Laboratory_12: Self-Signed Certificates and MiTM Detection

This laboratory covers the creation and verification of self-signed certificates and explores how they can be exploited for Man-in-the-Middle (MiTM) attacks.

Installation

- sudo apt update
- sudo apt-get install openssl –y
- openssl version

Creating Self-Signed Certificates

Part 1: Generate a Private Key

1. Create a directory for our certificates:

   ```
   mkdir -p ~/ssl-lab
   cd ~/ssl-lab
   ```

2. Generate a 2048-bit RSA private key:

   ```
   openssl genrsa -out server.key 2048
   ```

3. Verify the key was created:

   ```
   ls -la server.key
   ```

Part 2: Create a Certificate Signing Request (CSR)

1. Generate a CSR with our private key:
   openssl req -new -key server.key -out server.csr

2. Fill in the certificate information when prompted:

a. Country Name: SP
b. State: Navarre
c. Locality: Pamplona
d. Organization: Inkor
e. Organizational Unit: Security
f. Common Name: testserver.local
g. Email: admin@testserver.local

3. Review the Certificate Signing Request (CSR):

   openssl req -text -noout -in server.csr

Part 3: Generate the Self-Signed Certificate

1. Create a self-signed certificate valid for 365 days:

   openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt

2. Verify the certificate:

   openssl x509 -text -noout -in server.crt

Part 4: Create an Alternate Certificate (MiTM Scenario)

1. Generate a new private key for the attacker:

   openssl genrsa -out attacker.key 2048

2. Create a CSR with the SAME Common Name as the legitimate server:

   openssl req -new -key attacker.key -out attacker.csr

   a. Use the same CN: testserver.local

3. Generate attacker's self-signed certificate:
   openssl x509 -req -days 365 -in attacker.csr -signkey attacker.key -out attacker.crt

Part 5: Certificate Verification

1. Check the fingerprint of the legitimate certificate:

   ```
   openssl x509 -fingerprint -noout -in server.crt
   ```

2. Check the fingerprint of the attacker's certificate:

   ```
   openssl x509 -fingerprint -noout -in attacker.crt
   ```

Part 6: Creating a Simple HTTPS Server

1. Create a simple HTML file:

   ```
   echo "<html><body><h1>Secure Server</h1></body></html>" > index.html
   ```

2. Start a simple HTTPS server with the legitimate certificate:

   ```
   sudo openssl s_server -key server.key -cert server.crt -www -accept 443
   ```

3. In another terminal, test the connection:

   ```
   openssl s_client -connect localhost:443
   ```

4. Stop the server (Ctrl+C) and start it with the attacker's certificate:

   ```
   sudo openssl s_server -key attacker.key -cert attacker.crt -www -accept 443
   ```

5. Test the connection again and compare the certificate details:

   ```
   openssl s_client -connect localhost:443 | openssl x509 -fingerprint -noout
   ```

MiTM Vulnerability Explanation

Self-signed certificates are vulnerable to MiTM attacks because:

1. They lack verification from a trusted Certificate Authority (CA)

2. Browsers and clients cannot distinguish between legitimate and malicious self-signed certificates
3. Users often ignore certificate warnings
4. Attackers can create certificates with identical subject information

To mitigate these risks:

- Use certificates from trusted CAs
- Implement certificate pinning
- Educate users about certificate warnings
- Use proper certificate validation in applications