

Using SSL certificates with Syracuse & MongoDB

Mike Shaw

22/09/2021



Contents



- **Introduction**
- **Sage X3**
 - MongoDB
 - Syracuse

Introduction

Terminology

SSL (Secure Sockets Layer)	Network security protocol developed by Netscape in 1995. Superseded by TLS, but still widely used as a synonym for TLS
TLS (Transport Layer Security)	IETF (Internet Engineering Task Force) security protocol, first published in 1999 https://www.ietf.org/blog/tls13/
HTTPS	Secure HTTP (Hypertext Transfer Protocol) Application protocol for web browsing https://developers.google.com/search/docs/advanced/security/https https://wiki.mozilla.org/Security/Server_Side_TLS
Public Key Infrastructure (PKI)	Software and procedures needed to create, manage, distribute and revoke digital certificates and manage public-key encryption https://en.wikipedia.org/wiki/Public_key_infrastructure
Certificate Authority (CA)	Organisation that issues certificates and vouches for the identity of Subjects
Subject	Identifies the entity associated with a Public Key e.g. Server name, person or organisation
Digital certificate (aka: Public key certificate)	Digital certificate that identifies a subject and the Subject's Public Key. Digitally signed by a CA
Digital signature	Result of encrypting information with the private key of public/private key pair
Public-key encryption (aka: One-Way or Asymmetric encryption)	Private key is known only to the subject. Public key is shared with the world. <ul style="list-style-type: none">- Data encrypted using the Private key can only be decrypted with the Public key- Data encrypted using the Public key can only be decrypted with the Private key

Why?

Security objective	Purpose	Technical solution
Confidentiality	Prevent eavesdropping	Encrypt network traffic (TLS)
Authentication	Protect against forgery and masquerade	Server or client digital certificates
Message Integrity	Detect alteration	Digital signatures

- Without encrypting the network channel, data is passed in clear text across the network, so sensitive data could be viewed by anyone
- PKI allows a third-party trust relationship to be established over a public network

How?

- TLS handshake uses both Symmetric and Asymmetric encryption, in brief:
 - Initial contact uses Asymmetric encryption to establish a connection between two end-points. Client checks the server certificate to ensure they trust the CA who provided it. Server may (or not) ask for client certificate to validate
 - Generate and exchange a unique shared key for this session
 - Switch to Symmetric encryption for subsequent data exchange, using the shared key

IMPORTANT NOTE: The Private Key **MUST** be guarded carefully. If compromised, anyone can impersonate the Public Key subject

	Symmetric Encryption	Asymmetric (one-way)
Type of key	Single shared key	Key pair (Public/Private)
Key Exchange	Out of band	In band (over network)
Encryption/Decryption Speed	Fast	Slow
Use for	Bulk Encryption	Small blocks of data, digital signatures, digital certificates

- IETF <https://datatracker.ietf.org/doc/html/rfc5246>
- Wikipedia https://en.wikipedia.org/wiki/Transport_Layer_Security#TLS_handshake

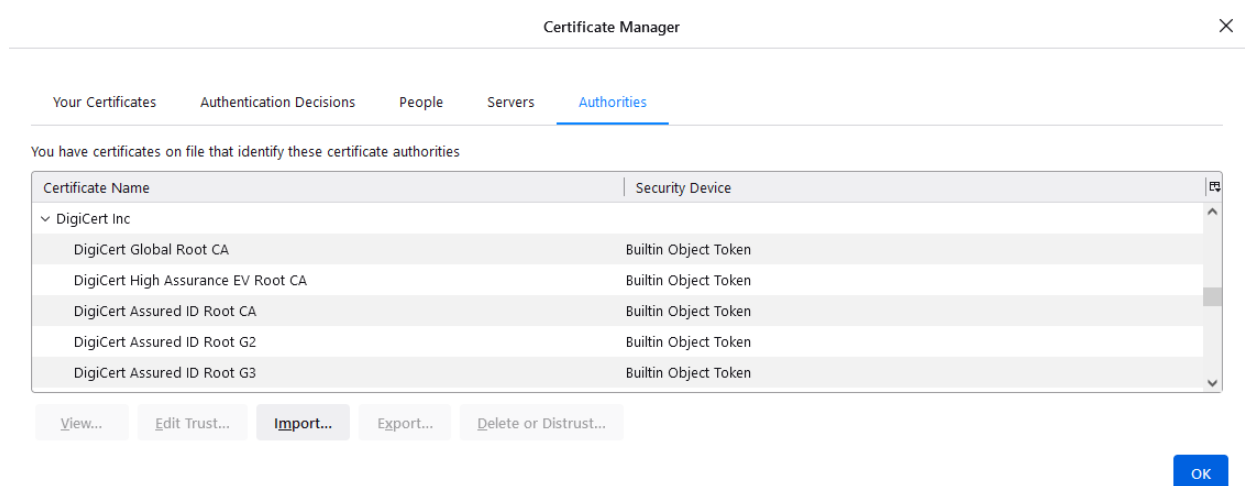
Certificate Authority



- A Certificate Authority (CA) is an "*Organisation that issues certificates and vouches for the identity of the subject*"
 - There are a variety of commercial organisations who you can pay to provide certificates, such as Verisign, Comodo, Digicert and a whole lot of others.
 - If you are using Microsoft AD (Other software providers are available), you can use the Microsoft Server CA to issue and manage certificates throughout your own organization.
 - You can generate your own certificates using free tools such as OpenSSL (other providers are available) which allow you to generate various types of digital certificates
 - Sage provides the script "certgen.bat" to generate certificates for Syracuse (Located in "..\syracuse\certs_tools")
 - The CA, in whatever form it takes, (External organisation, internal software, or a script stored on a local disk) is **implicitly trusted** by anyone using the public key of the CA
 - In physical terms, a CA is just a self-signed certificate

Trusting a Certificate Authority

- How do applications, such as browsers, recognize a CA as being "trusted"
- Each application maintains a certificate store, which is normally in a file or stored in a database. Usually this will be secured somehow, as it is vitally important to maintain a secure certificate store to ensure it cannot be updated by unauthorized people or software
- Browsers will ship with a variety of CA's already stored in its certificate store. These CAs will occasionally be maintained with browser updates
 - In Firefox for example, you can navigate to Tools, Settings, Security, View Certificates to see the "Authorities" that are considered trusted. Google Chrome currently uses the Microsoft Windows certificate store.



DEMO



- Show Firefox trusted CAs
- Review Server and CA certificates

Sage X3

- When you install or upgrade MongoDB, you can choose to accept only encrypted connections (recommended and defaulted) which sets the appropriate “tls” parameters in the mongod.conf



Service configuration

- ☒ The server uses and accepts only SSL encrypted connections.
☐ Redo the ssl configuration

Certificate Authority (CA) setup

Passphrases can contain all alphanumeric and nonalphanumeric characters except : ' and ".

Passphrase of CA:
Verification:

Certificate data

(*) All fields are mandatory.

Country code:
State/Province:
City/Locality:
Organization:
Organizational unit/Team:
Name/Certificate Owner:
Email:
Days of validity:

Mongodb server setup

Passphrases can contain all alphanumeric and nonalphanumeric characters except : ' and ".

(*) All fields are mandatory.

Please note : in order to start as a service the passphrase will be in clear text in the configuration file of this MongoDB server !

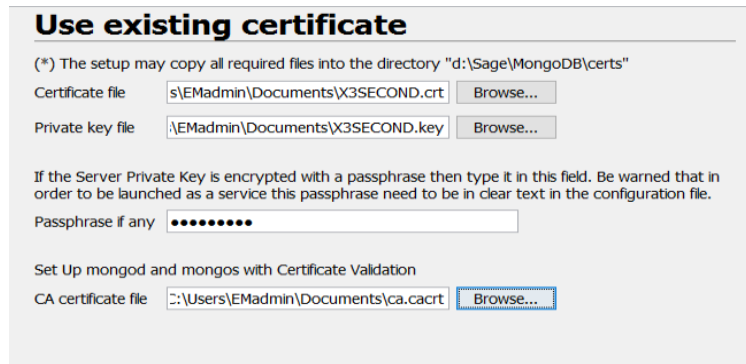
Passphrase of server:
Verification:
Host name (FQDN):

- MongoDB will then only accept client connections which use a certificate generated using the CA certificate used for the MongoDB installation. The Sage installer automatically generates a client certificate in “..\MongoDB\certs” directory which can be used by Syracuse, or other client applications such as the mongo shell

MongoDB clusters

MongoDB clusters allow you to have multiple instances of the MongoDB data and service on separate servers. This provides high availability for your environment.

- Each MongoDB server needs to use server certificates generated from the same CA
 - You therefore **cannot** use the certificate generated by the Sage installer for MongoDB. Instead, you need to obtain/create your own certificates** before installing MongoDB, then select “Use existing certificate” option in the installer for all MongoDB cluster nodes



The screenshot shows a dialog box titled "Use existing certificate". It contains the following fields and controls:

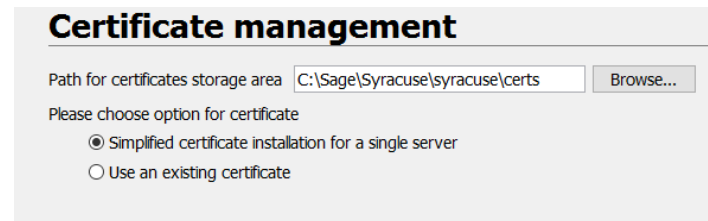
- A note: "(*) The setup may copy all required files into the directory "d:\Sage\MongoDB\certs"
- "Certificate file" field with the path "s:\EMadmin\Documents\X3SECOND.crt" and a "Browse..." button.
- "Private key file" field with the path "\EMadmin\Documents\X3SECOND.key" and a "Browse..." button.
- A text instruction: "If the Server Private Key is encrypted with a passphrase then type it in this field. Be warned that in order to be launched as a service this passphrase need to be in clear text in the configuration file."
- "Passphrase if any" field with a masked input (dots).
- A checkbox labeled "Set Up mongod and mongos with Certificate Validation".
- "CA certificate file" field with the path "C:\Users\EMadmin\Documents\ca.cacrt" and a "Browse..." button.

** Sage do not provide a way to do this currently (it is under consideration). Meanwhile, I have an OpenSSL script available if needed. Available via GitHub (see Investigation Scripts presentation for information)

- Review mongod.conf
- Show CA, server and client certificates
- Launch mongo shell, with and without client certificate

If you are installing only one Syracuse node, you can use the the default "Simplified certificate installation for a single server"

- This generates the Syracuse “internal“ CA and Server certificates and installs them for you



Certificate management

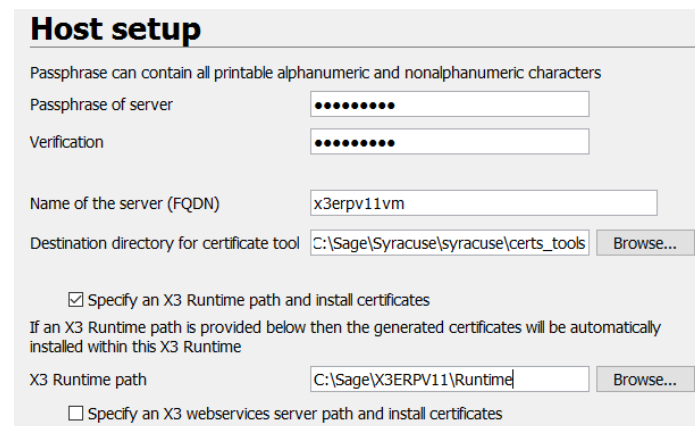
Path for certificates storage area

Please choose option for certificate

☒ Simplified certificate installation for a single server

☐ Use an existing certificate

- If you are running a Single-Node install, you can also automate the copy of the required certificate in the X3 Runtime “keys” directory (otherwise you need to manually copy the public key file)



Host setup

Passphrase can contain all printable alphanumeric and nonalphanumeric characters

Passphrase of server

Verification

Name of the server (FQDN)

Destination directory for certificate tool

☒ Specify an X3 Runtime path and install certificates

If an X3 Runtime path is provided below then the generated certificates will be automatically installed within this X3 Runtime

X3 Runtime path

☐ Specify an X3 webservices server path and install certificates

For the MongoDB connectivity where MongoDB is using SSL, you need to remember to check “The server uses and accepts only SSL encrypted connections” box (Default is unchecked) and locate the appropriate MongoDB certificates

Mongodb connection

Please fill in the parameters of the url to connect to MongoDB

Server name (FQDN) or a member of replica set

Service port number

☒ The server uses and accepts only SSL encrypted connections.

(*) The setup may copy all required files into the directory "D:\Sage\Syracuse\syracuse\certs"

Client certificate file (*.crt)	<input type="text" value="D:\Sage\MongoDB\certs\client.crt"/>	<input type="button" value="Browse..."/>
Client private key file (*.key)	<input type="text" value="D:\Sage\MongoDB\certs\client.key"/>	<input type="button" value="Browse..."/>
CA certificate file	<input type="text" value="D:\Sage\MongoDB\certs\ca.cacrt"/>	<input type="button" value="Browse..."/>

Syracuse Cluster



A Syracuse cluster is automatically created when you install 2 or more Syracuse servers, using the same MongoDB database

There is already a Blog article "[Illustrated guide to installing additional Syracuse nodes](#)" which describes the steps in some detail

In brief, from the point of view of the certificates only:

- You can install the first Syracuse server using "Simplified certificate installation for a single server"
- Create Syracuse certificates for the second and subsequent Syracuse nodes, using "certgen.bat" from the first Syracuse server
- Install second Syracuse server as normal, but select "Use an existing certificate"
 - You'll be connecting to the existing MongoDB, so will also need to have the appropriate Mongo certificates available

Syracuse Cluster



Errors with Syracuse clusters can occur due to:

- Different CAs have been used for the different server certificates
 - i.e. "Simplified certificate installation" was incorrectly used on multiple servers
- Firewall rules have not been updated to allow communication between the Syracuse servers
 - The first port specified in each server's "host" setup is used for inter-server communication

Syracuse to X3 Runtime communication



When a connection request is sent from Syracuse to a Sage X3 "classic" server (i.e. the X3 Runtime server) encrypted data is sent using the Syracuse private key. The Sage X3 classic server will identify the Syracuse server that connects to it by using the Syracuse public key to decrypt the connection request.

- The Syracuse certificate generation creates a public key file for the server in the “..\certs_tools\output” directory (e.g. “myServer.pem”) This file is transferred or copied to the X3 Runtime server “keys” directory (e.g. “D:\Sage\X3ERPv12\Runtime\keys”)
 - If you forget, you will see the following error when you login and try to access the Runtime

A screenshot of a dark-themed error message box. It features a red exclamation mark icon on the left, followed by the text "Create session error : X3 engine error: Unable to find x3erpv11web.pem".

! Create session error : X3 engine error: Unable to find x3erpv11web.pem

What are Internal certificates?



Both the CA and Server certificates used by the Syracuse installer are flagged as "INTERNAL" within Sage X3 and cannot be edited or deleted from within Sage X3. These certificates are required for Sage X3 to function and are not intended to be used other than for internal use. They have a 10-year life by default.

- If you need to regenerate these “internal” certificates for any reason:
 - Use “certgen.bat” to generate new certificates
 - Overwrite the previous files by copying the new certificate files into the “..\Syracuse\certs\MyHostName” directory
 - Restart Syracuse
 - You may need to manually reset the passphrase if you get errors such as “Passphrase cannot be decrypted” in which case you can do the following steps:
 - Launch Windows command prompt using “Run as” for the service user account
 - Change directory to the “..\syracuse\” directory
 - Run “**passphrase.cmd MyPassphrase**” where *MyPassphrase* is the passphrase you used for the certificate generation

DEMO



- Check the Syracuse certificates in “..\Syracuse\certs” and “..\Runtime\keys”
- Look at the certificates through the front end, notice these are flagged “internal”

HTTPS connectivity for client connections



Described in [How to implement SSL Certificates with Sage X3 Syracuse web server](#)

- Create certificate request for each server
 - You could also use a “Wildcard” certificate if needed for multiple servers in the same domain
- Submit certificate request to your CA
- Install CA and Server certificates received from your CA into Sage X3
- Update Syracuse host to use SSL

Port	Active	SSL	Client authentication	Server certificate	Client certificate
8124	✓	✗	✗		
444	✓	✓	✗	x3erov12vm	!

- NOTE: for testing purposes, you could use the “internal” certificate for client SSL connections

DEMO



- Generate certificate with Intermediate CA
- Load certificates
- Setup SSL port
- Discuss browser errors and/or install CA as trusted in browser

Final Thoughts



Don't forget that your customer certificates may need to be regenerated sometime in the future, so:

- Document how you created the original certificates
 - e.g. the location of any scripts
- Document the passphrase(s) used

Protect all private key files and passphrases as much as can be sensibly achieved. If these are compromised, the existing certificates are literally worse than useless

Appendices

Appendix A: When certificates expire



Server certificates are normally valid for one or two years, so what happens when the server certificate expires?

- Create new certificate request (Using existing private key)
- Submit certificate request to your CA
- Install Server certificate received from your CA into Sage X3
 - You can either load the new certificate into the existing record, or create a new entry
 - The CA certificates themselves are unlikely to need changing
- Update Syracuse host to use new entry (if you created a new entry)
 - This option may be safer/easier to implement (and revert if needed)
- See “How to implement SSL Certificates with Sage X3 Syracuse web server”
<https://support.na.sage.com/selfservice/viewdocument.do?externalId=100089>

Appendix B: Client authentication with certificates



For additional verification you can configure Client authentication to also use certificates. There are two configuration options:

1. User needs to provide login and password as normal, but additionally must present a client certificate signed by a CA the server knows about (Can be the same certificate for all users)
2. The Common Name in the client certificate is taken as the user's X3 login name, therefore not prompting the user for username/password (One certificate per user, installed in their PC/Laptop's browser)

The basic steps to configure are:

- Configure X3 Server for client authentication
- Create a new client certificate, which is now required for a user to connect
- Install the client certificate in the client browser, which should then allow login as normal (using a username/password)
- Optionally allow sign in by presenting a client certificate with X3 username

Appendix C: Further Reading - General

- **General information**
 - SSL/TLS Overview <https://sites.google.com/site/tlsssloverview/home>
 - TLS/SSL Technical Reference [https://technet.microsoft.com/en-us/library/cc784450\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc784450(v=ws.10).aspx)
 - SSL and TLS: A Beginners Guide <https://www.sans.org/white-papers/1029/>
 - Wikipedia (https://en.wikipedia.org/wiki/Transport_Layer_Security)

Appendix C: Further Reading – Sage specific



- **Sage Documents**

- How to implement SSL Certificates with Sage X3 Syracuse web server <https://support.na.sage.com/selfservice/viewdocument.do?externalId=100089>
- Configuring Syracuse for MongoDB X509 Authentication <https://online-help.sageerpx3.com/erp/12/staticpost/configuring-syracuse-for-mongodb-x509-authentication>
- Illustrated guide to installing additional Syracuse nodes <https://www.sagecity.com/gb/sage-x3-uk/b/sage-x3-uk-support-insights/posts/illustrated-guide-to-installing-additional-syracuse-nodes>
- Certificate Installation <https://online-help.sageerpx3.com/erp/12/staticpost/certificate-installation/>

Thank you

