**Assignment No : 6**

**Title** : Implement web security with Open SSL tool kit.

**Step 1: Install OpenSSL on Windows**

1. **Download OpenSSL**
   * Visit OpenSSL Windows Binaries.
   * Download the latest **OpenSSL Light** version for your system (Win32 or Win64).
   * Install OpenSSL and make sure to check the option **"Add OpenSSL to system PATH"**.
   * If not added to PATH, manually add C:\Program Files\OpenSSL-Win64\bin to the system's PATH.
2. **Verify Installation**  
   Open **Command Prompt** and type:
   * + openssl version

### ****Step 2: Generate SSL/TLS Certificates using OpenSSL****

#### **1. Generate a Private Key (2048-bit)**

* + - openssl genrsa -out private.key 2048
    - openssl req -new -key private.key -out request.csr
    - openssl x509 -req -days 365 -in request.csr -signkey private.key -out certificate.crt

**Step 3: Convert Certificates to Different Formats**

* Convert **PEM** to **PKCS12 (PFX)**
  + - openssl pkcs12 -export -out certificate.pfx -inkey private.key -in certificate.crt
* Convert **PEM** to **DER** (Binary format)
  + - openssl x509 -outform der -in certificate.crt -out certificate.der

Convert **PEM** to **Base64**

* + - openssl base64 -in certificate.crt -out certificate\_base64.crt

**Step 4: Verify and Check SSL Certificate Details**

* Check the contents of a certificate:
  + - openssl x509 -in certificate.crt -text -noout

Verify the CSR file:

* + - openssl req -in request.csr -text –noout

Verify the private key:

* + - openssl rsa -in private.key –check

Verify the certificate against the private key:

* + - openssl x509 -noout -modulus -in certificate.crt | openssl md5
    - openssl rsa -noout -modulus -in private.key | openssl md5

**Step 5: Test SSL/TLS Connection**

* Check SSL connection for a website:
  + - openssl s\_client -connect www.google.com:443

Check supported SSL/TLS versions:

* + - openssl s\_client -connect www.google.com:443 -tls1\_2
    - openssl s\_client -connect www.google.com:443 -tls1\_3

**tep 6: Create an HTTPS Server with OpenSSL**

1. Generate a self-signed certificate.
2. Run a simple HTTPS server (using Python as an example):
   * + python -m http.server 443 --bind 127.0.0.1 --certfile certificate.crt --keyfile private.key
3. Open your browser and visit:  
   https://127.0.0.1

### ****Conclusion****

These commands allow you to implement and test web security using OpenSSL on Windows. You can use them for creating and managing SSL/TLS certificates for websites, applications, or internal secure communications.

**Other Simple method for windows**

**Step 1: Install OpenSSL on Windows**

1. **Download OpenSSL**
   * Visit OpenSSL Windows Binaries.
   * Download the latest **OpenSSL Light** version for your system (Win32 or Win64).
   * Install OpenSSL and make sure to check the option **"Add OpenSSL to system PATH"**.
   * If not added to PATH, manually add C:\Program Files\OpenSSL-Win64\bin to the system's PATH.
2. **Verify Installation**  
   Open **Command Prompt** and type:
   * + openssl version

**Step 2: Open Command Prompt and Navigate to OpenSSL Directory**

* cd C:\Program Files\OpenSSL-Win64\bin

### Step 3 : ****Generate a Private Key****

- openssl genpkey -algorithm RSA -out server.key

{ if error occurred in the cmd then try this

 **Close Command Prompt** if it's already open.

 **Open Command Prompt as Administrator**:

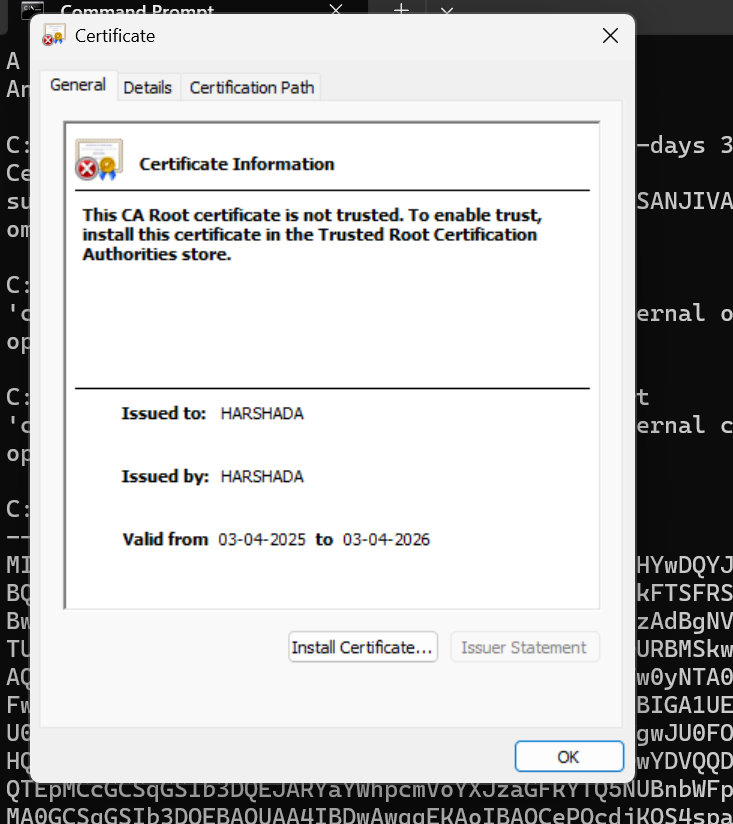
* Click **Start** → **Search "cmd"**
* Right-click **Command Prompt** → **Run as Administrator**

 **Try the command again**:

* openssl genpkey -algorithm RSA -out server.key }

**Step 4 : After that go the cmd**

- mkdir https-server

* cd https-server
* openssl genpkey -algorithm RSA -out server.key
* openssl req -new -key server.key -out server.csr
* openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
* type server.crt
* ****