
# TUTORIAL: How to Generate Secure Self-Signed Server and Client Certificates with OpenSSL

Devolutions

## IT IS NECESSARY TO GENERATE SECURE SELF-SIGNED SERVER AND CLIENT CERTIFICATES

For testing purposes, it is necessary to generate secure self-signed server and client certificates. However, I have found that many tutorials available on the web are complicated, and they do not cover certificates that use safe algorithms. And so, since "necessity is the mother of invention", I decided to create a simple tutorial and share it with all of you!

## Why OpenSSL?

I choose to use OpenSSL because it is available on all platforms (Linux, macOS, Windows) which means this tutorial can be followed on any platforms.

## About the Steps

While there are many steps in this process, please do not worry. My goal is to make this as simple as possible for you, and so I have broken every action down into a single step. This way, everything should be clear, and my hope is that you won't waste time or get frustrated along the way. There is one requirement before starting all of this, **you'll need to have OpenSS**L. Ok, ready? Let's get started!

## Step 1 - Certificate Authority

### Step 1.1 - Generate the Certificate Authority (CA) Private Key

Every certificate must have a corresponding private key. Generate this using the following command line:

```
openssl ecparam -name prime256v1 -genkey -noout -out ca.key
```

This will create a 256-bit private key over an elliptic curve, which is the industry standard. We know that Curve25519 is considered safer than this NIST P-256 curve but it is only standardized in TLS 1.3 which is not yet widely supported.

### Step 1.2 - Generate the Certificate Authority Certificate

The CA generates and issues certificates. Here is a link to additional resources if you wish to learn more about this.

Generate the Root CA certificate using the following command line:

```
openssl req -new -x509 -sha256 -key ca.key -out ca.crt
```

You will be prompted to provide some information about the CA. Here is what the request looks like:

```
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.
```

Below is an example using information that is specific to Devolutions (replace with your own specific information):

```
Country Name (2 letter code) [AU]:CA

State or Province Name (full name) [Some-State]:QC

Locality Name (eg, city) []:Lavaltrie

Organization Name (eg, company) [Internet Widgits Pty Ltd]:Devolutions inc.

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

Common Name (e.g. server FQDN or YOUR name) []:devolutions.net

Email Address []:security@devolutions.net
```

Your CA will be created once you enter your information.

# Step 2: Server Certificate

This step may be repeated for each server you need.

## Step 2.1 - Generate the Server Certificate Private Key

To generate the server private key, use the following command line:

```
openssl ecparam -name prime256v1 -genkey -noout -out server.key
```

This will create the file name server.key.

## Step 2.2 - Generate the Server Certificate Signing Request

To generate the server certificate signing request, use the following command line:

```
openssl req -new -sha256 -key server.key -out server.csr
```

**For maximum security, we strongly recommend that the signing request should only be generated on the server where the certificate will be installed. The server private key should never leave the server!**

You will be prompted to provide some information about the server certificate. You can enter the same information you used for the CA certificate. For example:

```
Country Name (2 letter code) [AU]:CA

State or Province Name (full name) [Some-State]:QC

Locality Name (eg, city) []:Lavaltrie

Organization Name (eg, company) [Internet Widgits Pty Ltd]:Devolutions inc.

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

Common Name (e.g. server FQDN or YOUR name) []:devolutions.net

Email Address []:security@devolutions.net
```

In addition, you will be prompted to create a password. Make sure to use a long, strong, and unique password. Here is an example (do not use this one!):

```
^x^GT+HEy]h9C@8>ZBrb%P>{
```

### Step 2.3 - Generate the Server Certificate

You are now ready to generate the server certificate, which can be done through the following command line:

```
openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt -
```

**This step should only be performed on the Certificate Authority server as the CA private key should never leave the host where it has been generated. You must transfer the signing request to the CA server.**

## Step 3: Client Certificate

This step may be repeated for each client you need.

### Step 3.1 - Generate the Client Certificate Private Key

Use the following command line to create the client certificate private key:

```
openssl ecparam -name prime256v1 -genkey -noout -out client1.key
```

This will create a file named "client1.key".

### Step 3.2 - Create the Client Certificate Signing Request

You need to create a signing request to generate a certificate with the CA. Use the following command line:

```
openssl req -new -sha256 -key client1.key -out client1.csr
```

**For maximum security, we strongly recommend that the certificate signing request should only be generated on the client where the certificate will be installed. The client private key should never leave the client!**

Next, you will be prompted to submit information about the client certificate. You can enter the same information as the CA certificate, except for the last two entries: Common Name and Email Address. These should be the name and email of an individual and not your company. For example:

```
Common Name (e.g. server FQDN or YOUR name) []:John Doe

Email Address []:JohnDoe@devolutions.net
```

You will also be asked to set a password on the certificate signing request. Once again, make sure that you choose a strong and safe password. Here is an example (do not use this one!):

```
^x^GT+HEy]h9C@8>ZBrb%P>{
```

### Step 3.3 - Generate the Client Certificate

You are now ready to generate the client certificate, which can be done through the following command line:

```
openssl x509 -req -in client1.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out client1.crt
```

**This step should only be performed on the Certificate Authority server as the CA private key should never leave the host where it has been generated. You must transfer the signing request to the CA server.**

**We recommend generating a single certificate for each client, as this lets you quickly identify the affected client in the event if an issue or problem. For maximum security, the client private key should remain on the client and never be copied on another host.**

I hope that you've found this tutorial simple and helpful. If you have any questions or comments, please post your feedback below!