infotopics
apps for tableau

SSL CERTIFICATES

Creating & configuring

CONTENT

INTRODUCTION

First and foremost, an introduction to this document. SSL certificates are complicated, and the process of acquiring and configuring them is sometimes cumbersome. The goal of this document is to guide you in performing this task, so that you spend less time on getting SSL configured.

DISCLAIMER

First we want to state that all information in this document is captured from several Internet sources. We are not experts on the subject matter, we just happen to be in the same boat as you: Struggling to get SSL-certificates to work.

Therefore, we are not responsible for any consequences or damage, that result from using instructions in this document in any way. At any time, you can research and validate the provided instructions and information on the Internet, or hire experts on the subject matter.

Use of instructions in this document is at your own risk.

ARE YOU STRUGGLING WITH SSL CERTIFICATES, TOO?

Infotopics and **Infotopics | Apps for Tableau** customers often apply SSL certificates into their Applications, generally for security reasons. Most commonly this applies to the application 'Tableau Server', to 'Tableau Dashboard Extensions', or to other supporting applications like Front-end web-portals, all kinds of web-servers like Apache, XAMPP, Nginx and IIS, AppsforTableau Mail Scheduler or even SAML authentication.

We see a lot of customers struggling to get the SSL-certificates right. Sometimes mistakes are made during the certificate request phase. Sometimes the wrong format is downloaded from your Certificates Authority/Certificate Provider. And lastly, in configuring the certificate in your application. This isn't a task that gets repeated often: maybe once a year, and you forget how it worked the last time.

GOAL

The objective is to guide you on this journey, with examples, tips and instructions.

It will show you how you request your certificate, and prepare it for use on Infotopics' & AppsforTableau's most commonly sold applications:

- Tableau Server,
- Tableau Dashboard Extension by AppsforTableau,
- A generic webserver like Apache2.

Technically, all three listed above are Webserver-like deployments with SSL certificates. This document will show examples of commands, how to prepare and convert your certificate files into the desired format, and apply them in these applications.

On the way, you will find comments and tips on how long a validation period you could/should request, how to organize working with SSL certificates, some (but certainly not all) best practices, other security best practices, and so on. We hope this guide solves your struggle with certificates and saves you time to get your application to work.

Hopefully, at the end, you feel confident with a couple commands (check out the 'This is done by comparing the MD5 checksum of both file, .key and .crt file.

```
openssl rsa -modulus -noout -in tableau_yourcompany_com.key | openssl md5
openssl x509 -modulus -noout -in tableau_yourcompany_com.crt | openssl md5
```

The MD5 checksums must match, of course.

The Fast Lane' chapter) to apply and configure your (web-) application with a SSL-certificate in a convenient way.

LET'S DIVE INTO IT

Firstly, some introductory notes on SSL certificates. In this chapter, the SSL certificate structure, naming conventions, other terminology, and the basic layout of the SSL certificate workflow are given.

WARNING – IMPORTANT

SSL certificates are used for security; that is, to secure your application. The goal is to protect your data and ultimately your company and your customers, from hackers, scammers, swindlers and other crooks.

The main part where you should, no, **MUST**, be careful with SSL certificates, are the files where your private key is stored. The name says it all: 'private'. You should not be mailing or handing out all of your certificates files to your suppliers during installation and configuration duties. Handle them like passwords. Most of our customer don't provide us with passwords. When we need a password, they will type it up for us. There is no need to give us a password in plain-text.

The same applies to the 'private key' part of your SSL-certificate files. Don't leave certificates files spreading around on your drives. Only store them at the needed locations, and preferably store them in a password vault/safe, like LastPass or KeePass.

TIP: How to keep your certificate under control: save them in a Password Vault.

What should you store in there? (these items will be explained later)

- The KEY file,
- Optionally the KEY file passphrase,
- Certificate CONF file,
- Certificate file(s) themselves (the .zip downloaded from your CA),
- Notes / links to instructions how to apply them in your application (like a link to this document),
- All server names/hostnames of systems where the certificate is used,
- Expiration date of the certificate.

OVERVIEW OF SSL CERTIFICATES FILES EXTENSION

Often there is confusion about filename extensions (filename.ext) of certificate files. There are some fix extensions but also non-fixed like '.crt' or '.cer'. Or some certificate extension can have two (2) (or more) conventions.

Some file extensions can contain multiple other forms of the SSL-certificate, like a .zip file containing other files. Some SSL certificate files are binary (unreadable) files. Some are encrypted text files. The readable, but encoded, files can be opened with an text editor like Notepad++ and often start with a header to indicate what type of certificate file it is.

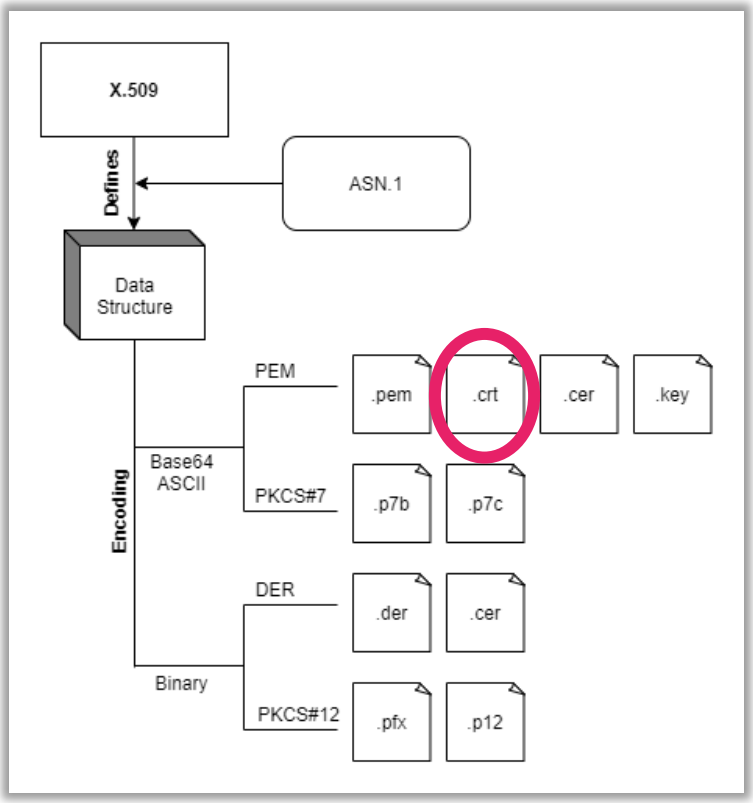
Examples are:

- '---- BEGIN CERTIFICATE----' (.crt or .cer; but .cer can be a binary file, too!)
- '---- BEGIN RSA PRIVATE KEY----- ' (.key)
- '-----BEGIN CERTIFICATE REQUEST-----' (.csr)
- '-----BEGIN PKCS7-----' (.p7b)
- and they have a corresponding end marker like '----END …. ----'.

The SSL-certificates we are talking about, are essentially X.509 certificates. X.509 defines a standard structure of a certificate. A X.509 certificate comes is various formats. It's best practices to use file extensions for those formats, in conjunction with the readable-but-encoded form (base64 encoded) or binary form (DER).

OVERVIEW OF THE SSL-CERTIFICATE STRUCTURE

This overview shows most certificate formats in a glance. It will help when converting a certificate from one format to another, to get the OpenSSL parameters right.



A list of the most common certificate file extensions:

.pem	The PEM file beside the end-entity certificate and intermediate certificates it can include the KEY as well
.crt	Base64 readable certificate file with at least one (1) certificate. Sometimes it is mix up with .cer file naming, and contain the binary .CER format
.cer	Either a DER-binary or base64-readable certificate file
.key	A private KEY file which corresponds with your end-entity certificate
.p7b	A windows format, often containing the full chain of certificates
.p7c	Like p7b
.der	Binary DER-encoded X.509 certificate file
.pfx	Personal Information Exchange format containing all certificates in the certification path and optionally the KEY file. Used on Windows
.p12	Like .pfx
.csr	A file with encoded information concerning a Certificate Signing Request
.crl	Certificate Revocation List (not discussed here)

Windows TIP:
Show extensions and hidden files in File-Explorer (+ E) via 'View' in the top-header, check 'File name extensions' and 'Hidden Items'.
Or 'old-school' via (+ E) -> Option-menu -> Change Folder and search options -> select 'View' Tab and uncheck 'hide extensions for known file types' and check 'Show hidden Files, folders and drives' and hit 'Apply to Folders'.

NAMING CONVENTION

We like the use of a naming convention for the certificate files. Of course, you can figure out one of your own, but this one suits us.

If your domain is 'tableau.yourcompany.com', then we would name the certificate files:

tableau_yourcompany_com.<ext>

Where the '.<ext>' represents the file extensions listed above.

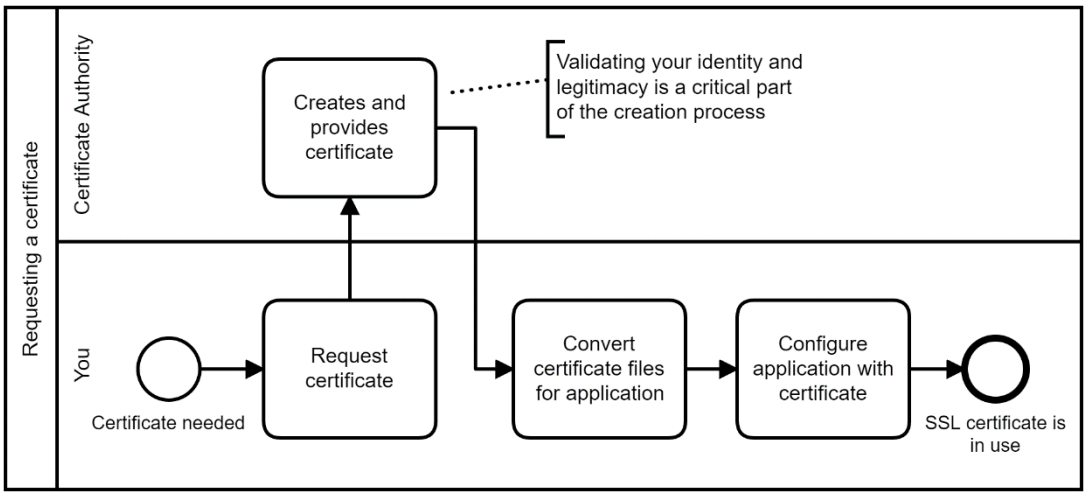
TERMINOLOGY

Besides the filename-extension a number of Terms is used. A non-exhaustive glossary can be found here.

Term	Meaning
CA **Certificate Authority**	The Certificate Authority (CA) delivers your certificate, after you provide them with your certificate-request file (.csr). This body conducts a number of (security) checks to confirm you, as the requestor, as legitimate, and that you are who you say that you are. In the first place using your company's email-address. The certificate is often delivered by e-mail.
Issuer	A Trusted Third Party that issues SSL-certificates to companies after verification.
CA bundle	A file, mostly with .crt or .ca-bundle extension, that contains the root and intermediate certificates. The end-entity certificate along with a CA bundle constitutes the certificate chain. The chain is required to improve compatibility of the certificates with web browsers and other kind of clients so that browsers recognize your certificate and no security warnings appear.
Certification Path	The stack of certificates, used to check the validity of the end-certificate. Top of the certification is the root-certificate, then one or more Intermediate certificates and finally your end-entity certificate. Browsers, and some but not all other types of clients applications, are familiar with a lot of root and intermediate certificates. These root & intermediate certificates are considered valid and trusted. If your end-entity certificate refers to a certification-path which your browser marks as valid, your certificate will also be considered valid and no security warning or error is displayed.
Root certificate	The top certificate of a chain of certificates, provided by a verified organization who's allowed to issue Root and Intermediate certificates.
Intermediate certificate	See above

Base64	Text based Encoding method – not encryption
sha1 **sha256**	Hash algorithms used for encryption. SHA-1 has been deprecated for some years. You should use SHA-256 if possible.
2048 / 4096 bits	Length of your KEY
SAN (Subject Alternative Names)	Multi-domain certificate feature. A single certificate file validates multiple domains.
TLS & SSL	TLS = Transport Layer Security SSL = Socket Layer Security Both are protocols used by websites/webservers to communicate with end-user browsers in your company workspace and/or the Internet.
Windows Certificate Store	Synonym for the Certificate Manager in Windows via the 'Certificates' MMC Snap-in (MMC = Microsoft Management Console) Always choose 'local computer' when working with the Certificate Manager in MMC.

THE 4-STEP WORKFLOW



The workflow consists of four (4) steps:

a) Requesting the certificate
b) CA creates and provides the certificate
c) Converting certificate files for your application
d) Configuring your application with the certificate

WHAT WE NEED: RESULTING CERTIFICATES FILES FOR OUR APPLICATIONS

Basically, all three (3) of the mentioned applications (Tableau Server, Tableau Dashboard Extension and Apache2 webserver) need the same files:

Certificate File	**Tableau Server**	**Tableau Extension**	**Apache2 webserver**
The certificate (CRT) for your domain / hostname	Yes *	Yes *	Yes *
KEY file, corresponding with the above cert file	Yes	Yes	Yes
Optionally: passphrase for KEY file	Optional **	Not possible***	Yes ***?
Certificate Chain file (cacert or ca-bundle) (CRT)	Yes *	Not Possible * & ****	Yes *

* The main (or end-entity) certificate file (.crt) at least must contain the certificate for your domain. Additionally, it can contain all certificates in the Certification Path. Meaning the Root certificate, all Intermediate Certificates and the main Certificate (in opposite order; root at the bottom).

All 3 applications can use this 3-in-1 fullchain certificate file. If you use such a file, you do not have to specify a Certificate Chain file in your application.

** Sometimes your company's security procedures demand that the KEY file is encrypted with a passphrase. If that's the case, then you can do that within Tableau Server after installing the certificate.

*** No passphrase can be configured for Tableau Extensions and Apache2/Nginx webserver (or other webservers). Although tricks to do so for **Apache** and **Nginx** are explained here.

**** For Tableau Extensions, you can't specify a Certificate Chain file.

Outcome:
You need at least:

- One certificate file,
- A Key file,
- Optionally a passphrase for your KEY file,
- Optionally a certificate-chain / ca-bundle file.

REQUESTING YOUR SSL CERTIFICATE

In order to get an SSL certificate, it has to be requested. This chapter explains how to construct such a request.

TIP: If your company has a certificate/security officer who handles certificates, you may want to check with them first in case there is a specific procedure, requirements, templates and so on. It might save you a lot of time and struggle.

▪ CREATE YOUR OPENSSL.CONF FILE

Create a folder from where you work with your SSL-certificates. For example 'D:\tmp' or 'C:\tmp'

TIP: don't make it unnecessarily difficult; use a short path without any spaces.

Copy this openssl.conf template (or your company's own template) into the folder and open it in 'Notepad++' or a similar text editor:

```
[req]
distinguished_name = req_distinguished_name
x509_extensions = v3_req
prompt = no
default_bits = 4096
default_md = sha256
x509_extensions     = SAN
req_extensions      = SAN
extensions          = SAN

[req_distinguished_name]
C = NL
ST = Overijssel
L = Oldenzaal
O = Your-Company-Naam
OU = IT
CN = *.yourcompany.com

[v3_req]
keyUsage = critical, digitalSignature, keyAgreement
extendedKeyUsage = serverAuth

[SAN]
subjectAltName = @alt_names

[alt_names]
DNS.1 = *.yourcompany.com
DNS.2 = yourcompany.com
```

Edit the entries in the 'req_distinguished_name' section. Where:

C	is your country code
ST	is your state code but might be left empty if not applicable
L	Is your companies city
O	your company name (which might be checked by your CA)
OU	your department

Most important are the 'CN' (Common Name) and the DNS entries under 'alt_names'. These are also called SAN, the 'Subject Alternative Name', often referred to as 'multi-domain SSL certificate'.

These should match the domain(s) you want to cover with the certificate. For example, if your domain URL is 'tableau.yourcompany.com'. You can request a wildcard certificate as above for '*.yourcompany.com' but it can be useful to also add 'yourcompany.com' – as you can then use the certificate on that domain as well.

TIP: The SAN list can contain multiple 'hostnames' from within a single certificate. This comes in handy if you want a single certificate, usable for a series of servers and applications.
Often de hostname-without-fqdn (full-qualified-domain name) and hostname-with-fqdn and several URL's for the application are listed. The advantage is less work (only one (1) certificate instead of many), only once the cost of a certificate (although they are not that expensive anymore).

Let's imagine, you have an application with a Development, Acceptance and a Production environment sitting on three different (windows or linux) servers. Then you could specify:

```
CN = tableau.yourcompany.com

DNS.1=tableau.yourcompany.com
DNS.2=tableau-acc.yourcompany.com
DNS.3=tableau-dev.yourcompany.com
DNS.4=svrprdtab01.yourcompany.com
DNS.5=svrdevtab91.yourcompany.com
DNS.6=svracctab51.yourcompany.com
DNS.7=svrprdtab01
DNS.8=svrdevtab91
DNS.9=svracctab51
DNS.10=tableau
DNS.11=tableau-acc
DNS.12=tableau-dev
```

That would make the certificate valid on many trivial aliases for the same application URL.

You can replace the 'yourcompany.com' for the server's active-directory-domain (when not requesting a wildcard certificate? TODO). For example:

```
DNS.4=svrprdtab01.active-directory-domain-alias.local   (or .intra)
```

Where `DNS.4` to .9 are (host)names of the servers.

CN is more of the past, when SAN was not supported yet in the X.509 definitions. It's important to include the CN in the SAN as well, often as the first entry.

IMPORTANT: Well, due to higher and stricter security demands certain browsers will only work with certificates being valid for **maximum of 397** days, when issued after September 1 2020. So, effectively, you'll need to renew the certificate **each year**.

When annually renewing your certificate, you might TODO: Finish this part

▪ SETUP TO WORK WITH OPENSSL

We use **OpenSSL** as the cryptographical toolkit to work with SSL-certificates. You can download it for Windows, Linux or Mac (OS X), and it's often included in software packages. Tableau Server

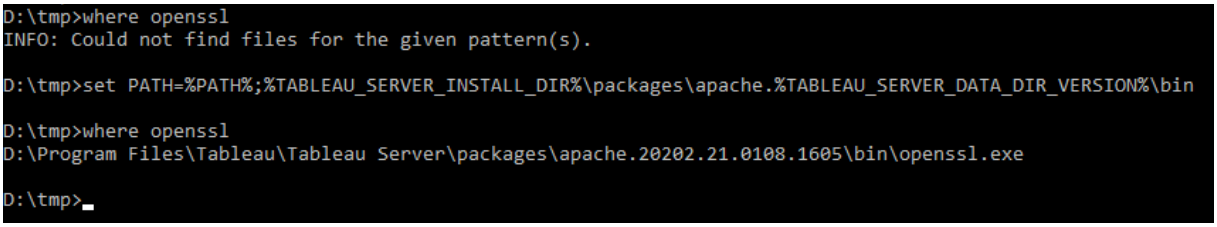
ships with OpenSSL, as does XAMPP (Apache, MySql/MariaDB, PHP, Perl combined package). On Linux, OpenSSL is often already installed. OpenSSL is the most common used tool working with SSL-certificates.

OpenSSL is a command line tool. That makes is it sometimes hard, avoiding typo's and referencing multiple folders/directories with absolute or relative pathnames containing spaces. To work with OpenSSL from any folder/directory, we add the location of the OpenSSL program to the PATH system variable. Both on Windows as on Linux. In this way you don't need to add absolute or relative pathnames while typing OpenSSL commands.

- ## OPENSSL ON WINDOWS WITH TABLEAU SERVER

Start your dos-cmd-shell/box by pressing + R and enter 'cmd'.

TIP: be sure to enable 'Quick Edit Mode' in the default or properties settings



Copy/Paste the following commands:

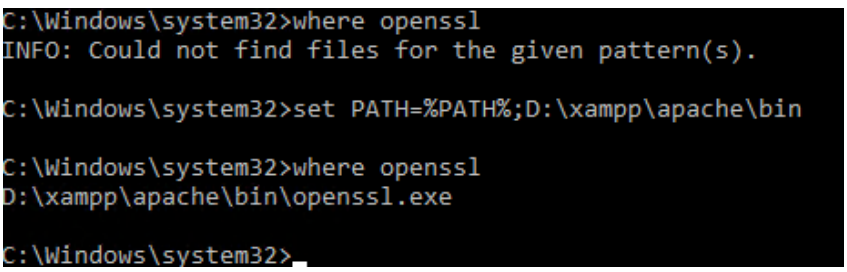
```
where openssl
set
PATH=%PATH%;%TABLEAU_SERVER_INSTALL_DIR%\packages\apache.%TABLEAU_SERVER_DA
TA_DIR_VERSION%\bin
where openssl
```

Note: the 'set PATH=…..' command is a one-liner, as shown in the printscreen.

Now you can run openssl from within any folder where you are located in this dos-cmd-shell.
When you open a second dos-cmd-shell, you have to repeat these commands.
Common practices is to download the certificates files into a TEMP folder like here 'D:\tmp'.

- ## OPENSSL ON WINDOWS WITH XAMPP

Start your dos-cmd-shell/box by pressing + R and enter 'cmd':



Commands:

```
where openssl
set PATH=%PATH%;D:\xampp\apache\bin
where openssl
```

OPENSSL ON LINUX

On Linux, OpenSSL is generally already installed and added to the PATH variable.

To verify, start your Linux shell using putty or SSH.
Commands:

```
which openssl
export PATH=$PATH:/usr/bin
which openssl
```

If the output of the first 'which' command already finds 'openssl' (in '/usr/bin' or any other directory), the 'export' command is redundant.

HELP REGARDING OPENSSL

If you need help with the syntax of the OpenSSL command, in your dos-cmd-shell or linux-shell type:

```
openssl.exe help
```

(Optionally adding a specific keyword):

```
openssl.exe help rsa
```

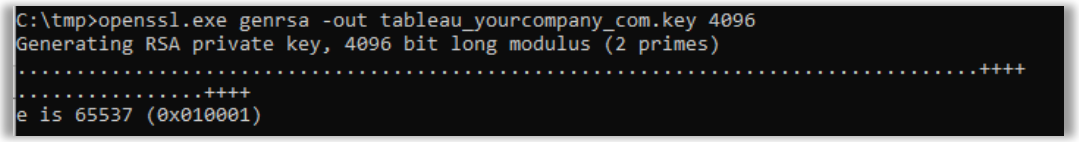
Or search the Internet for it, ideally with the error/warning message you are getting (if any).

GENERATE YOUR CERTIFICATE KEY FILE

In your dos-cmd-shell or linux-shell, run the command (on linux without '.exe' behind it):

```
openssl.exe genrsa -out tableau_yourcompany_com.key 4096
```

Output:



The start of the 'tableau_yourcompany_com.key' file looks like:

```
-----BEGIN RSA PRIVATE KEY-----
MIIJKQIBAAKCAgEA2tUHhW/4SmBS15lN3gqr04iVAKg4BjXrPP/dvO66Jb8XW5PY
W5HKE
.
.
.
YiXc6gnGWEeMXtNdMmmZYPaGnKeYGCi3eDiBK/u6bSlvnGhS8Jin8GNQQb11
-----END RSA PRIVATE KEY-----
```

This will generate an RSA KEY file of 4096 bits. The longer/more bits, the harder it is to break the key. 2048- or 4096-bit keys are pretty standard nowadays. 1024-bit keys are considered insecure.

Note: the CONF file is not used in the command. It's just a (private) secret.

- GENERATE YOUR CERTIFICATE REQUEST CSR FILE

In your dos-cmd-shell or linux-shell, run the command:

```
openssl.exe req -new -key tableau_yourcompany_com.key -config
tableau_yourcompany_com.conf -out tableau_yourcompany_com.csr
```

No output is generated if the command executes without errors. Instead, the CSR file is created.

The 'tableau_yourcompany_com.csr' contains:

```
-----BEGIN CERTIFICATE REQUEST-----
MIIFMjCCAxoCAQAwfDELMAkGA1UEBhMCTkwxEDAOBgNVBAgMB1V0cmVjaHQxEzAR
.
.
.
wkP9uzCenJdv0hRM6Q8NsRQ+zj9NDM9ZSOdkYYvsxAl+jQyDtHI=
-----END CERTIFICATE REQUEST-----
```

You can extract information in the CSR file with:

```
openssl.exe req -in tableau_yourcompany_com.csr -text -noout [ -config
tableau_yourcompany_com.conf ]
```

(This should work without the -config parameter. But without it, it'll returns an error message.)

- SEND CSR TO YOUR CERTIFICATE PROVIDER/CA

Either you or your company's certificate/security officer will request the new certificate, depending on the policies in place.

Request a new certificate, based on your type of certificate on your provider's website, like HTTPS://SSLCERTIFICATEN.NL (language can be changed). Either a 'wildcard', 'multi-domain' (multiple SAN entries) or 'single-domain' certificate. For example, select a Wilcard certificate from Sectigo / PositiveSSL with Domain validation, or a Multi-Domain certificate from Sectigo / PositiveSSL.

Upload or paste content of your CSR and provide company information and so on
Your certificate provider may have has different offerings. Depending the type of the certificate and its validation process, it can take 15 minutes to a number of days before your certificate request is processed and your certificate is ready to be downloaded.

TAKING DELIVERY OF YOUR CERTIFICATE

Once the Certificate Authority has received and completed your request, you will be able to take delivery of the certificate. This chapter discusses receiving and transforming these delivered certificates in such a way that they can be applied to your use case.

▪ RECEIVING YOUR CERTIFICATE

You will be notified either via your certificate/security officer, or through an e-mail from your certificate provider, when the certificate is ready. Your certificate/security officer will deliver the certificate files to you, or you can download them from your certificate provider's website.

Typically, a bunch of different files can be downloaded. A short, non-exhaustive overview:

a) The end-entity certificate, the one it's all about, might be downloadable as a single .CRT file. For instance '*_yourcompany_com.crt' or 'tableau_yourcompany_com.crt'
b) A zip-bundle '*_yourcompany_com.zip'. It will contain various formats of the certificate among the above mentioned .crt, and folders with, a windows style .p7b, a Linux styled, a Plesk/cPanel/DirectAdmin styled, and/or Apache/Nginx styled format.
c) For Plesk, cPanel or DirectAdmin: 2 files 'tableau_yourcompany_com-crt.txt' and 'tableau_yourcompany_com-cacert-crt.txt'. Probably being the end-entity certificate and the ca-bundle, aka cacert; the files can be renamed from '.txt' to '.crt'. The latter may contain multiple certificates of the Root and Intermediate certificates.
d) A 'Linux' styled variant which, like the CAcert under c), contains the full certification-path certificates. Likely named as 'tableau_yourcompany_com.ca-bundle'. It probably contains the Root and Intermediate certificates. The '.ca-bundle' file may be renamed to '_cacert.crt' or '_ca-bundle.crt'
e) An 'Apache/Nginx' style which may be named like 'tableau_yourcompany_com-fullchain.txt'. The file can be renamed from '.txt' to '.crt', and probably will contain at least three (3) certificates: your end-entity certificate and Intermediate and Root certificate. Likely in this order; first your end-entity cert, then at least one Intermediate cert, and last the root cert.
f) The Windows styled .p7b cert file. It contains the full chain of certificates.
g) The Windows binary .PFX cert file. It contains the full chain of certificates and may – but that's not a certainty! – contain the private KEY. See Troubleshooting & FAQ.
h) When your certificate/security officer replies that the certificate has be upload in the Windows Certificate Store, this often leads to trouble as it might be unclear if the certificate is installed as 'exportable' and with the private KEY file.

▪ WHAT FORMATS DO WE NEED?

As explained in paragraph

What We Need: Resulting certificates files for our applications .

We only need two (2) files:
- Full chain of certificates in a single CRT file,
- Unencrypted KEY file.

CONSTRUCT THE FULLCHAIN CERTIFICATE FILE

In dos-cmd or Linux-shell, run the commands (on Linux it is 'cp' to copy and more is less):

```
copy tableau_yourcompanay_com.crt tableau_yourcompany_com-fullchain.crt
more tableau_yourcompanay_com-cacert.crt >> tableau_yourcompany_com-
fullchain.crt
```

Open 'tableau_yourcompany_com-fullchain.crt' with a text editor (Notepad++ or Nano), and make sure the END and BEGIN markers are not on the same line:

```
…
      CgDKbaLi6PAPt2Qei2s3xflVk1N40hQjDvJxqLB+MFWBjQf3
-----END CERTIFICATE----------BEGIN CERTIFICATE-----
MIIGEzCCA/ugAwIBAgIQfVtRJrR2uhHbdBYLvFMNpzANBgkqhkiG9w0BAQwFADCB
…
```

Change it to:

```
…
      CgDKbaLi6PAPt2Qei2s3xflVk1N40hQjDvJxqLB+MFWBjQf3
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIGEzCCA/ugAwIBAgIQfVtRJrR2uhHbdBYLvFMNpzANBgkqhkiG9w0BAQwFADCB
…
```

Save your fullchain CRT file as it is ready for deployment.

EXTRACT FULLCHAIN CRT FILE FROM WINDOWS P7B FILE

In dos-cmd or linux-shell, run the commands (on linux it is 'cp' to copy and more is less):

```
openssl.exe pkcs7 -print_certs [ -inform der ] -in
tableau_yourcompany_com.p7b -out tableau_yourcompany_com-fullchain.crt
```

```
Use "-inform der" if there is an error like "unable to load PKCS7"
```

Now open 'tableau_yourcompany_com-fullchain.crt' with an editor (Notepad++ or nano), and remove all lines NOT between "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----" throughout the file.

EXTRACT FULLCHAIN CRT AND KEY FILE FROM PFX FILE

Of course, you can only extract the KEY file if it is included in the PFX file.

The PFX file is password protected. To extract the CRT and/or KEY file, you need to specify the password of the PFX file.

In dos-cmd or linux-shell, run the commands (on linux it is 'cp' to copy and more is less):

```
openssl pkcs12 -in tableau_yourcompany_com.pfx -nocerts -out
tableau_yourcompany_com-key.pem -nodes
```

Enter Import Password:

```
openssl rsa -in tableau_yourcompany_com-key.pem -out
tableau_yourcompany_com.key

openssl pkcs12 -in tableau_yourcompany_com.pfx -nokeys -out
tableau_yourcompany_com.pem
```

Enter Import Password:

```
openssl x509 -in tableau_yourcompany_com.pem -out tableau_yourcompany_com-
fullchain.crt
```

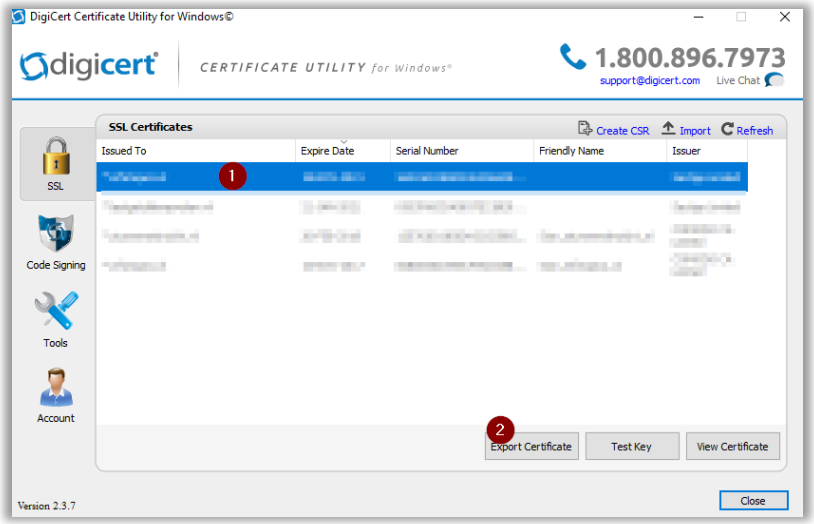
TODO: check the result in tableau_yourcompany_com-fullchain.crt. It looks a bit small.

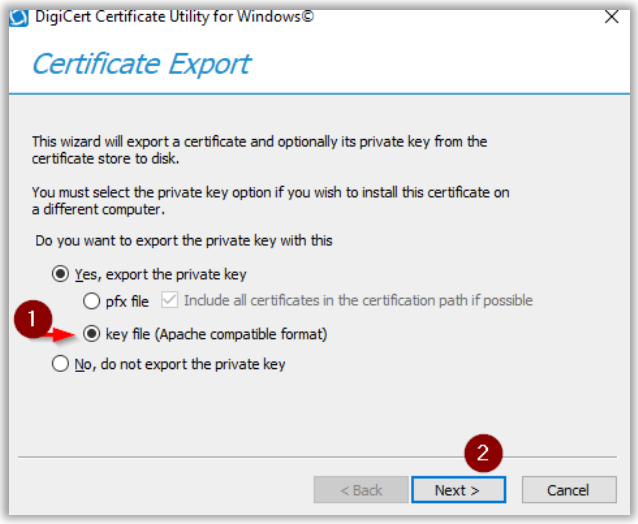
Alternatively, install the PFX in a windows certificate store marked as 'exportable' and follow the procedure below with DigiCertUtil.

See also: Useful Links

EXPORT FULLCHAIN CRT AND KEY FILE OUT OF THE WINDOWS CERTFICATE STORE WITH DIGICERTUTIL

Download the free utility DigiCertUtil from **https://www.digicert.com/support/tools/certificate-utility-for-windows**. Unzip it and run DigiCertUtil.exe'; then click the buttons indicated in the graphics.





Save the KEY in a folder of your choice. In the same folder you will find 'tableau_yourcompany_com.crt' and CAcert.crt. You can construct a fullchain.crt with both CRT files export by DigiCertUtil as described above.

VERIFY CERTIFICATE KEY PAIR

This is done by comparing the MD5 checksum of both file, .key and .crt file.

```
openssl rsa -modulus -noout -in tableau_yourcompany_com.key | openssl md5
openssl x509 -modulus -noout -in tableau_yourcompany_com.crt | openssl md5
```

The MD5 checksums must match, of course.

THE FAST LANE

This chapter contains a highly condensed version of chapter 3 and 4.

First, create your openssl.conf file as described in paragraph 'Create your openssl.conf file'.

We'll assume your working folder/directory is: D:\tmp. If it isn't, substitute it for your actual working directory.

▪ CREATE YOUR CSR AND KEY FILE IN A SINGLE COMMAND

In your dos-cmd-shell or linux-shell, run the command (on linux without '.exe' behind it):

```
cd /d D:\tmp

openssl.exe genrsa -out tableau_yourcompany_com.key 4096

openssl.exe req -new -key tableau_yourcompany_com.key -config
tableau_yourcompany_com.conf -out tableau_yourcompany_com.csr
```

TODO: I could not find a one-line command to generate both KEY and CSR in the desired format. (either encrypted or non-RSA).

▪ REQUEST CERTIFICATE

Request your certificate via your CA's website or your company's certificate/security officer based on the CSR file.

IMPORTANT: Never send anyone your KEY file. Keep it to yourself (or perhaps only share it with your certificate/security officer).

▪ ON CERTIFICATE DELIVERY: GET OR CONSTRUCT THE 'FULLCHAIN' CRT FILE

Download the fullchain CRT file, or Construct the fullchain CRT file from the end-entity CRT and CAcert/CA-bundle CRT file.

In dos-cmd or linux-shell, run the commands (on linux it is 'cp' to copy and more is less):

```
copy tableau_yourcompanay_com.crt tableau_yourcompany_com-fullchain.crt
more tableau_yourcompanay_com-cacert.crt >> tableau_yourcompany_com-
fullchain.crt
```

Open 'tableau_yourcompany_com-fullchain.crt' with an editor (Notepad++ or nano), and make sure the END and BEGIN markers are not on the same line:

```
…
      CgDKbaLi6PAPt2Qei2s3xflVk1N40hQjDvJxqLB+MFWBjQf3
-----END CERTIFICATE----------BEGIN CERTIFICATE-----
MIIGEzCCA/ugAwIBAgIQfVtRJrR2uhHbdBYLvFMNpzANBgkqhkiG9w0BAQwFADCB
…
```

Change it to:

```
…
       CgDKbaLi6PAPt2Qei2s3xflVk1N40hQjDvJxqLB+MFWBjQf3
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIGEzCCA/ugAwIBAgIQfVtRJrR2uhHbdBYLvFMNpzANBgkqhkiG9w0BAQwFADCB
…
```

Save your fullchain CRT file as it is ready for deployment.

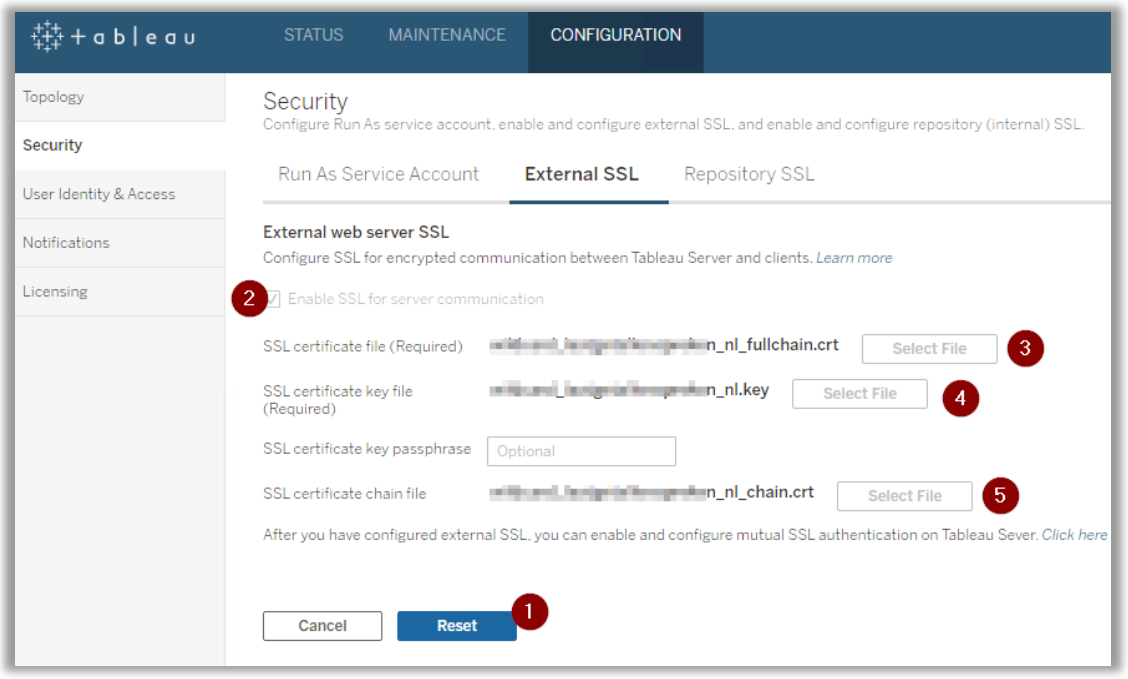
CONFIGURE YOUR APPLICATIONS WITH THE CERTIFICATE

This chapter discusses the process of applying your SSL certificate to a few applications, specifically Tableau Server (and SAML), Tableau Dashboard Extensions, and Apache2 & Nginx web servers.

▪ TABLEAU SERVER SSL-CERTIFICATE CONFIGURATION

For help on this procedure, see **https://help.tableau.com/current/server/en-us/ssl_config.htm**

- Setup an RDP session, by pressing ⊞ + R and enter 'mstsc', to your server where Tableau Server is installed.
- Open a browser to **https://servername:8850** (or **https://localhost:8850** and continue on the certificate warning.
- Log in with a local adminstrator account.
- In the top menu, navigate to 'Configuration'.
- In de side menu select 'Security', and then 'External SSL'.
- If a certificate is already configured, hit 'Reset'.
- Check 'Enable SSL for server communication'.
- Specify your certificate CRT and KEY files.
- Optionally, specify the chain (cacert/ca-bundle CRT file).



- Click 'Save pending changes'.
- Click 'Pending Changes' in the top right corner.
- Click 'Apply Changes and Restart' during a service window as a restart may take 10 to 15 minutes.
- After the restart, check if the certificate is valid in the browser.

To tighten security a bit more you might apply some hardening to your Tableau Server. See: **https://help.tableau.com/current/server/en-us/security_harden.htm**

In dos-cmd or linux-shell, run these commands:

```
tsm configuration set -k ssl.protocols -v "all -SSLv2 -SSLv3 -TLSv1 -
TLSv1.1"
tsm pending-changes apply
```

TABLEAU SERVER WITH SAML AUTHENTICATION USING SSL-CERTIFICATE

Although this is not advised, you can use the same fullchain or end-entity CRT file and KEY to configure SAML authentication. See **https://help.tableau.com/current/server/en-us/saml.htm** and **https://help.tableau.com/current/server/en-us/saml_requ.htm#Cert_Name**

CONFIGURE TABLEAU DASHBOARD EXTENSION WITH A SSL CERTFICATE

In dos-cmd or linux-shell, change directory to the folder where the extension has been unzipped:

```
cd /d “d:\Program Files\Tableau\Extensions\Supertables”
```

And run the extension with the cert and key parameters. In this example the supertables extension:

```
super-tables-win.exe --port 443 --cert tableau_yourcompany_com-
fullchain.crt --key tableau_yourcompany_com.key
```

That’s it!

APACHE2 & NGINX WEB SERVER CONFIGURATION

This process depends on the operating system (Windows or Linux) where the certification CRT and KEY files are stored. Between Apache2 and Nginx, configuration is also different.

For now, an example of an Apache virtual-host config with SSL will explained.

Within Apache2, you define a host in the ‘https-vhosts.conf’ file, like:

```
<VirtualHost *:443>
	ServerName pdfmail.infotopics.nl
		DocumentRoot "d:/xampp/htdocs/Scheduler_v1.7.6/public"
	<Directory  "d:/xampp/htdocs/Scheduler_v1.7.6/">
		Options Indexes FollowSymLinks MultiViews
		AllowOverride All
		Require all granted
	</Directory>

	ServerAdmin admin@example.com
	ErrorLog "D:/xampp/apache/logs/error_pdfmail.log"
	TransferLog "D:/xampp/apache/logs/access_pdfmail.log"
```

```
        SSLEngine on
        SSLOptions +ExportCertData +StrictRequire +OptRenegotiate +StdEnvVars

        SSLCertificateFile "D:/xampp/apache/conf/ssl.crt/star_infotopics_nl.crt"
        SSLCertificateKeyFile "D:/xampp/apache/conf/ssl.key/star_infotopics_nl.key"
        SSLCertificateChainFile "D:/xampp/apache/conf/ssl.crt/CACert.crt"
</VirtualHost>
```

Please note that the folder name for the CRT files and the KEY file, differ slightly. They are located in 'ssl.crt' and 'ssl.key' folder separately.

You can specify the fullchain CRT file in the 'SSLCertificateFile' variable and comment out the 'SSLCertificateChainFile' variable.

You have to restart the web server to apply the changes.

RENEW YOUR CERTIFICATE

This can be done is several ways. We have no preference over which method is the best. But often we just handle it as being a new certificate request as describe above.

TODO: explore other ways to renew

Ways to renew your certificate:

- Handle it as if it was a new certificate
- Renew via windows mmc-certificate snap-in with a new key
- Renew via mmc-certificate snap-in with the same key
- ….

Especially, when you only receive the public certificate (.CRT) file for your URL, validate that the new .CRT and existing .KEY file match with each other. See chapter .

SELF SIGNED CERTIFICATES

Using a self-signed certificate, means that it can only be used on company controlled workspaces.

WARNING: We do NOT recommend the use of self-signed certificates.

Browsers need to validate certificates via certification-authority certficates, which are already available to the browsers for most certificate providers like DigiCert, Comodo, UserTrust, Sectigo, VeriSign, Go Daddy, Thawte are installed on workspaces by default.

A self-signed certificate must be validated by either your company own certification-authority certficates or the self-signed certificate itself. Meaning that these need to be installed on all workspaces of enduser using a browser to use the Tableau Server application and Tableau Extensions.
The operations needed to keep self-signed certificates up-to-date on those workspaces is much higher than buying a official SSL-certificate from one of the above suppliers.

I some cases users who work with Tableau Desktop get error message when signing in to Tableau Server equipped with a self-signed certificate, which can not be validated.

Ok. So if you insist to use a self-signed certificate, here it the command to created a standalone SSL-certificate without certificate-authority chain:

```
openssl x509 -req -sha256 -in tableau_yourcompany_com.csr -out
tableau_yourcompany_com.crt -signkey tableau_yourcompany_com.key -days 397
```

Validate to a maximum of 397 days, as modern browser do not except longer validation periods.

Have the certificate installed on the 'Computer Account' (not a Personal User Account) of all workspaces where it will be used. So, only the .crt file!

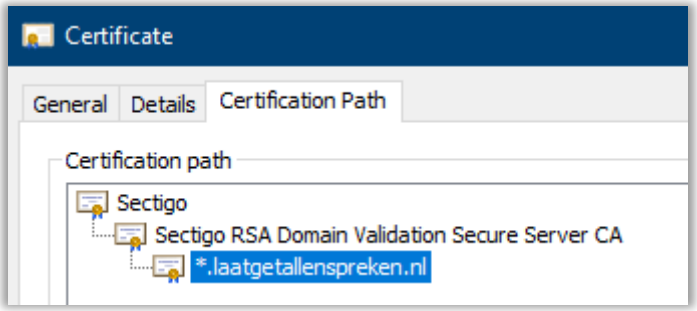
TROUBLESHOOTING & FAQ

Q: Where can I find DigiCertUtil?
A: **https://www.digicert.com/support/tools/certificate-utility-for-windows**

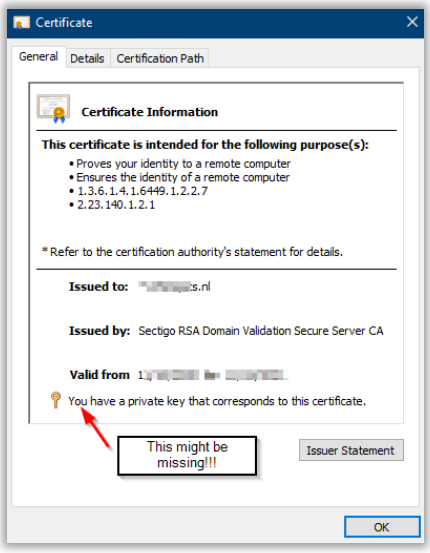
Q: How to verify if a .CRT file contains a single or multiple and if so, which ones?
A: paste the content of a .CRT file in **https://www.xolphin.com/decoder**
Or run:

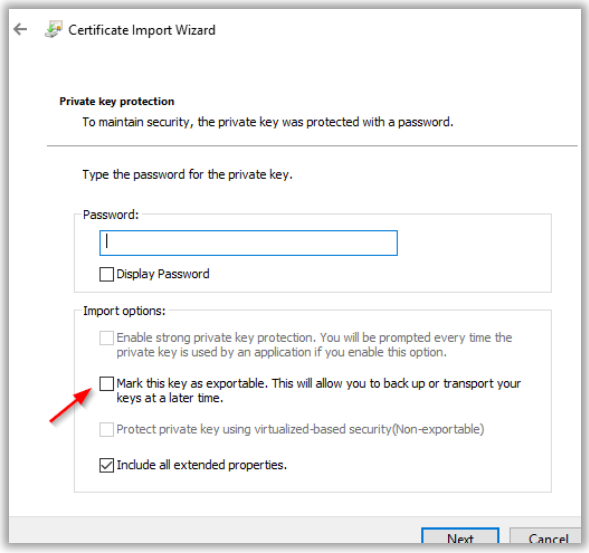
```
openssl.exe x509 -noout -text -in tableau_yourcompany_com.crt
```

Q: How can I find the Certification Path?
A: in the browser while on the website, view the certificate in front of the URL and select the 'certification path' tab. Or double-click the .CRT file.



Q: Why can I not export the certificate from the Windows Certificate Store (MMC-Certificate Snap-in) as .PFX or KEY format?
A: The certificate was installed without the private KEY. Or it was not installed with 'exportable' checked.





Q: Do I have to attach the KEY to a certificate in the Windows Certificate Store?
A: Don't know, yet. But is possible.

Q: How to make it exportable although not install as exportable from the Windows Certificate Store?
A: This will probably never be possible, due to security reasons.

Q: Why does SAML authentication not work with Tableau Server?
A: A lot can go wrong; here are five (5) common issues:

- Your KEY file is not a RSA KEY file. The first line should contain '---- BEGIN RSA PRIVATE KEY'.
- Tableau configuration parameter 'wgserver.saml.sha256' may not be set to 'true'.
- The SAML-assertion may lack the attribute 'username', or if it is specified not contains a username known to your Tableau Server.
- During metadata file exchange between your IdP and SP (Service Provider=Tableau Server) the XML file was corrupted. Copy/Pasting from a webpage might not be OK, better is right-click as choose 'download as'.
- You might want to research and set the tableau configuration parameter 'wgserver.saml.maxauthenticationage' to 2073600. As a SAML assertions has a limited validation period which might not be adjust between your Idp (f.i. Auzre AD) and Tableau Server (the Service Provider (SP)).

USEFUL LINKS

Comprehensive List of Certificate Extensions
https://knowledge.digicert.com/generalinformation/INFO2824.html

How to convert a certificate into the appropriate format
https://knowledge.digicert.com/solution/SO26449.html

https://stackoverflow.com/questions/13732826/convert-pem-to-crt-and-key

Common Name and Subject Alternative Name compatibility
https://www.digicert.com/faq/subject-alternative-name-compatibility.htm

https://www.digicert.com/tls-ssl/multi-domain-ssl-certificates

Explanation about SSL certificates in Dutch, English or German
https://www.sslcertificaten.nl/support/OpenSSL

Maximum SSL/TLS certificate Validity is now One (1) year
https://www.globalsign.com/en/blog/maximum-ssltls-certificate-validity-now-one-year

Tableau Server SSL-certificate configuration
https://help.tableau.com/current/server/en-us/ssl_config.htm

https://www.namecheap.com/support/knowledgebase/article.aspx/986/69/what-is-ca-bundle/

https://www.ssls.com/knowledgebase/what-are-certificate-formats-and-what-is-the-difference-between-them/

https://www.tutorialsteacher.com/https/ssl-certificate-format

https://support.dnsimple.com/articles/what-is-ssl-san/