# ESET Tech Center



## Set up an HTTPS/SSL connection for ESET PROTECT (8.x)

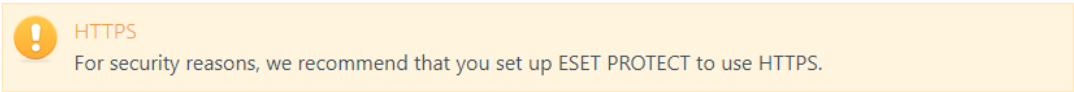
Steef | ESET Nederland - 2021-03-19 - Comments (0) - Legacy

### Issue

- You receive the warning message **Using unencrypted connection! Please configure the webserver to use HTTPS** when accessing ESET PROTECT via HTTP. This occurs after manual installation of ESET PROTECT
- Reinstall ESET PROTECT using the All-in-one installer
- Use an existing certificate
- Create a new certificate

### Solution

Are you a Linux user?




### Reinstall ESET PROTECT using the All-in-one installer

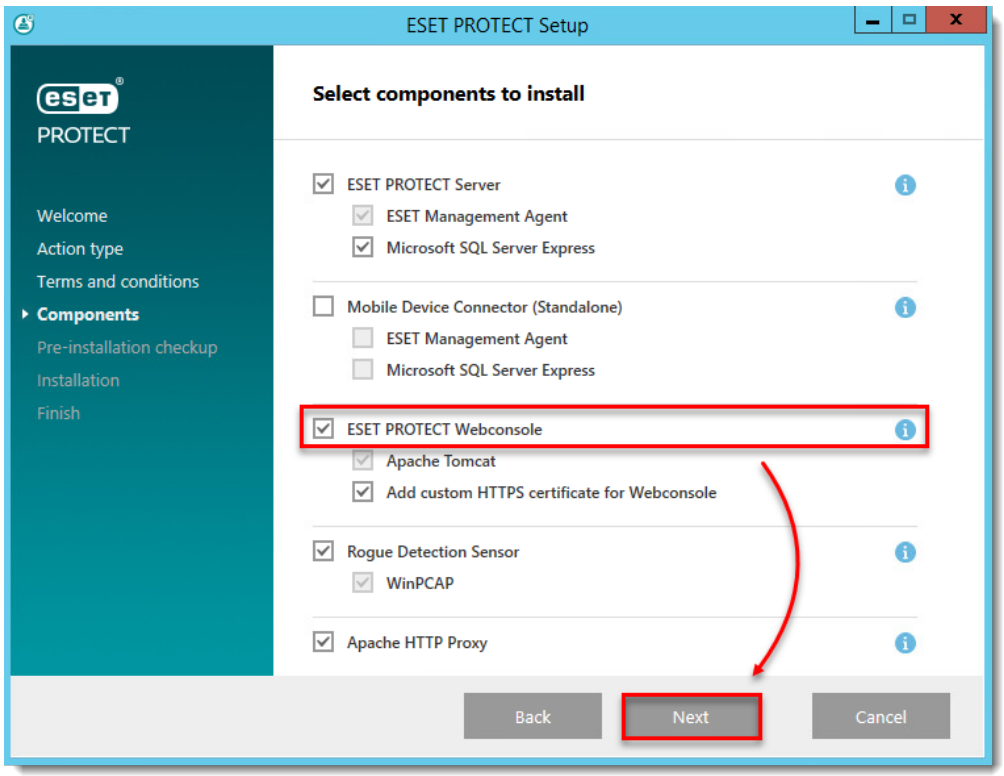
Reinstall ESET PROTECT using the All-in-one installer to automatically generate the secure connection (HTTPS) certificate.

1. Make sure Apache Tomcat is not used by any other application than ESET PROTECT.

2. Uninstall Apache Tomcat. This step also uninstalls ESET PROTECT.

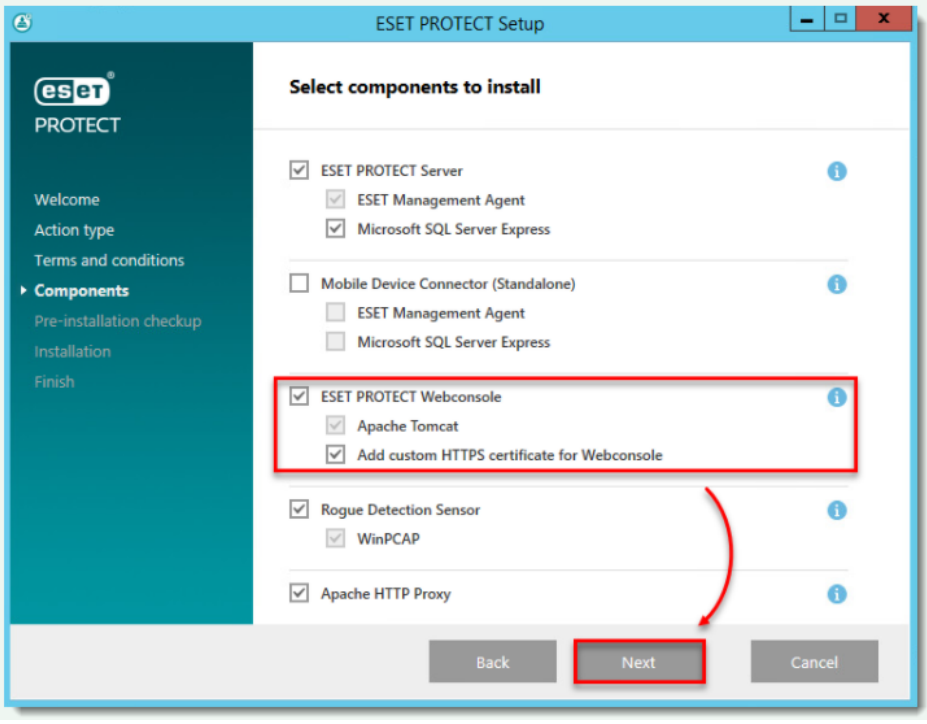
3. Download the ESET PROTECT All-in-one-installer. Use the same version as your ESET PROTECT Server.

4. Run the ESET PROTECT All-in-one-installer. Select **Install** and accept the EULA. Under **Select components to install**, select the check box **ESET PROTECT Web Console** and click **Next** The secure connection certificate will be automatically generated during the installation.



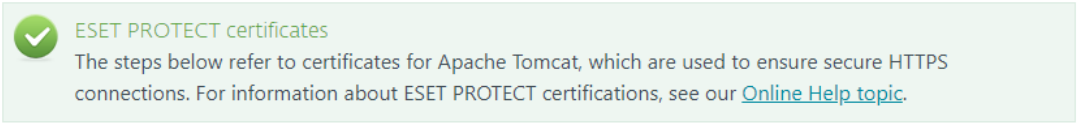

Generate a custom HTTPS certificate for ESET PROTECT Web Console

If you are installing ESET PROTECT using the All-in-one installer and you want to use a custom certificate, select the check box next to **Add Custom HTTPS certificate for Webconsole** and click **Next**.




5. Complete ESET PROTECT installation. If you installed ESET PROTECT on a different computer than the ESET PROTECT Server, configure the connection to ESET PROTECT Server.

## Use an existing certificate




he steps 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.

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 installation of Apache Tomcat), copy the following string into the Server.xml after </Engine> (use your 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"
      sslEnabledProtocols="TLSv1.2,TLSv1.3"
      ciphers="TLS_AES_256_GCM_SHA384,
      TLS_CHACHA20_POLY1305_SHA256, TLS_AES_128_GCM_SHA256,
      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 the full path to the certificate file (`.pfx`, `.keystore`, or other). If you use a non-JKS certificate (for example, a `.pfx` file), delete the `keyAlias` (it is present in Server.xml by default) and add the proper `keystoreType`.
   - `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 have a password.

---

## Create a new certificate and get it signed

Use a secure HTTPS/SSL connection https://www. for ESET PROTECT.

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

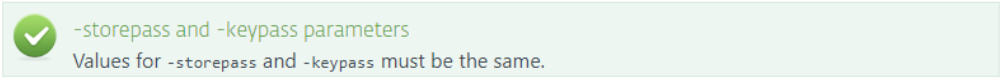
   > Apache Tomcat requires Java:
   > - Make sure that Java, ESET PROTECT, and Apache Tomcat have the same bitness (32-bit or 64-bit).
   > - If you have multiple Java versions installed on your system, we recommend that you uninstall older Java versions and keep only the latest Java.
   > - Starting January 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 enables 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"




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:ESETPROTECT


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 ESET PROTECT via HTTPS.
4. Import the root certificate and intermediate certificate of your CA to your keystore. These certificates are usually 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. After you have received the signed certificate with the Root CA, import the 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"

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.

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