important Terms To Know

Term	Meaning
HTTP	Unencrypted web traffic
HTTPS	Secure encrypted HTTP
TLS	Encryption protocol
SSL	Older predecessor to TLS
Certificate	Verifies server identity
CA	Certificate Authority

Term	Meaning
Port 443	HTTPS secure port
Encryption	Protecting data confidentiality
TLS Handshake	Secure session negotiation

Revision #1
Created 21 May 2026 00:46:40 by Cesar Gzz
Updated 21 May 2026 00:46:46 by Cesar Gzz