Encrypt In Transit

Student: Mohammed JBILOU
ID: 7702249
Course: Cybersecurity
Institution: Keimyung University
Github reposiroty:
https://github.com/Molaryy/KMU_cybersecurity_assignment_2025

Table of Contents

1 - Objective

The objective of this lab is to secure communication between clients and the server using modern encryption protocols. This includes implementing HTTPS with TLS certificates, enabling HTTP Strict Transport Security (HSTS), and validating that encryption in transit is functioning correctly.

2 - Implementation Overviews

This lab uses Nginx running in a Docker container to provide HTTPS services secured with a self-signed TLS certificate. Traffic arriving over HTTP is redirected to HTTPS, and the server enforces strict transport rules through HSTS.

Technologies Used

- Nginx 1.29.3 (Alpine)
- Docker
- TLSv1.2 and TLSv1.3
- Self-signed X.509 certificate

Architecture Overview

- HTTP port 8080 redirects to HTTPS
- HTTPS port 8443 handles encrypted traffic
- Certificates stored in `/etc/nginx/ssl/`
- Web root at `/usr/share/nginx/html/`

3 - TLS Certificate Implementation

Certificate Generation

A self-signed certificate was generated using OpenSSL:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout ssl/key.pem \
  -out ssl/cert.pem \
  -subj "/C=US/ST=State/L=City/O=Organization/OU=IT/CN=localhost"
```

The certificate uses a 2048-bit RSA key and is valid for one year.

Certificate Verification

```
openssl s_client -connect localhost:8443 -showcerts </dev/null 2>/dev/null |
  openssl x509 -noout -dates -subject -issuer

notBefore=Nov 17 11:29:14 2025 GMT
notAfter=Nov 17 11:29:14 2026 GMT
subject=C=US, ST=State, L=City, O=Organization, OU=IT, CN=localhost
issuer=C=US, ST=State, L=City, O=Organization, OU=IT, CN=localhost
```

Output confirmed:

- One-year validity
- Matching issuer and subject (self-signed)
- Expected CN and organizational attributes

Certificate Verification

The server enforces modern TLS versions and secure cipher suites:

```
nginx.conf

ssl_protocols TLSv1.2 TLSv1.3;
ssl_ciphers 'ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384';
```

4 - HSTS Implementation

HSTS Header

HSTS is configured with a one-year policy, applied to all subdomains, and marked for preload:

```
nginx.conf

add_header Strict-Transport-Security "max-age=31536000; includeSubDomains; preload" always;
```

Verification

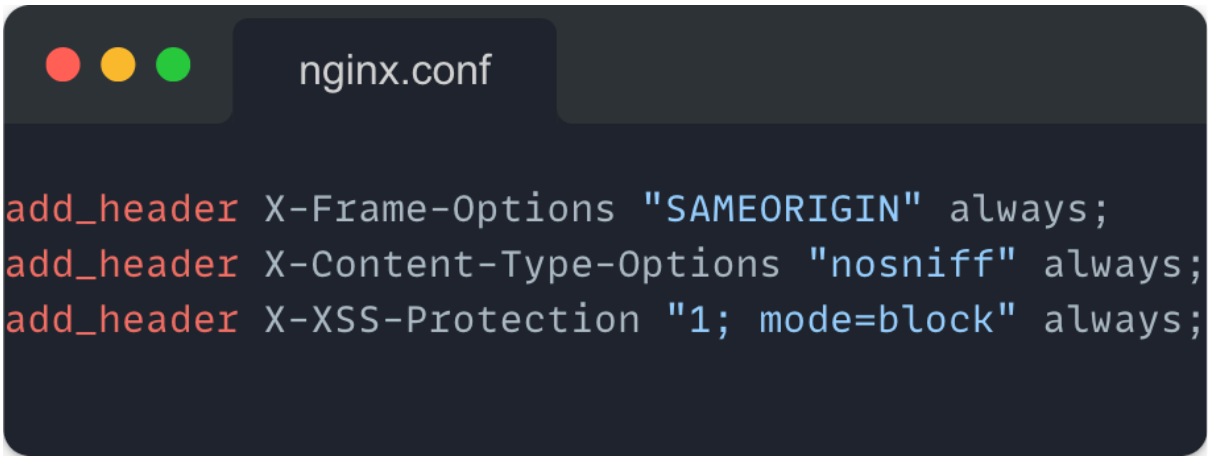
Using curl:

```
curl -I -k https://localhost:8443 | grep -i strict-transport-security


  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0  1077    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0

# Result right here
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
```

Output confirmed that the header is correctly returned in HTTPS responses.

5 - Additional Security Headers

To strengthen the server's defense-in-depth posture, the following headers were added:



These help mitigate clickjacking, MIME-sniffing, and reflective XSS attacks.

6 - Verification and Testing

HTTPS Response Validation

Using curl:

```
curl -I -k https://localhost:8443

# Response
HTTP/1.1 200 OK
Server: nginx/1.29.3
Date: Mon, 17 Nov 2025 12:38:28 GMT
Content-Type: text/html
Content-Length: 1077
Last-Modified: Sun, 16 Nov 2025 11:21:27 GMT
Connection: keep-alive
ETag: "6919b3b7-435"
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Accept-Ranges: bytes
```

The response included TLS-related headers and security headers, confirming correct HTTPS behavior.

TLS Protocol Verification

TLSv1.3 negotiation was confirmed:

```
curl -vI -k https://localhost:8443 2>&1 | grep -i "ssl\|tls"

# Reponse
* (304) (OUT), TLS handshake, Client hello (1):
* (304) (IN), TLS handshake, Server hello (2):
* (304) (IN), TLS handshake, Unknown (8):
* (304) (IN), TLS handshake, Certificate (11):
* (304) (IN), TLS handshake, CERT verify (15):
* (304) (IN), TLS handshake, Finished (20):
* (304) (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / AEAD-CHACHA20-POLY1305-SHA256 / [blank] / UNDEF
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
```

The server successfully negotiated TLSv1.3 with strong authenticated cipher suites.

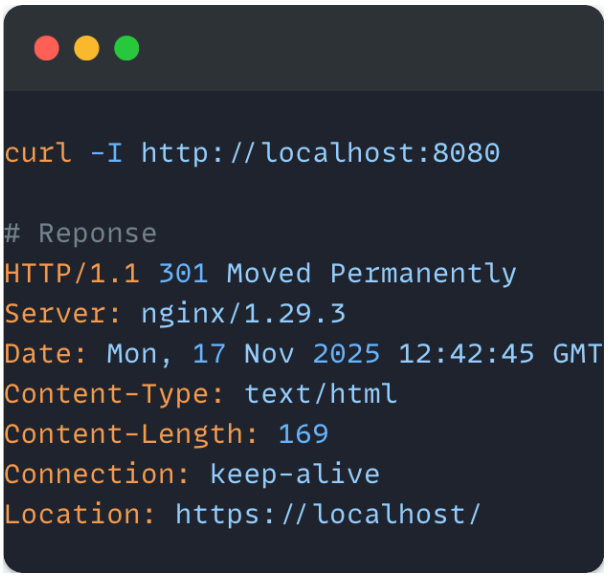
HTTP to HTTPS Redirect Test

Based on this configuration where I redirect the default HTTP to https

nginx.conf

```
server {
    listen 80;
    server_name localhost;

    return 301 https://$host$request_uri;
}
```

HTTP requests are redirected using a 301, thanks to the configuration above
⬆



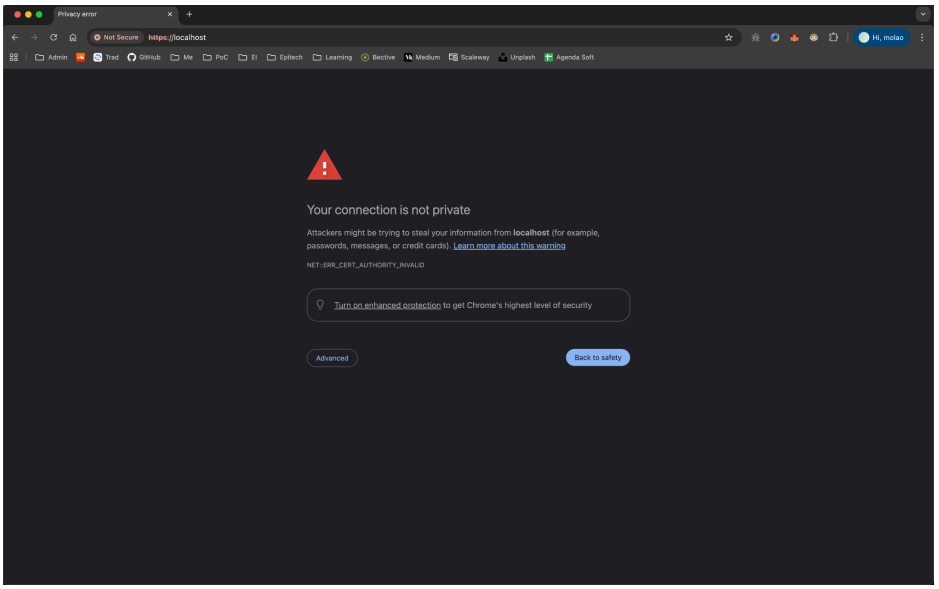
Output shows a permanent redirect to HTTPS, ensuring all traffic is encrypted.

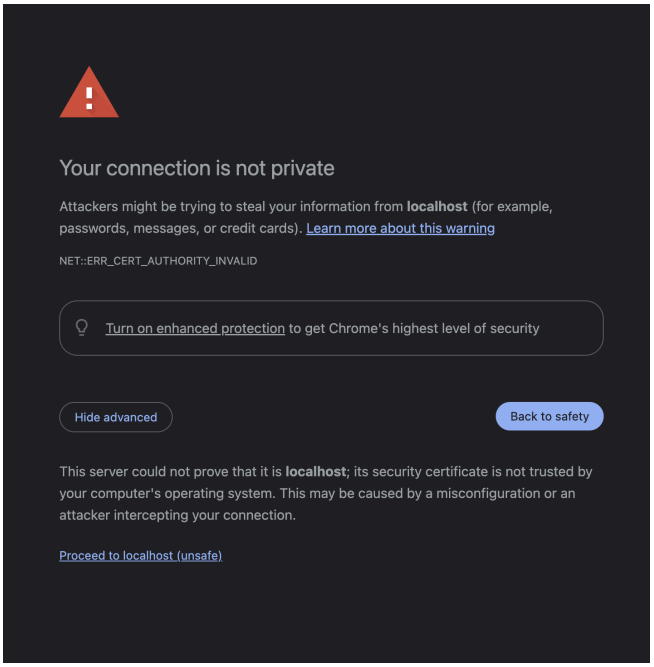
Browser Verification

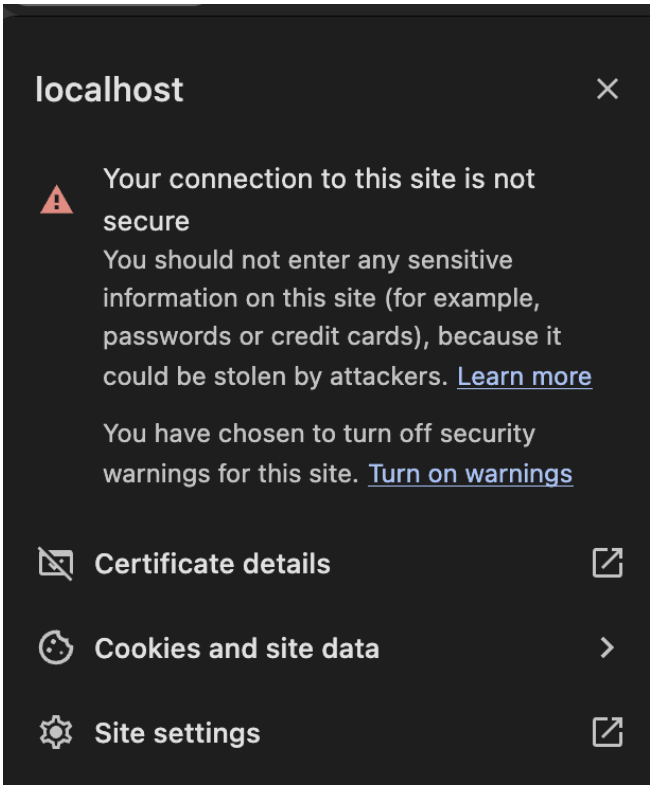
In a browser:

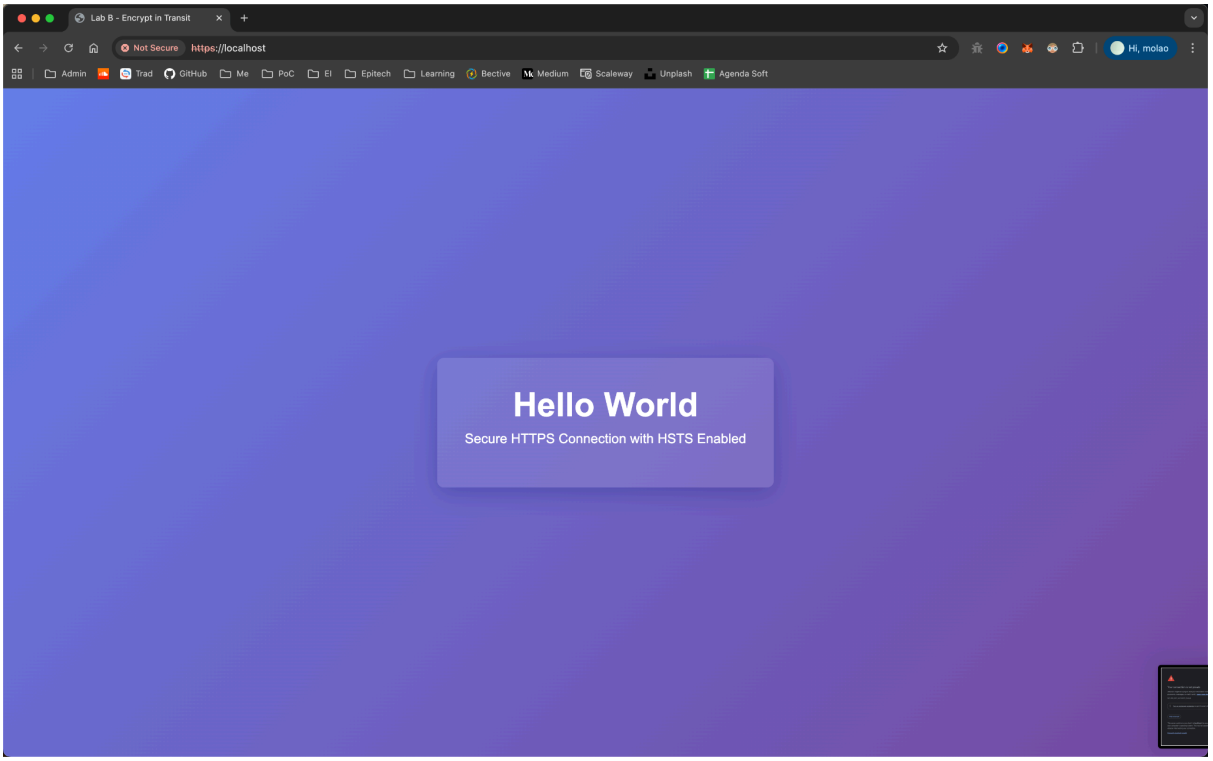
- HTTPS loads with a lock icon (with expected warning for self-signed certificates)
- Certificate details show correct CN, issuer, and validity
- Developer Tools confirm the presence of the HSTS header
- Attempting to visit HTTP results in automatic redirection to HTTPS

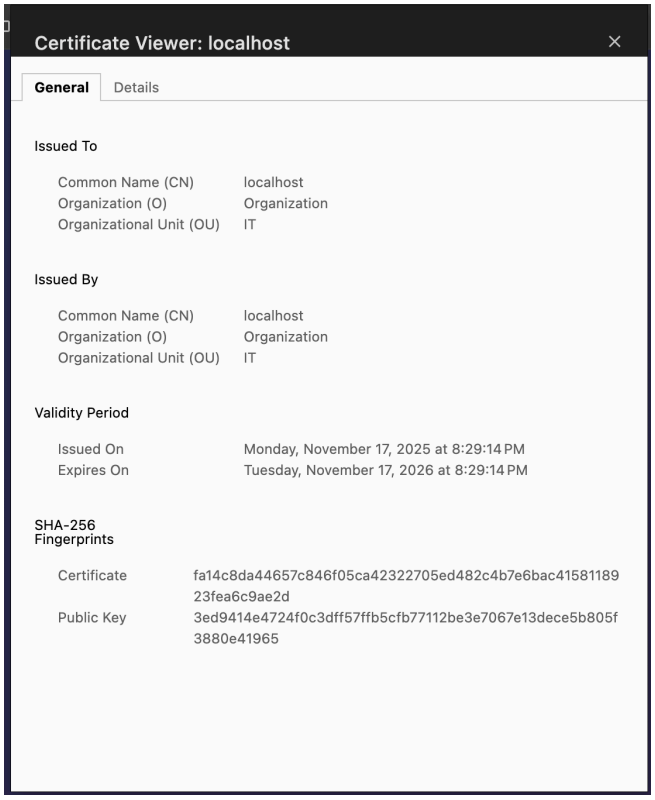
Here are the steps:

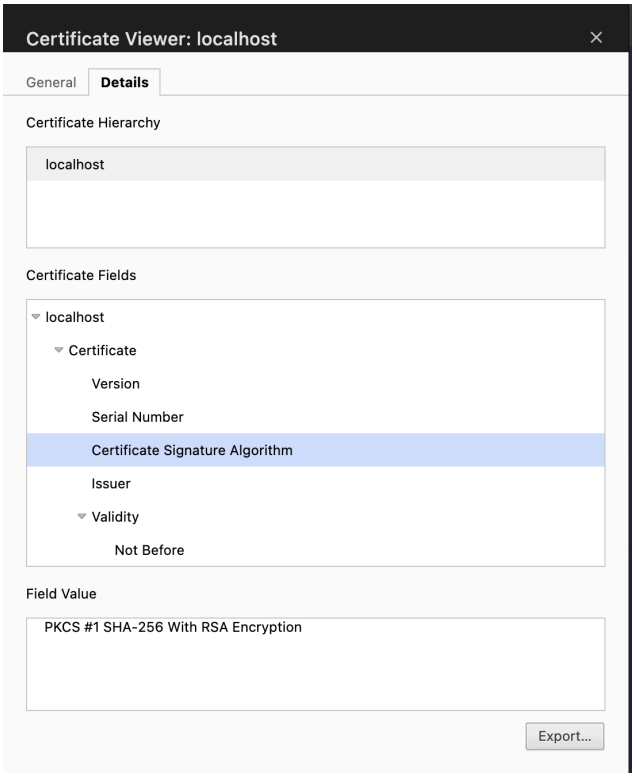







Lab B - Encrypt in Transit
Not Secure
https://localhost
Hi, molao
Admin
Trad
GitHub
Me
PoC
EI
Epitech
Learning
Bective
Medium
Scaleway
Unplash
Agenda Soft
Hello World
Secure HTTPS Connection with HSTS Enabled


Certificate Viewer: localhost
General
Details
Issued To
Common Name (CN) localhost
Organization (O) Organization
Organizational Unit (OU) IT
Issued By
Common Name (CN) localhost
Organization (O) Organization
Organizational Unit (OU) IT
Validity Period
Issued On Monday, November 17, 2025 at 8:29:14 PM
Expires On Tuesday, November 17, 2026 at 8:29:14 PM
SHA-256 Fingerprints
Certificate fa14c8da44657c846f05ca42322705ed482c4b7e6bac4158118923fea6c9ae2d
Public Key 3ed9414e4724f0c3dff57ffb5cfb77112be3e7067e13dece5b805f3880e41965


Certificate Viewer: localhost
General
Details
Certificate Hierarchy
localhost
Certificate Fields
localhost
Certificate
Version
Serial Number
Certificate Signature Algorithm
Issuer
Validity
Not Before
Field Value
PKCS #1 SHA-256 With RSA Encryption
Export...

7 - Security Best Practices Implemented

Practice	Implementation	Benefit
TLS 1.3	Enabled	Modern, secure protocol
Strong cipher suites	ECDHE with AES-GCM and ChaCha20	Forward secrecy and authenticated encryption
HSTS	with preload and includeSubDomains	Prevents downgrade attacks
HTTP redirect	301 redirect to HTTPS	Ensures encrypted connections
Security headers	Clickjacking, MIME-sniffing, XSS protection	Defense in depth
2048-bit RSA key	Certificate strength	Meets industry security standard

8 - Conclusion

This lab demonstrates proper implementation of encrypted communication using TLS, secure Nginx configuration, HSTS enforcement, and verification using both browser and command-line tools. The configuration successfully enforces HTTPS, strengthens transport-layer security, and meets all assignment requirements.