CIPHERMAIL EMAIL ENCRYPTION

Ciphermail Gateway EJBCA integration guide

April 4, 2016, Rev: 5460

Contents

1 Introduction **3**

2 Configure Ciphermail **3**

3 Configure EJBCA **5**
3.1 Certificate Profile 5
3.2 End Entity Profile 5

4 Finish **6**

A EJBCA certificate request handler configuration **8**

1 Introduction

This guide explains how to configure a Ciphermail gateway to make the gateway request certificates from an external EJBCA server. EJBCA (http://ejbca.org/) is a widely used, flexible, enterprise Java based open source CA server. Ciphermail contains basic CA functionality. If however more advanced CA functionality is required, for example add constraints to the issued certificates, EJBCA is advised.

Note: this guide assumes that a Ciphermail gateway and an EJBCA server have already been setup and that the EJBCA server is setup to accept incoming connections to the Web Service Interface.

2 Configure Ciphermail

To configure Ciphermail for EJBCA support, the *EJBCA certificate request handler* should be configured. All the default certificate request handlers are defined in the spring configuration file found in the folder where Ciphermail is installed[1]:

`conf/spring/certificate-request-handlers.xml`.

The *eJBCACertificateRequestHandlerSettings* bean specifies the EJBCA specific settings and should be configured (see Appendix A for the default configuration).

The following properties can be set:

enabled Set to true if the EJBCA certificate request handler should be enabled (it's disabled by default).

webServiceURL The *webServiceURL* is the URL of the EJBCA server web service. The URL should typically be of the form:

`https://HOST:8443/ejbca/ejbcaws/ejbcaws`

where *HOST* should be replaced by the real host name or the IP address of the EJBCA server[2].

keyStoreFile EJBCA requires that the web service client is authenticated with a client side certificate. The *keyStoreFile* should be set to the path to the file containing the administration certificate.

[1] For the Virtual Appliance and the Deb and RPM packages, the default folder is /usr/share/djigzo

[2] Since EJBCA requires client side authentication, the webServiceURL should always be accessed over HTTPS.

Note: With a default EJBCA installation, the administration certificate is stored in `EJBCA_HOME/p12/superadmin.p12`. The file `superadmin.p12` should be copied to a location accessible by Ciphermail.

keyStorePassword The password of the *keyStoreFile*.

Note: With a default EJBCA installation, the password for the `superadmin.p12` file is `ejbca`

keyStoreType The type of the *keyStoreFile* (PKCS12, JKS or some other key store type supported by Java). If *keyStoreType* is not set, the key store type is based on the filename extension of the key store file[3].

skipCertificateCheck The web service connection to the EJBCA server is protected with TLS/SSL. By default the connection to the EJBCA server is only setup if the SSL certificate used by the EJBCA server is trusted by the Ciphermail server (see *trustStoreFile*). To disable checking the web service SSL certificate, set *skipCertificateCheck* to true.

Note: Only set *skipCertificateCheck* to true for testing purposes or when the SSL web service certificate cannot be imported for some reason and the connection between the Ciphermail gateway and EJBCA server is fully trusted.

trustStoreFile The path to the key store with the trusted issuer of the web service SSL certificate. If not set, the default Java system trust store will be used. To completely skip all server certificate checks, set *skipCertificateCheck* to true.

Note: In a default EJBCA installation, the trusted certificate is stored in `EJBCA_HOME/p12/truststore.jks`. The file `truststore.jks` should be copied to a location accessible by Ciphermail.

trustStorePassword The password of the *trustStoreFile*.

Note: In a default EJBCA installation, the password for the `truststore.jks` file is `changeit`

trustStoreType The type of the *trustStoreFile*. See *keyStoreType* for more information.

[3].pfx and .p12 are opened as PKCS12 and jks and a file without an extension is opened as JKS

disableCNCheck By default, when the TLS/SSL connection is setup, a check is done to see whether the SSL certificate of the EJBCA server is issued to the hostname (or IP) address of the *webServiceURL*. If the hostname from the SSL certificate does not match the hostname of the *webServiceURL*, the TLS/SSL connection is terminated. If the certificate is issued to a different hostname than the hostname of the *webServiceURL*, the hostname check can be disabled by setting *disableCNCheck* to true.

CAName This should be set to the name of the *Certificate Authority* (CA) used for the issued certificates.

endEntityProfileName This should be set to the name of the *End Entity Profile* used for the issued certificates.

certificateProfileName This should be set to the name of the *Certificate Profile* used for the issued certificates.

defaultUserPassword By default EJBCA will generate a password for every new user. If for some reason a static password must be used, set the *defaultUserPassword* to the password selected for new users.

3 Configure EJBCA

To issue S/MIME certificates, certain *Certificate Profile* and *End Entity Profile* settings must be specified.

3.1 Certificate Profile

The default (fixed) *ENDUSER* certificate profile can be used since it contains all the relevant settings. If another certificate profile should be used instead of the *ENDUSER* profile, the following profile settings are required:

Key Usage If the key usage extension is checked, either no key usages should be selected (which implies that the key can be used for all purposes) or *Digital Signature* and *Key encipherment* should be selected[4].

Extended Key Usage If the extended key usage extension is checked, make sure that either *Any Extended Key Usage* or *Email Protection* is checked.

3.2 End Entity Profile

Subject DN Attributes Certain *Subject DN Attributes* should be added to the *End Entity Profile*: *Organization*, *givenName*, *surname* and *emailAddress*.

[4] *Digital Signature* is required for a certificate valid for digital signing and *Key encipherment* is required for a certificate valid for encryption.

Other subject attributes The *RFC 822 Name* should be added to *Other subject attributes* and *Use entity e-mail field* should be enabled.

See figure 1 for an example of the relevant *End Entity Profile* settings.

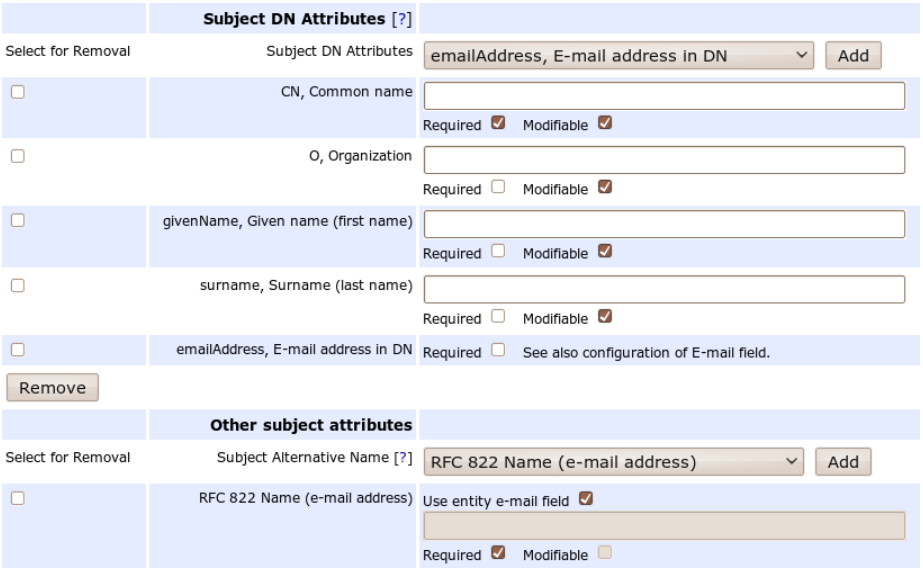

Figure 1: EJBCA End Entity Profile

4 Finish

After the EJBCA settings have been changed, the Ciphermail gateway should be restarted. After a restart, the administrator can select the *EJBCA* certificate request handler (see figure 2).

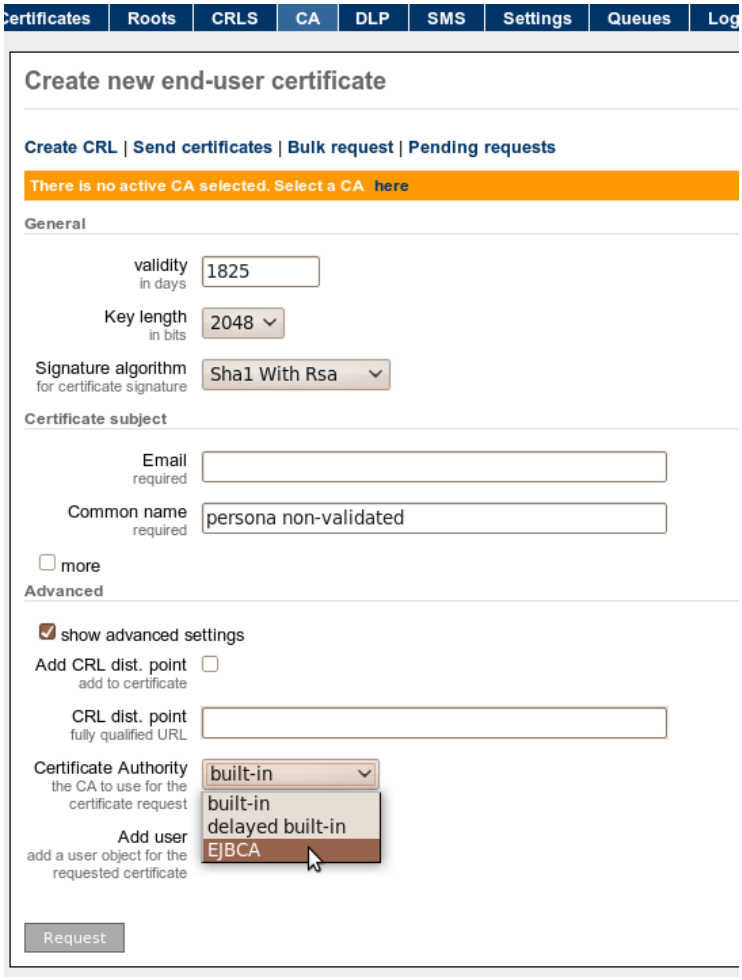

Figure 2: EJBCA certificate request handler

A EJBCA certificate request handler configuration

```
<!-- EJBCA certificate request handler settings -->
<bean id="eJBCACertificateRequestHandlerSettings"
    class="mitm.common.security.ca.handlers.ejbca.StaticEJBCACertificateRequestHandlerSettings">

    <!-- set enabled to true to use the EJBCA request handler  -->
    <property name="enabled" value="false" />

    <!-- The URL of the EJBA web service -->
    <property name="webServiceURL" value="https://192.168.178.113:8443/ejbca/ejbcaws/ejbcaws" />

    <!-- The path to the Admin certificate key store (.p12 or .pfx) -->
    <property name="keyStoreFile" value="/home/martijn/temp/superadmin.p12"/>

    <!-- The password of the Admin certificate key store -->
    <property name="keyStorePassword" value="ejbca"/>

    <!-- The type of the key store file (JKS, PKCS12 etc.). Leave empty to 'guess' type based on file extension -->
    <!--
    <property name="keyStoreType" value="PKCS12"/>
     -->

    <!-- If true, no server certificate trust check will be done (i.e., all certificates are accepted)-->
    <property name="skipCertificateCheck" value="false" />

    <!--
        The path the key store with the trusted issuer(s). This should be the store containing the root of the
        EJBCA server certificate. If not set, the default Java system trust store will be used. To completely
        skip all server certificate checks, set skipCertificateCheck to true.
    -->
    <property name="trustStoreFile" value="/home/martijn/temp/truststore.jks"/>

    <!-- The password of the trust store -->
    <property name="trustStorePassword" value="changeit"/>

    <!-- The type of the trust store file (JKS, PKCS12 etc.). Leave empty to 'guess' type based on file extension -->
    <!--
    <property name="trustStoreType" value="JKS"/>
     -->

    <!-- If true, the CN of the certificate is not checked if it's equal to the domain name of the webServiceURL -->
    <property name="disableCNCheck" value="true" />

    <!-- The EJBCA CA to use -->
    <property name="CAName" value="AdminCA1" />

    <!-- The EJBCA end entity profile to use -->
    <property name="endEntityProfileName" value="test" />

    <!-- The EJBCA certificate profile to use -->
    <property name="certificateProfileName" value="ENDUSER" />

    <!--
        The default password for newly requested certificates.
        Note: Only set this if password is required and not automatically generated by EJBCA
    -->
    <!--
    <property name="defaultUserPassword" value="changethis!" />
     -->
</bean>
```