Creating an authorized TLS certificate

XLeap Server

Content

XLeap servers require a valid authorized TLS certificate by which its XLeap Center is identified, and traffic is encrypted. This document provides a step-by-step guide for creating such a certificate with Java 'keytool'.

For your XLeap Center, a single domain SSL certificate will do.

Technically, all current SSL certificates are TLS certificates. That said, most people and, indeed, the keytool program referenced below, talk of SSL when they mean TLS.

XLeap defaults to TLS 1.3 encryption of traffic.

---

1. The keytool program

The XLeap server reads the SSL certificate from a password protected keystore, which can be generated with a Java program called 'keytool'. This is a command line (non-graphical) program.

You can run 'keytool' on any computer where a Java runtime environment (version 8 or higher) is installed correctly with path variables

- On Linux
  $JAVA_HOME/bin
- On Windows
  %JAVA_HOME%\bin

If required, change directory to the java directory which contains program 'keytool'.

If you have installed the XLeap Server

- on a Windows Server server, you can use the Java runtime on that machine.
- on a Linux server, you find 'keytool' in directory

  ```
  /srv/meetingsphere/java/bin/
  ```

  To run keytool from any directory on your Linux server you can add this to you path

  ```
  export PATH=/srv/meetingsphere/java/bin/:$PATH
  ```

2. Create a private key-store

2.1 Call 'keytool'

Execute the following command from the command-line prompt:

```
keytool -genkey -alias tomcat -keyalg RSA -keysize 2048
-keystore domainname.kdb
```

for "domainname.kdb" substitute your domain name. In the example below this is "example.com.kdb".

```
bash ~ # keytool -genkey -alias tomcat -keyalg RSA -keysize 2048 -keystore example.com.kdb
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  example.com
What is the name of your organizational unit?
  [Unknown]:  Meeting Management
What is the name of your organization?
  [Unknown]:  Example Inc.
What is the name of your City or Locality?
  [Unknown]:  Austin
What is the name of your State or Province?
  [Unknown]:  Texas
What is the two-letter country code for this unit?
  [Unknown]:  US
Is CN=example.com, OU=Meeting Management, O=Example Inc., L=Austin, ST=Texas, C=US correct?
  [no]:  yes

Enter key password for <tomcat>
        (RETURN if same as keystore password):

bash ~ #
```

bash command prompt: creating a keystore file

2.2 Keystore password

When prompted, specify the password for your keystore. Write this down!

You require this password when deploying the keystore in the Server console.

There may be no feedback on specification of your password.

2.3 Domain name (or sub domain)

When asked “What is your first and last name?” specify the (sub)domain name by which your XLeap Center is registered in the DNS.

For a single domain certificate you have to specify the full domain name (here “example.com”) If, however, the hostname registered in the DNS results in a URL like https://xleap.example.com, specify “xleap.example.com” (everything after ”https://”). In this case, use “xleap.example.com” also as the file name of your keystore!

Any discrepancy between (sub) domain name in the keystore and the actual address of your XLeap Center will cause security alerts in the users’ browsers!

Any combination of characters to the left of the domain name and separated by a “dot” constitutes a sub domain: “www.example.com” is a sub domain of “example.com” and is not covered by a single-domain certificate for “example.com” (some certificate authorities however, may include it for you).

Wildcard certificates such as “.example.com” cover all sub domains of example.com.*

2.4 Organizational unit / Organization

Specify the name of your department and the complete legal name of your organization. In the example this is “Meeting Management” and “Example Inc”. You may specify your organization name also for “Organizational unit”.

Do not use characters **[! @ # $ % ^ ( ) ~ ? > < & / \ , . " ']** as they are not supported.

2.5 City, state, and country

In the example given above these are “Austin” (city), “Texas” (state) and “US” (country code). The country is specified with its “2-letter country code” according to ISO 3166-1 alpha 2 which is also used by e.g. NATO. Examples: US, DE, GB, FR, ES, JP.

2.6 Verify your specification

keytool will display your specification for confirmation.

If correct, confirm with “`yes`“.

2.7 Password for <tomcat>

keytool prompts you again for a password. You do not need to type the password again and can simple press “enter” to confirm the password given above.

Depending on you Java version, you may get a proprietary format warning and a recommendation to migrate to a different format.

You can disregard such warning.

2.8 Creation and backup of the keystore file

On confirmation of the password for <tomcat>, the specified keystore file (in the example “example.com.kdb“) will be created and stored in the directory from which “keytool” was called.

Create a backup of the keystore file.

3. Certificate signing request (CSR)

From the command prompt, call “keytool“:

```
keytool -certreq -alias tomcat -keystore domainname.kdb
-file domainname.csr
```

Substitute the file name you have specified in step 1 above (e.g. “example.com.kdb”) for “domainname.kdb“. Use that name also for the signing-request file. In our example “domain-name.csr” should read “example.com.csr”.

When prompted, give the password of the keystore (here: “changeit”).

```
bash ~ # keytool -certreq -alias tomcat -keystore example.com.kdb -file example.com.csr
Enter keystore password:

bash ~ #
```

bash command line prompt: certificate signing request

Depending on you Java version, you may get a proprietary format warning and a recommendation to migrate to a different format.

You can disregard such warning.

Create another backup of the keystore, as a ‘Certificate signing request’ (this step) will produce different results if repeated.

You can open the certificate signing request (in this example example.com.csr) in a text editor or display it on the command line it will be required in the next step.

```
bash ~ # cat example.com.csr
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIC7TCCAdUCAQAweDELMAkGA1UEBhMCVVMxDjAMBgNVBAgTBVRleGFzMQ8wDQYD
VQQHEwZBdXN0aW4xFTATBgNVBAoTDEV4YW1wbGUgSW5jLjEbMBkGA1UECxMSTWV1
dGluZyBNYW5hZ2VtZW50MRQwEgYDVQQDEwtleGFtcGxlLmNvbTCCASIwDQYJKoZI
hvcNAQEBBQADggEPADCCAQoCggEBAJz9r4rGqExWuRSzZ786WpKCC2pP8uoUo3NW
JAXLEG10fVlvKVUqdqwiV4GKKD2ZWf2trLtLj/kEIyNN+KgfNviW+TN4QxgTrqVE
YmHqkT81rRnYqwZx05GcXYGWqRramsncvLw+p2veas3ITSBZVIkCLku6ulJDSpXA
gXMtejACuB/fIgxb+GccnqSi3YQZLCiG9UpCIiFrL8+rTiNjtFLswmJL4yXfM2mf
G/PQHvp5XdwiMbiJq6XLrS+QDllNLmj49l4vXcupWiGKvtguY6xJUisPeNIyLrYm
EWNNPXacHk2QvuklrMMyfNlo+GOjfG0Pp7/kx9PJSutOnvwuieMCAwEAAaAwMC4G
CSqGSIb3DQEJDjEhMB8wHQYDVR0OBBYEFPaB8IM2RA9aalBHAVd0MSy2CkWDMA0G
CSqGSIb3DQEBCwUAA4IBAQBusG+FkZ++sKvE/ZSYdMZbzHfQSs3mqhF/kWLL+zlY
yzl8BeBmMTt+rFPCoiP2NXB2E7fxzu2muUnW9QMinPdDy5lbje+TvEEqhlOeB9w0
o+FlX02JdjQcxRACThCEih2FbGkWz2t5BYSAMEqvFWuuBLk0womqlay5m9GSQbpw
QLOk/lKywBCpO4NPImPwNjxjxlWT0AFgVGKmGkL9+j7fq9J5//gDBOMloeEfohG3
ZN46qTk30M///aAd/ME/UzeNoBfHGF2CGJH6VIcp+uyEylIh5748lLGmsZpBxPFc
ZDdfO/2cvopNd/OkvMA7cYko7GanFvgZS9CjnY7oZAVz
-----END NEW CERTIFICATE REQUEST-----
bash ~ #
```

Content of a certificate signing request file

4. Buying the SSL certificate

With the certificate signing request you can go to the homepage of your SSL provider (certificate authority). Follow instructions for creating an SSL certificate. Typically, you will be asked to paste the contents of the certificate signing request into a text field (see above)

At the end of this procedure you will receive (by download or email) a certificate for your domain and one or more SSL intermediate certificates of the certificate authority.

5. Finalize the keystore

To finalize the keystore, you must copy the received certificates to the directory from which you have called “keytool” and where the keystore (in our example “example.com.kdb“) and the signing request (in our example “example.com.csr“) reside.

Before you continue please create a backup of your current keystore file, e.g. example.com.kdb.bak. In case of mistakes you can start over.

The following description presupposes two certificates i.e. the “intermediate” certificate and the “domain” certificate. Should your SSL provider supply more than two certificates, follow the directions given by your SSL provider. However, make sure that the “alias” given for your certificate in steps 1 to 3 above is “**tomcat**”.

5.1 Import the intermediate certificate into the keystore

Call “keytool” again:

```
keytool -import -trustcacerts -keystore example.com.kdb
-alias intermediate -file IntermediateCA.crt
```

where

- “domainname.kdb“ is your kdb file (in our example “example.com.kdb“)
- “IntermediateCA.crt“ is an intermediate certificate of your certificate authority

```
bash ~ # keytool -import -trustcacerts -keystore example.com.kdb -alias intermediate -file Intermed
iateCA.crt
Enter keystore password:
Certificate was added to keystore

bash ~ #
```

Importing an intermediate certificate into a keystore

If you have received multiple intermediate certificates you must import them with an alias e.g. “intermediate1”, “intermediate2”and so on.

When prompted, give the password as specified in step 1.

Depending on you Java version, you may get a proprietary format warning and a recommendation to migrate to a different format.

You can disregard such warning.

5.2 Import the SSL certificate into the keystore

Call 'keytool' again:

```
keytool -import -trustcacerts -keystore example.com.kdb
-alias tomcat -file ssl_certificate.crt
```

Where

- "domainname.kdb" is your keystore file (in our example "example.com.kdb")
- „ssl_certificate.crt" is the domain certificate you received from your SSL provider.

The alias for your certificate must be "tomcat" as this was used in step 3.

When prompted, give your password.

```
bash ~ # keytool -import -trustcacerts -keystore example.com.kdb -alias tomcat -file ssl_certificat
e.crt
Enter keystore password:
Certificate reply was installed in keystore

bash ~ #
```

Keytool: Import of the domain certificate into the keystore

Your keystore domainname.kdb (in the example: "example.com.kdb") is now complete and ready for use. Create a backup and store of this file in a safe place!

Depending on you Java version, you may get a proprietary format warning and a recommendation to migrate to a different format.

You can disregard such warning.

6. Upload the keystore in the XLeap Server console

Open the XLeap Server console > Server administration > SSL keystore control.

- Specify "uploaded keystore".
- Upload the keystore and specify the password