Lab-5: Securing Apache Web Server

Solution (Commands & Configs)

Md. Rakibul Kabir
Reg: 2020831051

November 8, 2025

Contents

1 Overview

This document provides step-by-step commands and configuration snippets to:

- Create your own Certificate Authority (CA).
- Generate server key and CSR for `example.com` and `webserverlab.com`.
- Sign CSRs with your CA to produce certificates.
- Launch a temporary OpenSSL TLS server for testing.
- Configure Apache to serve HTTPS using the generated certificates.
- Provide suggested checkpoints to show the instructor.

2 Prerequisites

- Linux environment (Ubuntu/Debian recommended) with `openssl` and `apache2` installed.
- sudo privileges.

3 Step 0: Prepare workspace

```
# create workspace
mkdir -p ~/lab5_ca
cd ~/lab5_ca
```

3.1 Create openssl.cnf (starter)

Copy your system template (if available) or use a minimal config. Example minimal configuration (save as `openssl.cnf`):

```
[ ca ]
default_ca = CA_default

[ CA_default ]
dir               = ./demoCA
certs             = $dir/certs
new_certs_dir     = $dir/newcerts
database          = $dir/index.txt
serial            = $dir/serial
private_key       = $dir/private/ca.key
certificate       = $dir/cacert.pem
default_md        = sha256
policy            = policy_any

[ policy_any ]
commonName = supplied
```

Create the directories and files referenced:

```
mkdir -p demoCA/{certs,crl,newcerts,private}
touch demoCA/index.txt
echo 1000 > demoCA/serial
```

4 Task-1: Become a Certificate Authority (root CA)

```
# generate CA private key and self-signed certificate
openssl req -new -x509 -days 3650 -extensions v3_ca \
    -keyout demoCA/private/ca.key -out demoCA/cacert.pem \
    -config openssl.cnf
```

You will be prompted for a passphrase (remember it) and certificate details (Country, Common Name etc.). The CA certificate is `demoCA/cacert.pem`.

5 Create certificate for **`example.com`**

5.1 Step 1: Generate server key

```
# create server key protected by passphrase
openssl genrsa -des3 -out example.com.key 2048
```

5.2 Step 2: Create CSR (use example.com as Common Name)

```
openssl req -new -key example.com.key -out example.com.csr -config openssl.
   cnf
```

5.3 Step 3: Sign CSR with CA

```
# sign CSR to create server certificate
openssl ca -in example.com.csr -out example.com.crt -cert demoCA/cacert.pem
     -keyfile demoCA/private/ca.key -config openssl.cnf
```

If OpenSSL asks to confirm or mentions policy mismatch, ensure CN in CSR is `example.com` and matches CA policy.

6 Launch OpenSSL test server (quick verification)

Combine key and certificate into a PEM file and launch s_server:

```
cp example.com.key example.com.pem
cat example.com.crt >> example.com.pem

# run temporary TLS server on port 4433
openssl s_server -cert example.com.pem -www -accept 4433
```

Browse to `https://localhost:4433/` or `https://example.com:4433/` (point `example.com` to 127.0.0.1 in /etc/hosts). Browser will complain because CA isn't trusted.

6.1 Trusting the CA in Firefox (for testing)

- Preferences → Privacy & Security → View Certificates → Authorities → Import

- Import `demoCA/cacert.pem` and select “Trust this CA to identify websites”

7 Repeat for **webserverlab.com**

Repeat the key generation, CSR creation and signing steps replacing common name with `webserverlab.com`.

8 Deploy HTTPS into Apache (Task-3)

8.1 1. Enable ssl module

```
sudo a2enmod ssl
```

8.2 2. Create or edit virtual host

Example: `/etc/apache2/sites-available/example.com.conf`

```
<IfModule mod_ssl.c>
<VirtualHost *:443>
    ServerAdmin admin@example.com
    ServerName example.com
    ServerAlias www.example.com
    DocumentRoot /var/www/example.com/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    SSLEngine on
    SSLCertificateFile /path/to/example.com.crt
    SSLCertificateKeyFile /path/to/example.com.key
</VirtualHost>
</IfModule>
```

8.3 3. Enable site and test config

```
sudo a2ensite example.com.conf
sudo apache2ctl configtest
sudo systemctl restart apache2
```

8.4 4. Browse to **https://example.com/**

If browser warns, import the CA certificate as above.

9 Checkpoints (what to demonstrate)

1. **Checkpoint 1 (5 marks):** Launch OpenSSL s_server as above and show `https://localhost` or `https://example.com:4433/`. Show the browser warning, then import CA cert into browser, reload and show the page loads without warning.
2. **Checkpoint 2 (5 marks):** Repeat for `webserverlab.com`.
3. **Checkpoint 3 (5 marks):** Configure Apache virtual host with SSL, enable module, restart Apache and show `https://example.com/` serving the site.
4. **Checkpoint 4 (5 marks):** Repeat Apache HTTPS setup for `webserverlab.com`.

10 Troubleshooting tips

- If `openssl ca` complains about `index.txt` or serial, ensure `demoCA/index.txt` exists and `demoCA/serial` contains a number (e.g., 1000).
- If Apache fails to start, check `/var/log/apache2/error.log` and run `apache2ctl configtest`.
- If the browser still complains after importing CA, ensure you imported the CA certificate into the correct store (Authorities/trusted roots).

11 Appendix: Useful commands summary

```
# make CA
mkdir -p demoCA/{certs,newcerts,private}
touch demoCA/index.txt
echo 1000 > demoCA/serial
openssl req -new -x509 -days 3650 -extensions v3_ca \
    -keyout demoCA/private/ca.key -out demoCA/cacert.pem \
    -config openssl.cnf

# server key + CSR + sign
openssl genrsa -des3 -out server.key 2048
openssl req -new -key server.key -out server.csr -config openssl.cnf
openssl ca -in server.csr -out server.crt -cert demoCA/cacert.pem -keyfile demoCA/private/ca.key -config openssl.cnf

# test server
cp server.key server.pem
cat server.crt >> server.pem
openssl s_server -cert server.pem -www -accept 4433

# apache
sudo a2enmod ssl
sudo a2ensite example.com.conf
sudo apache2ctl configtest
sudo systemctl restart apache2
```