ESET Tech Center

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

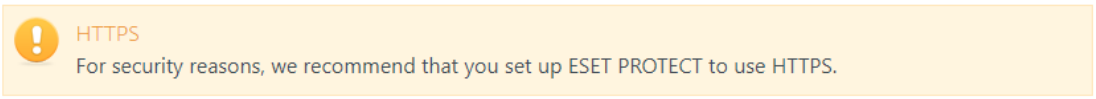
Steef | ESET Nederland - 2021-03-19 - Reacties (0) - Legacy

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 the ESET PROTECT installation
- Use an existing certificate
- Create a new certificate

Solution

Are you a Windows user?



Use an existing certificate

The steps below refer to certificates for Apache Tomcat, which are used to ensure secure HTTPS connections. For information about ESET PROTECT certifications, see our Online Help topic.

1. Move the certificate file (for example certificate_file.pfx) to a Tomcat configuration directory (for example /etc/tomcat/).

2. Open the Server.xml file located in /etc/tomcat/. The Location may vary depending on the Linux distribution.
    - If there is no <Connector after <Service name="Catalina"> in Server.xml, copy the following string into the Server.xml. Use your own values for keystoreFile, keystorePass, and keystoreType:

```
<Connector port="8443"
                protocol="HTTP/1.1"
                SSLEnabled="true"
                maxThreads="150"
                scheme="https"
                secure="true"
                clientAuth="false"
                sslEnabledProtocols="TLSv1.2,TLSv1.3"
                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"
                keystoreFile="/etc/tomcat/certificate_file.pfx"
                keystorePass="Secret_Password_123"
                keystoreType="PKCS12"                />
```

    - If <Connector is present after <Service name="Catalina"> in Server.xml, 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.

3. Restart the Tomcat service (`sudo systemctl tomcat restart`).
    - If you use a `.keystore` file, use the path to the file (`keystoreFile="/etc/tomcat/tomcat.keystore"`) and define `keyAlias` (`keyAlias="tomcat"`) instead of `keystoreType`.
    - If you want to disable HTTP:

```
<!--    <Connector port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
```

> SELinux Enabled
> Users that have SELinux enabled and receive an invalid certificate flag, may need to run the `restorecon` command to restore the SELinux security context.
>
> ```
> restorecon /etc/tomcat/my_cert_file.pfx
>
> ls -lZ
>
> -rw-r--r--. root root unconfined_u:object_r:etc_t:s0   /etc/tomcat/my_cert_file.pfx
> ```

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**, 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** file, for example `/usr/lib/jvm/”java version”/jre/bin` (the directory depends on the OS and Java version) and run the command:sudo 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:
sudo keytool -certreq -alias tomcat -file "/etc/tomcat/tomcat.csr" -keystore "/etc/tomcat/tomcat.keystore" -ext san=dns:ESETPROTECT

Replace values appropriately

Replace the value `"/etc/tomcat/tomcat.csr"` for the `-file` parameter with the actual path and filename where you want the certificate to be exported.

Replace the value `ESETPROTECT` for the `-ext` parameter with the actual hostname of the server on which your Apache Tomcat with ESET PROTECT 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 ESET PROTECT via HTTPS.

4. Import the root certificate and intermediate certificate of your CA to your `keystore`. These certificates are usually made available by the entity that signed your certificate. It is necessary because the certificate reply is validated using trusted certificates from the `keystore`.

   sudo keytool -import -alias root -file "/etc/Tomcat/root.crt" -keystore "/etc/tomcat/tomcat.keystore"

   sudo keytool -import -alias intermediate -file "/etc/Tomcat/`intermediate.crt.pem`" -keystore "/etc/tomcat/tomcat.keystore"

5. When you receive the signed certificate with the Root CA, import the public key of CA and the certificate (tomcat.cer) into your keystore. Below is a sample command that imports a signed certificate into the keystore:

   sudo keytool -import -alias tomcat -file "/etc/tomcat/tomcat.cer" -keystore "/etc/tomcat/tomcat.keystore"

   Replace values appropriately

   Replace the value `"/etc/tomcat/tomcat.csr"` for the `-file` a parameter with the actual path and filename 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="/etc/tomcat/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. ESET PROTECT may use the service name `tomcat9`.
   ```
   sudo systemctl tomcat restart
   ```

What if a 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 (for example, from port 443 to port 8443):

1. Allow remote Web Console access:
   `sudo iptables -A INPUT -p tcp -m tcp --dport 8443 -j ACCEPT`
   (Alternatively, you can open and edit the firewall configuration file (`nano /etc/sysconfig/iptables`) and 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`)

2. Remove port 8080 to disable HTTP:
   `sudo iptables -D INPUT -p tcp -m tcp --dport 8080 -j ACCEPT`

3. Save the Firewall rules:
   `iptables-save > /etc/network/iptables.rules`

4. Disable SELinux. The instructions provided are for Virtual Appliance running CentOS7 and may differ based on your Linux distribution.