Custom SSL certificate creation or Woody servers

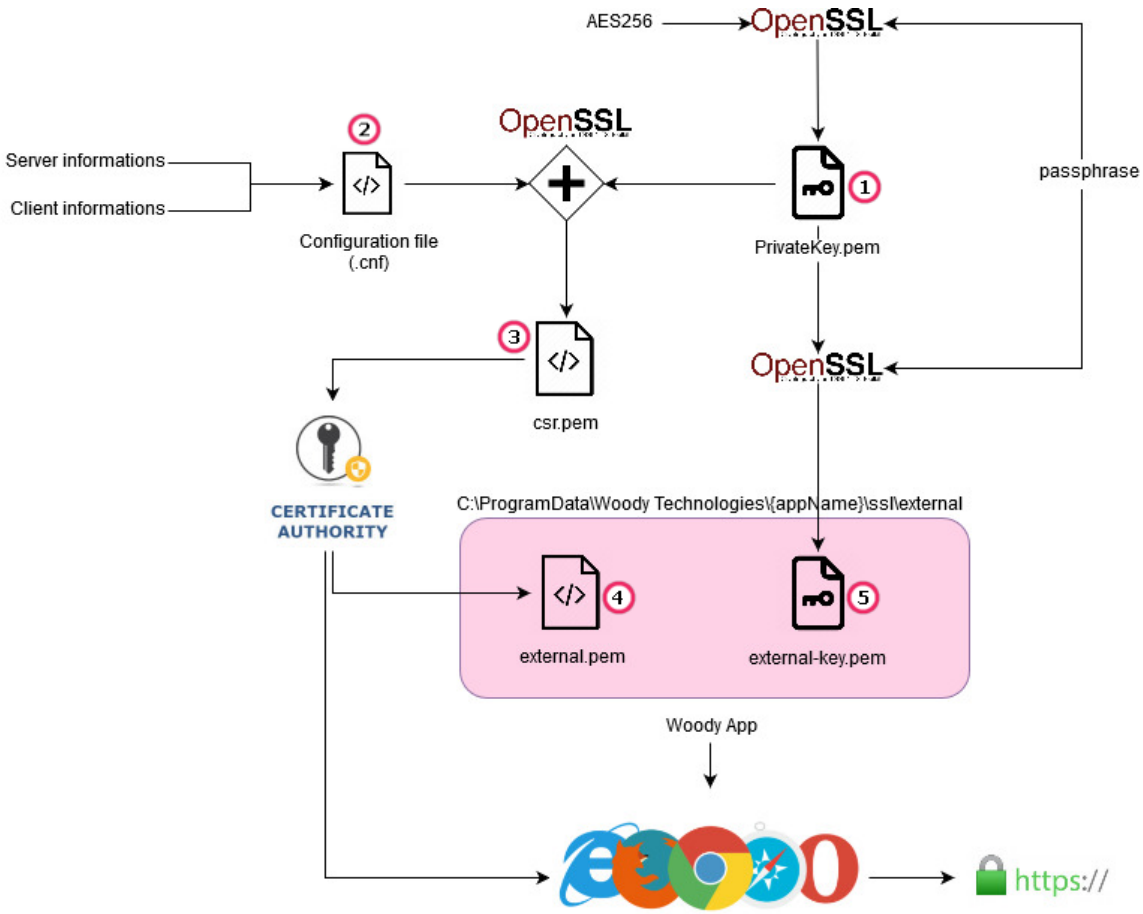
Your IT department should be able to provide a custom **.pem certificate + private key** for Woody servers.
The key must **not** be password protected.

The certificate must be valid for the domain name that you will use to access Woody pages.
If your IT department already provided the certificate and the key, you can directly jump to **step 15** to import it through the Woody configuration wizard.

If your IT department asks you for a certificate request, please follow the steps below

The workflow for custom SSL certificate generation is the following :

- You need first to create a **privateKey** ❶ (with OpenSSL) and a **configuration file** ❷
- With these two files, you can generate a **Certificate Signing Request** ❸ (with OpenSSL)
- Send this CSR to your Certificate Authority (check with your IT department), they will provide a signed certificate.
- You can then import this **signed certificate** ❹ and the **privateKey** ❺ in Woody software

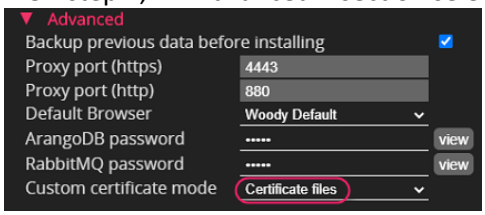


Steps to do it :

1. Install OpenSSL
   more information at: https://tecadmin.net/install-openssl-on-windows
2. Create a working folder (i.e. `C:\Temp`) and put there the config file "`openssl.cnf`" on it.
   see the example next page of this document in the appendix
3. Edit the config file with your values.
   Replace red values from the document in appendix by yours.
4. Open a CMD terminal and go to the working directory that you create at step 2 (i.e. "`cd C:\Temp`")
5. In the terminal, enter:
   `openssl genrsa -aes256 -out private-key.pem`
6. OpenSSL will ask for a private passphrase (we'll remove it later). Enter whatever you want.
   Keep this passphrase, we will need it later.
7. OpenSSL will generate a file : "`private-key.pem`"
8. Enter in the terminal :
   "`openssl req -config openssl.cnf -key private-key.pem -new -sha256 -out csr.pem`"
9. Then confirm your passphrase (same as you entered it at step 6)
10. Confirm (or correct) information of the config file.
11. OpenSSL will generate a "`csr.pem`" file.
    This is your Certificate Signing Request.
12. Enter in the terminal :
    "`openssl rsa -in private-key.pem -out private-key-no-password.pem`"
13. Confirm your passphrase.
    OpenSSL Will generate a private key "`private-key-no-passwd.pem`" without passphrase.

> You can now sent your Certificate Signing Request "`csr.pem`" to your Certificate Authority (IT department).
> Ask for a **.pem certificate** file in return.

14. When you have this signed certificate from your IT, copy it, with the private key (without passphrase) to the Woody server.
15. Run the "Woody Wizard" on the server.
    • On step 4, in "Advanced » section select "Certificate files" for "Custom certificate mode".



    • On step 5, select your **signed certificate** (from yout IT) file and **private key** (generated on step 13)

    • Complete the wizard process

Appendix

Content of "openssl.cnf" :

```
[ req ]
default_bits       = 4096
default_keyfile    = san.key #name of the keyfile
distinguished_name = req_distinguished_name
req_extensions     = req_ext

[ req_distinguished_name ]
countryName                 = FR
countryName_default         = FR
stateOrProvinceName         = France
stateOrProvinceName_default = France
localityName                = Paris
localityName_default        = Paris
organizationName            = Woody Technologies
organizationName_default    = Woody Technologies
commonName                  = woody-technologies.com
commonName_default          = woody-technologies.com
commonName_max              = 64

[ req_ext ]
subjectAltName = @alt_names

[alt_names]
DNS.1   = srv-woody-01
DNS.2   = srv-woody-01.woody-technologies.com
DNS.3   = srv-woody-02
DNS.4   = srv-woody-02.woody-technologies.com
```

Some usefull OpenSSL commands :

• Calculate md5 checksum for a .pem private key :

```
openssl pkey -in external-key.pem  -pubout -outform pem | openssl md5
```

• Calculate md5 checksum for a .key private key:

```
openssl rsa -noout -modulus -in privateKey.key | openssl md5
```

• Calculate md5 checksum for a .pem certificate :

```
openssl x509 -in external.pem  -pubkey -noout -outform pem | openssl md5
```

• Calculate md5 checksum for a .csr certificate signing request :

```
openssl req -in CSR.csr -pubkey -noout -outform pem | openssl md5
```

All these md5 must return the same value.

• reading a .pem file

```
openssl x509 -noout -text -in certificate.pem
```

• Convert .crt in .pem

```
openssl x509 -in mycert.crt -out mycert.pem -outform PEM
```

• Convert .key in .pem

```
openssl rsa -in hostname.key -out hostname.key.pem -outform PEM
```