ESET Tech Center

Set up a HTTPS/SSL connection for ESET Security Management Center Web Console (7.x)

Anish | ESET Nederland - 2019-08-16 - Reacties (0) - ESET Security Management Center

Issue

- You receive the warning message **Using unencrypted connection! Please configure the webserver to use HTTPS** when accessing the ESET Security Management Center Web Console (ESMC Web Console) via HTTP.
  For security reasons, we recommend that you set up ESMC Web Console to use HTTPS.

Solution

Before you start

- The steps below refer to certificates for Apache Tomcat, which are used to ensure secure HTTPS connections. For information about ESET Security Management Center certifications, see our Online Help topic.
- The steps as described below are performed on a 64-bit Microsoft Windows Server operating system (with 64-bit Java and 64-bit Apache Tomcat installed). Some paths may vary depending on the operating system you are using.

Use an existing certificate

1. Move the certificate .pfx file to your Tomcat install directory (the folder name may vary - substitute "Tomcat_folder" with the actual folder name).

   ```
   C:\Program Files\Apache Software Foundation\Tomcat_folder\
   ```

2. Open the **conf** folder in the Tomcat install directory and locate the **Server.xml** file. Edit this file using a text editor (such as Notepad ++).

   1. If there is no <Connector after </Engine> in Server.xml (for example when you perform a new instalation of Apache Tomcat), copy the following string into the Server.xml after </Engine> (use you values for keystoreFile, keystorePass, and keystoreType):

      ```
      <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\Tomcat_folder\certificate_file.pfx"
keystorePass="Secret_Password_123"keystoreType="PKCS12" ssl
EnabledProtocols="TLSv1,TLSv1.1,TLSv1.2"
ciphers="TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
TLS_RSA_WITH_AES_128_CBC_SHA256,
TLS_RSA_WITH_AES_128_GCM_SHA256,
TLS_RSA_WITH_AES_128_CBC_SHA,
TLS_RSA_WITH_AES_256_CBC_SHA256,
TLS_RSA_WITH_AES_256_GCM_SHA384,
TLS_RSA_WITH_AES_256_CBC_SHA"/>
```

2. If `<Connector` is present after `</Engine>` in Server.xml (for example when you restore Server.xml after Apache Tomcat upgrade), replace the values of parameters listed below with your values:
   `keystoreFile` - Provide full path to the certificate file (`.pfx, .keystore`, or other).
   `keystorePass` - Provide certificate passphrase.
   `keystoreType` - Specify the certificate type.

Apache Tomcat documentation:

Read Apache Tomcat documentation for more information about the HTTP Connector.

3. Restart the **Tomcat** service.

Always use .pfx with password!

The `.pfx` certificate must **not** use blank password.

Create a new certificate and get it signed

To use a secure HTTPS/SSL connection https://www. for ESMC Web Console, follow the steps below:

1. Create a **keystore** with an **SSL certificate**. You must have **Java** installed.

Apache Tomcat requires Java:

Make sure that Java, ESMC, and Apache Tomcat have the same bitness (32-bit or 64-bit).

If you have multiple Java versions installed on your system, we recommend that you uninstall older Java versions and keep only the latest Java.

Starting January 2019, Oracle JAVA SE 8 public updates for business, commercial or production use will require a commercial license. If you do not purchase a JAVA SE subscription, you can use this guide to transition to a no-cost alternative.

Java includes the **keytool** (*keytool.exe*), which allows you to create a certificate via command line. You must generate a new certificate for each tomcat instance (if you have multiple tomcat instances) to ensure that if one certificate is compromised, other tomcat instances will remain secure.

Below is a sample command to create a keystore with an SSL certificate.

Navigate to the exact location of the **keytool.exe** file, for example `C:\Program Files\Java\jre1.8.0_201\bin` (the directory depends on the OS and Java version) and then run the command:

keytool.exe -genkeypair -alias "tomcat" -keyalg RSA -keysize 4096 -validity 3650 -keystore "C:\Program Files\Apache Software Foundation\`Tomcat_folder`\tomcat.keystore" -storepass "yourpassword" -keypass "yourpassword" -dname "CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown"

Are you a Linux user?

keytool -genkeypair -alias "tomcat" -keyalg RSA -keysize 4096 -validity 3650 -keystore "/etc/tomcat/tomcat.keystore" -storepass "yourpassword" -keypass "yourpassword" -dname "CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown"

The file path `/etc/tomcat/tomcat.keystore` is only an example, choose your own secure and accessible destination.

-storepass and -keypass parameters

Values for -storepass and -keypass must be the same.

2. Export the certificate from the keystore. Below is a sample command to export the certificate sign request from the keystore:

keytool.exe -certreq -alias tomcat -file "C:\Install\tomcat\tomcat.csr" -keystore "C:\Program Files\Apache Software Foundation\Tomcat_folder\tomcat.keystore" -ext san=dns:ESMC7-2008R2

Are you a Linux user?
keytool -certreq -alias tomcat -file "/etc/tomcat/tomcat.csr" -keystore "/etc/tomcat/tomcat.keystore" -ext san=dns:ESMC7-2008R2

Replace values appropriately

Replace the value "C:\Install\tomcat\tomcat.csr" for the -file parameter with the actual path and file name where you want the certificate to be exported.

Replace the value ESMC7-2008R2 for the -ext parameter with the actual hostname of the server on which your Apache Tomcat with ESMC Web Console is running.

3. Get the SSL certificate signed with the Root Certificate Authority (CA) of your choice.

You can proceed to step 6 if you plan to import a Root CA later. If you choose to proceed this way your web browser may display warnings about a self-signed certificate and you will need to add an exception to connect to ESMC Web Console via HTTPS.

4. Import the root certificate and intermediate certificate of your CA to your keystore. These certificates are usualy made available (on web page) by the entity who signed your certificate. It is necessary because the certificate reply is validated using trusted certificates from the keystore.

```
keytool.exe -import -alias root -keystore "C:\Program Files\Apache Software Foundation\Tomcat_folder\tomcat.keystore" -trustcacerts -file "C:\root.crt"
```

```
keytool.exe -import -alias intermediate -keystore "C:\Program Files\Apache Software Foundation\Tomcat_folder\tomcat.keystore" -trustcacerts -file "C:\intermediate.crt.pem"
```

5. Once you have received the signed certificate with the Root CA public key of CA and then certificate (`tomcat.cer`) into your keystore. Below is a sample command that imports a signed certificate into the keystore:

keytool.exe -import -alias tomcat -file "C:\Install\tomcat\tomcat.cer" -keystore "C:\Program Files\Apache Software Foundation\Tomcat_folder\tomcat.keystore"

Are you a Linux user?
keytool -importcert -alias tomcat -file "/etc/tomcat/tomcat.cer" -keystore "/etc/tomcat/tomcat.keystore"

Replace values appropriately

Replace the value `" C:\Install\tomcat\tomcat.cer "` for the `-file` parameter with the actual path and file name where the signed certificate is located.

If you want to use an already existing certificate (for example company certificate), follow these instructions.

6. Edit the `server.xml` configuration file so that the tag `<Connector` is written similar to the example below:

```
<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\Tomcat_folder\tomcat.keystore"
keystorePass="yourpassword"/>
```

This modification also disables non-secure Tomcat features, leaving only HTTPS enabled (`scheme=` parameter). For security reasons, you may also need to edit `tomcat-users.xml` to delete all Tomcat users and change ServerInfo.properties to hide the identity of the Tomcat.

Are you a Linux user?
<Connector server="OtherWebServer" port="443" protocol="org.apache.coyote.http11.Http11NioProtocol" SSLEnabled="true" maxThreads="150" scheme="https" secure="true" clientAuth="false" sslProtocol="TLS" keystoreFile="/etc/tomcat/tomcat.keystore" keystorePass="yourpassword"/>

7. Restart the **Apache Tomcat** service.

Are you a Linux user?
`sudo service tomcat restart`
Note that some distributions use service name `tomcat7`.

What if secure connection is still failing on Linux?

Error message in the `/var/...../tomcat` directory:

`failed to initialize end point associated with ProtocolHandler ["http-bio-443"]`

If the problem persists, change the `port` in the `server.xml` file to a value higher than *1024*, because ports below *1024* may not be accessible to non-root users. If for some reason you have to use port *443*, you can still change the value and then forward the port. Follow the steps below to enable port redirection (e.g. from port 443 to port 8443):

1. Open and edit the firewall configuration file:

   nano /etc/sysconfig/iptables

2. Add this line to the section starting with `*nat` and ending with `COMMIT`:

   ```
   -A PREROUTING -p tcp -m tcp --dport 443 -j REDIRECT --to-ports 8443
   ```

3. Disable SELinux.