Server SSL Certificate

If you maintain Agiloft on your own server, you will need to update the SSL security certificate occasionally. The procedure to update the certificate on your server will depend on the type of web server you have integrated with Agiloft. We recommend Nginx as the front-end web server for stability and performance. If Agiloft is integrated with Nginx on your server, please follow these steps to upload a new certificate.

This page also includes information about generating self-signed certificates for testing purposes, and converting a certificate to the correct format for other built-in Agiloft modules like SAML SSO.

Updating the SSL Certificate

You need to update the SSL security certificate occasionally. The procedure to update the certificate on your server depends on the type of web server you have integrated with Agiloft. We recommend Nginx as the front-end web server for stability and performance. If Agiloft is integrated with Nginx on your server, please follow these steps to upload a new certificate.

Converting to PEM Format

If your SSL certificate and private key aren't already in PEM format, you first need to convert them. Nginx and most other applications accept certificates in PEM format. If they are already in PEM format, you can skip to the next section.

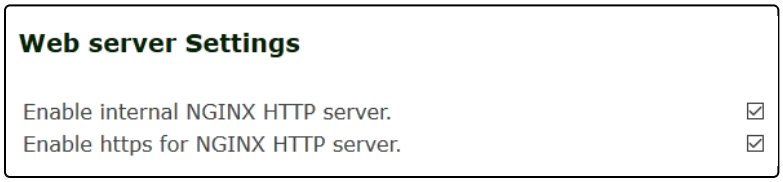
1. Open the OpenSSL utility included in the Agiloft package here: `\Agiloft\wildfly\bin\openssl.exe`
2. Use these commands to make changes:
    - To convert the certificate from .cer to .pem format: `openssl x509 -inform der -in certificate.cer -out certificate.pem`
    - To convert the certificate from .pfx to .pem format: `openssl pkcs12 -in keyStore.pfx -out keyStore.pem -nodes`
    - To extract the key from the certificate: `openssl x509 -inform der -in certificate.cer -pubkey -noout > certificate_publickey.pem`
    - To remove the password from the key file: `openssl rsa -in privateKey.pem -out newPrivateKey.pem`
3. Now you can copy the PEM-formatted SSL certificate and private key into a directory on the server, such as C:\certs\.

Updating the SSL Certificate

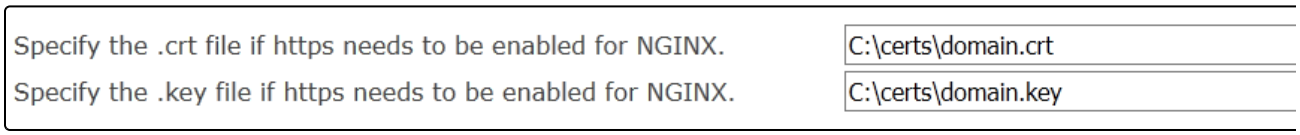
If Agiloft is integrated with Nginx on your server, please follow these steps to upload a new certificate.

1. Copy the SSL certificate and the private key for it to a directory on the server, for example: `C:\certs\`. Make sure the certificate and the private key are in PEM format.
2. Run the Setup.exe utility found under the Agiloft installation directory and access the server through HTTP on port 8888 with a web browser. The browser will show "Agiloft Setup Assistant".

3. Click the "Web Server" link to bring up the Web server settings page.
4. Select the checkboxes to enable Nginx HTTPS.



   Note: Nginx will reserve ports 80/443, so please set a different port for Jboss's internal web server - for example 8080/8443 - while enabling Nginx.
5. Scroll to the bottom of the page and specify the full path to the certificate and private key as shown in the screenshot below.



   Note: The certificates must be in the PEM format.
6. Click the "Change web server settings" button to apply and restart the application.
7. Test the new certificates by accessing the site via HTTPS through a browser, at https://www.<domain>.com /gui2/login.jsp.
   Note: If Agiloft is integrated with some other web server like IIS, please refer to the corresponding documentation from the vendor.

Generating a Self-Signed SSL Certificate

If you don't intend to purchase a certificate for the server, such as in cases where you are installing Agiloft for testing purposes only, you can install a self-signed certificate for the host.

You can generate a self-signed certificate using OpenSSL on a Linux system. On a Windows system, you can access the tool under `C:\Agiloft\wildfly\bin` or find an alternative source online.

Example

Below is a typical command to generate a self-signed certificate in PEM format:

```
openssl req -x509 -newkey rsa:4096 -keyout key.pem -out cert.pem -sha256 -days
365 -nodes
```

Converting to Java Keystore Format

Several of Agiloft's built-in modules, like SAML SSO, require the SSL certificate in Java keystore format. You can use the ssl-cert-convert utility to convert the file.

Example

Below is a sample run of the tool.

```
C:\> C:\Agiloft\bin\ssl-cert-convert.exe
2009-08-24 22:32:42  Select input file format (may be set via -mode command line
option):

PEM SSL certificate and private key in separate files (Apache default) [1,
Enter], PFX or PKCS12 SSL certificate and private key in a single file (typical
IIS export format) [2]
==>1
2009-08-24 22:32:50  Enter the path to the SSL certificate or the PFX file
containing the certificate and private key. The path may be set by the -cert
command line option.
==>cert.pem
2009-08-24 22:33:02  Enter the path to the file containing the private key. The
path may be set by the -key command line option.
==>key.pem
2009-08-24 22:33:09  Enter the password for accessing the private key, or press
Enter if the private key is not protected by a password. This may be set by the -
keypass command line option.
==>
2009-08-24 22:33:15  Enter the path to the keystore to be created or updated.
After successfully creating this file, it may be used as your EnterpriseWizard
SSL certificate. This may be set by the -out command line option.
==>new_cert.keystore
2009-08-24 22:33:32  Enter a password to protect the keystore to be created. If
the keystore already exists, enter the existing password this keystore. This may
be set by the -outpass command line option.
==>NewPassword1
2009-08-24 22:33:43  openssl pkcs12 -export -in server.crt -inkey server.key -out
/tmp/sslcertconvert6771932136351170198p12 -passout pass:rmi+ssl
2009-08-24 22:33:43  /usr/local/EnterpriseWizard/jre/jre/bin/keytool -
importkeystore -srckeystore /tmp/sslcertconvert6771932136351170198p12 -
srcstoretype PKCS12 -srcstorepass rmi+ssl -destkeystore chap8.keystore -
deststorepass rmi+ssl -noprompt
```