# ESET PROTECT Webconsole ( Apache Tomcat) - SSL Certificate from windows keystore

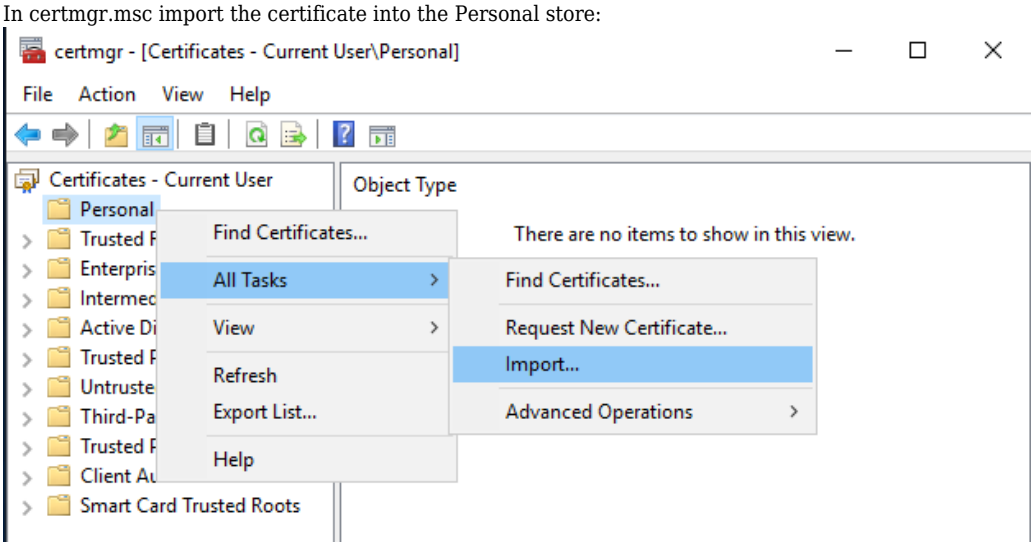
Mitchell | ESET Nederland - 2022-10-26 - Comments (0) - ESET PROTECT On-prem

## Pre-requisites:

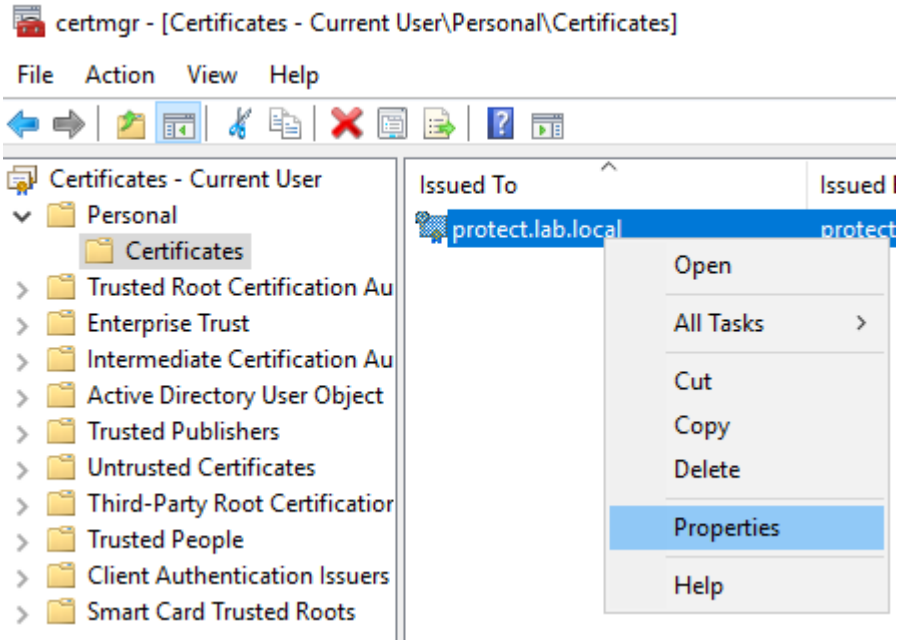
1. Certificate in p12 or pfx format (with private key)

## Configuring Apache Tomcat:

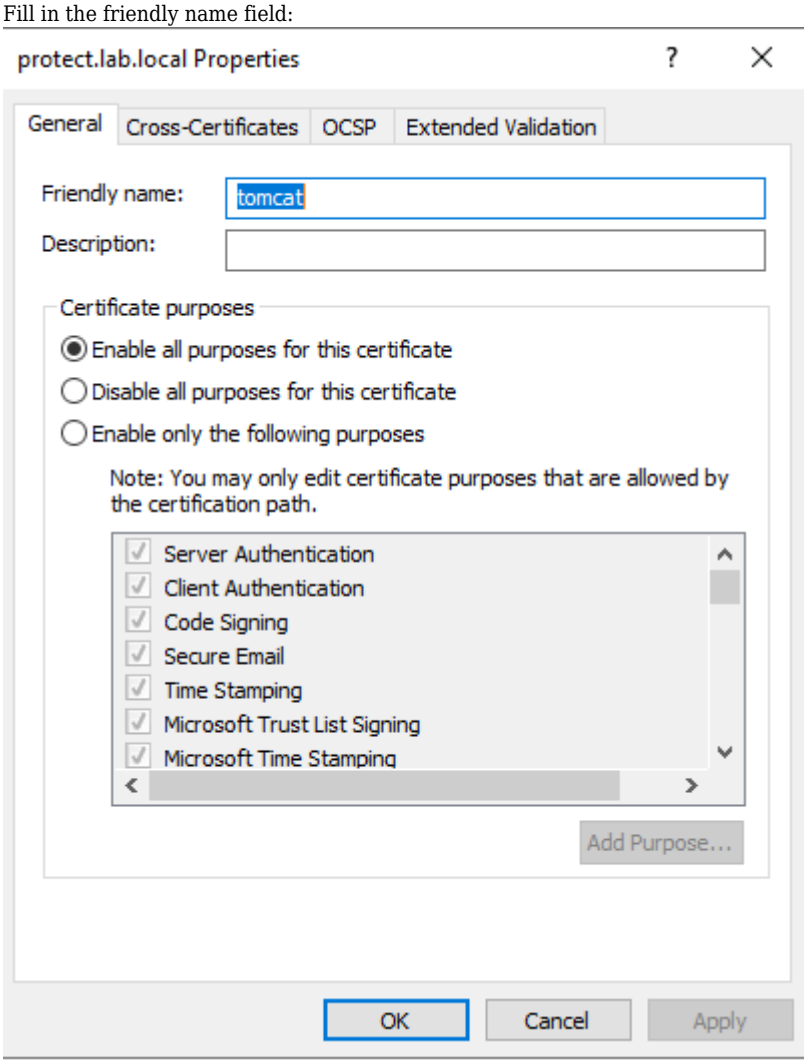
1. open powershell as the service account used by Apache Tomcat:
   runas /user:domain\sa-eset powershell.exe
2. Open certmgr.msc in the powershell session running as the service account.
   certmgr.msc
3. In certmgr.msc import the certificate into the Personal store:



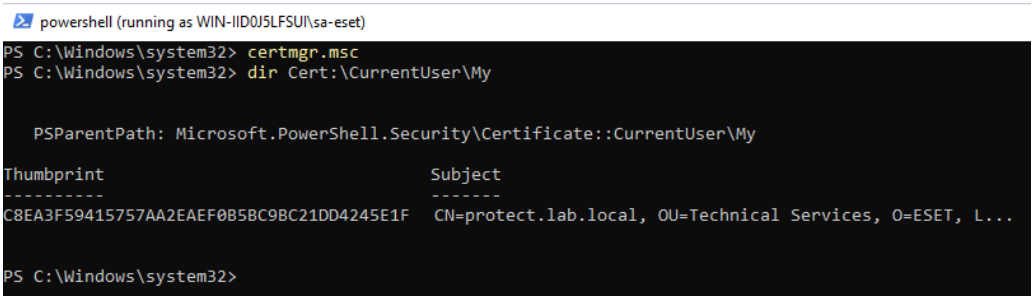

4. Note that the "friendly name" should contain a value in order for Apache Tomcat to use the certificate, if the friendly name value is empty, right click the certificate after importing and select properties:




Fill in the friendly name field:




5. Go back to the powershell session and verify the certificate exists:
dir Cert:\CurrentUser\My




6. Open the Apache Tomcat server.xml, by default located at: "C:\Program Files\Apache Software Foundation\apache-tomcat-9.0.64\conf\server.xml" (path may vary, based on installed version of Apache Tomcat)

   A. Find the following line:

```
<Connector server="OtherWebServer" port="443"
protocol="org.apache.coyote.http11.Http11NioProtocol" SSLEnabled="true"
maxThreads="150" scheme="https" secure="true" clientAuth="false"
sslProtocol="TLS" keystoreFile="C:\Program Files\Apache Software
Foundation\apache-tomcat-9.0.64\.keystore" keystorePass="********"
keyAlias="tomcat" sslEnabledProtocols="TLSv1.2,TLSv1.3" honorCipherOrder="true"
ciphers="TLS_AES_256_GCM_SHA384, TLS_CHACHA20_POLY1305_SHA256,
TLS_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256,
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256" />
```

   B. Change it to: (note that the keyAlias value should match the friendly name set in step 4.

```
<Connector server="OtherWebServer" port="443"
protocol="org.apache.coyote.http11.Http11NioProtocol" SSLEnabled="true"
maxThreads="150" scheme="https" secure="true" clientAuth="false"
sslProtocol="TLS" keystoreType="Windows-My" keystoreFile="" keystorePass=""
keyAlias="tomcat" sslEnabledProtocols="TLSv1.2,TLSv1.3" honorCipherOrder="true"
ciphers="TLS_AES_256_GCM_SHA384, TLS_CHACHA20_POLY1305_SHA256,
TLS_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256,
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256"
sslImplementationName="org.apache.tomcat.util.net.jsse.JSSEImplementation" />
```

7. Restart the Apache Tomcat Service, It should now use the certificate from the certificate store.