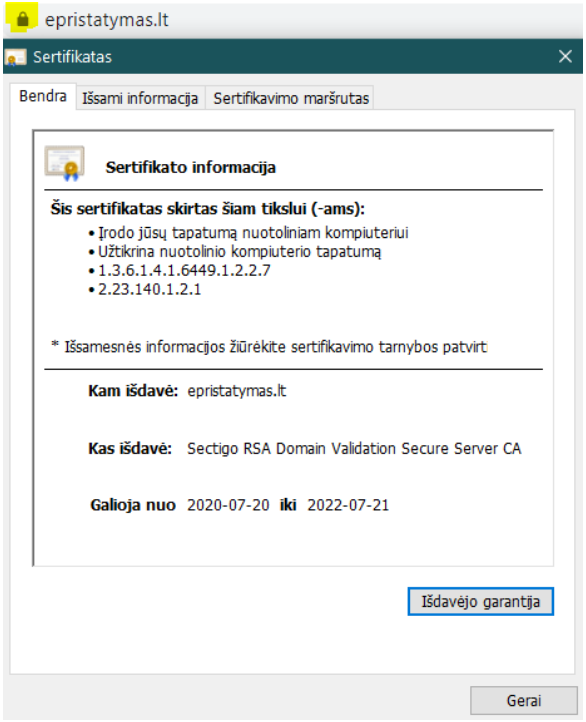
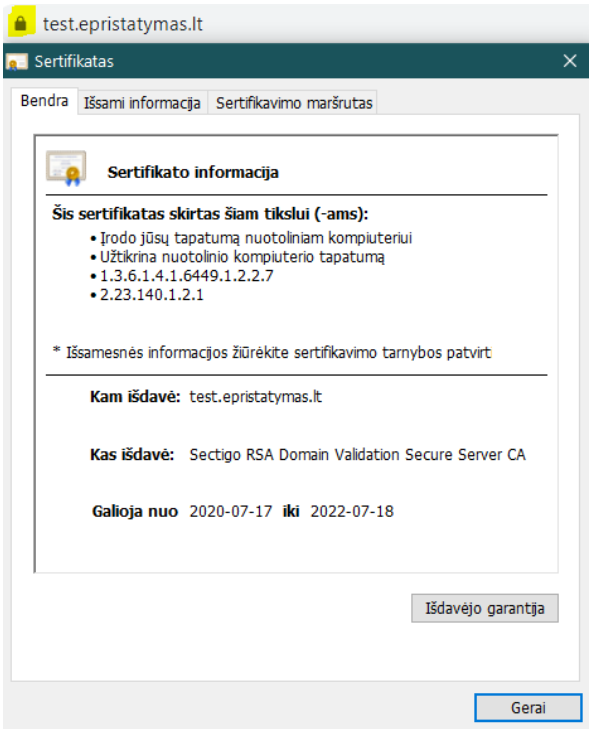
Naudinga informacija sistemų administratoriams

ePristatymas.lt naudojamas SSL sertifikatas:



test.ePristatymas.lt naudojamas SSL sertifikatas:



Jei pakeitus naudojamus adresus (URL) neveikia ePristatymo ir Jūsų sistemos integracinės sąsajos - siūlome patalpinti epristatymas.lt sertifikatus į Jūsų aplikacijų “keystore”.

Pvz. iš redhat:

Issue

SSL/TLS Connection fails and log has this error:

```
Raw

javax.net.ssl.SSLHandshakeException:sun.security.validator.ValidatorException: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested
target
```

Resolution

To resolve this issue, import the server's certificate into a truststore and then configure the client to use that truststore. Some connections must be trusted by both the client and server, also called 2-way SSL. In this case the server's truststore would alse need to contain one of the client's certificates. If the certificates are signed by an authority, you must import the authority's certificate into the trustStore.

The following command can be used to import the server's certificate into a trust store:

```
Raw

keytool -import -trustcacerts -alias server_cert -file server.cert -keystore client.truststore
```

The biggest challenge with this type of error will be to determine which keystore needs to have the certificate added to it.

The general order of precendence is:

1. The framework, application or component making the call may have a specific way to configure the keystore to be used as a trustStore. However, the specific configuration options vary by framework (Spring, Apache Axis, etc) used or the application itself.
2. If there is not a more specific method to configure the trustStore, then the default SSL context may be used. In that case a trustStore location can be set globally for Java's default SSL context via these system properties with notional values:

   ```
   Raw

   javax.net.ssl.trustStore=/path/to/client.truststore
   javax.net.ssl.trustStorePassword=wxyz1234
   ```

   - For standard java applications, you can pass these on the Java command line via "-Djavax.net.ssl.trustStore=...".
   - For JBoss EAP, see this solution: Add/remove/update system properties in JBoss EAP 6/7

> **Note** If you are not setting `javax.net.ssl.trustStore`, then Java's default SSL context will use `${java.home}/jre/lib/security/cacerts`, which means other side of the connection must present a certificate signed by one of the Root certificate authorities.

Root Cause

This exception means that the client-side application does not trust the SSL server that it is attempting to communicate with.

You can see this for any number of reasons but you generally see this message when your application or JBoss can not validate the server certificate of an endpoint. When connecting over SSL to a endpoint, the client validates that the server is considered trusted by comparing the server's certificate's and certificate chain (i.e. the certificates that signed the server certificate) and comparing to the certificates registered as trust anchors, or in a defined truststore.

At least one CA certificate in the presented certificate's chain must be in the trustStore for it to be considered trusted. For self-signed certificates, which is what you can generated on the command line, then it's fine to import the certificate itself into the trustStore, but if the certificate is signed by another authority, then importing the certificate itself won't work because the trust is based on the signing authority, not the certificate itself.