# Creating Self-Signed SSL Certificates for Apache on Linux

TLS stands for Transport Layer Security

SSL stands for Secure Sockets Layer

**SSL :-** (Secure Sockets Layer) is the standard security technology for establishing an encrypted link between a web server and a browser. This link ensures that all data passed between the web server and browsers remain private and integral. SSL is an industry standard and is used by millions of websites in the protection of their online transactions with their customers. To be able to create an SSL connection a web server requires an SSL Certificate. When you choose to activate SSL on your web server you will be prompted to complete a number of questions about the identity of your website and your company. Your web server then creates two cryptographic keys - a Private Key and a Public Key.

**OpenSSl :-** OpenSSL is an open source tool for using the Secure Socket Layer (SSL) and Transport Layer Security (TLS) protocols for Web authentication.

**mod\_ssl** :- mod\_ssl is an Apache module that provides SSL v2/v3 and TLS v1 support for the Apache HTTP Server.

**What is a CSR? :-** A CSR or **Certificate Signing request** is a block of encoded text that is given to a Certificate Authority when applying for an SSL Certificate. It is usually generated on the server where the certificate will be installed and contains information that will be included in the certificate such as the organization name, common name (domain name), locality, and country. It also contains the public key that will be included in the certificate. A private key is usually created at the same time that you create the CSR

## PassPhrase :-“A passphrase is like a password except it is longer”. In the early days passwords on Unix system were limited to 8 characters, so the term passphrase for longer passwords. Longer is the password harder it is to guess. Nowadays Unix systems use MD5 hashes which have no limitation in length of the password.

**Public key** authentication is more secure than password authentication. ... Each **key** is a large number with special mathematical properties. The **private key** is kept on the computer you log in from, while the **public key** is stored on the .**ssh**/authorized\_keys file on all the computers you want to log in to.

[**crt**](https://en.wikipedia.org/wiki/X.509#Certificate_filename_extensions) and **key** files represent both parts of a certificate, key being the private key and Public key to the certificate and crt being the signed certificate.

Configure SSL in Apache :-

**HTTPD using rpm :-**

1. Stop the httpd sevice

# service httpd status

# service httpd stop

1. Install the following rpms

# yum install openssl

# yum install mod\_ssl

1. Edit the configuration file of apache /etc/httpd/conf/httpd.conf and edit/modify the following entry in it in order to include ssl.conf file in it.View the ssl.conf to review all the default SSL configurations. Here \* stands for including all the .conf files in httpd.conf file. (Uncommend)

# cd /etc/httpd/conf/

# vim httpd.conf

Include conf.d/\*.conf

1. The SSL certificate file and key file are required before we start the Apache. The server.crt and server.key files mentioned in the ssl.conf needs to be created before we move forward. Please modify the path of the certificate and key files in the ssl.conf file.

# vim /etc/httpd/conf.d/ssl.conf

SSLCertificateFile "/etc/httpd/conf.d/server.crt"

SSLCertificateKeyFile "/etc/httpd/conf.d/server.key"

1. Generate the server.key using openssl in the /etc/httpd/conf.d/ directory since we had given the path of .crt and .key file in the ssl.conf file.

# cd /etc/httpd/conf.d/

# openssl genrsa -des3 -out server.key 1024

The above command will ask for the passphrase. You need this while starting your Apache later. Note enter a strong, passphrase to protect the Apache web server key pair. You can give any name to key instead of “server.key” and you need to remember the key name. You can also use 2048 instead of 1024.

1. Generate a certificate request file (server.csr) using the above server.key file.

# openssl req -new -key server.key -out server.csr

Sample Output:

Enter pass phrase for apachekey.pem:

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

-----

Country Name (2 letter code) [GB]:IN

State or Province Name (full name) [Berkshire]:Maharashtra

Locality Name (eg, city) [Newbury]:Nagpur

Organization Name (eg, company) [My Company Ltd]:Dada International

Organizational Unit Name (eg, section) []:IT

Common Name (eg, your name or your server's hostname) []:amit.di.com

Email Address []:root@amit.di.com

Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []:

An optional company name []:

**Note :-** A challange Passward and optional company name is not neccesary to EnterYou need to provide the information fill or hit [Enter] key to accept defaults, but the Common Name field is very important. You must match the fullyqualified domain name of your server exactly (e.g. sunil.dicom) or the certificate will not work. No need to enter the challenge password.

1. Generate a self signedssl certificate (server.crt) using server.key and server.csr file.

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

1. Start the apache and verify SSL: This will prompt you to enter the password for your private key

#service httpd start

Apache/2.2.17 mod\_ssl/2.2.17 (Pass Phrase Dialog)

Some of your private key files are encrypted for security reasons.

In order to read them you have to provide the pass phrases.

Server www.example.com:443 (RSA)

Enter pass phrase:

OK: Pass Phrase Dialog successful.

1. You can verify whether the ssl starts properly or not by following command which shows the port number 443 is strated.

# netstat -an | grep –iw listen

1. By default Apache SSL runs on 443 port. Open a web browser and verify that you can access your Apache using https://{your-ip-address}
2. You can also verify from another system in the network, then it will ask to add certificate to your web browser.
3. Whenever you start the system then it will ask for passphrase each time.

**Disabling the ssl in HTTPD web server :**

1. Service Stop

# service httpd stop

2. If we comment the “Include” directive in the httpd.conf file and start the httpd service then it will restrict all .conf files in “conf.d” directory. It may cause an error for other programs which uses that directory files.

3. To avoid the above problem we will move the ssl.conf file to any directory.

# cd /etc/httpd/conf.d/

# mkdir ssl

# mv ssl.conf ssl/

4. Please do not touch the other configurations of httpd.conf file. And start the httpd service then it will start properly, not asking for the passphrase.

# service httpd start

5. This is how you disable the ssl in httpd web server. Now if you need to start it again then you need to modify the httpd.conf file to include ssl.conf file.

# vim /etc/httpd/conf/httpd.conf

Include conf.d/\*.conf

Include conf.d/ssl/\*.conf

6. Then restart the httpd service then it will ask for the passphrase that is set for the key file.

# service httpd restart