Enabling SSL connection with AppServer

By default, all requests to AppServer are sent over HTTP protocol. To configure a secure SSL connection, you need to enable SSL for AppServer as described in this section.

> You can enable SSL connection with AppServer for WProofreader, SCAYT, and Web API using deployment scheme A of WebSpellChecker Deployment option. If you set up your environment using deployment scheme B, FastCGI setup is not required.

1. Before making any changes, it is recommended to stop AppServer.

2. Locate the **AppServerX.xml** configuration file in the WebSpellChecker installation folder on your server. The default path to the AppServerX.xml file is **<WebSpellChecker_Installation_Path>/AppServer/AppServerX.xml**

3. Scroll down to the section with parameters responsible for secure connection: **<SSL>...</SSL>**.

This section applies to SSL setup on Windows-based environments

```
                <!-- Secure connection -->
        <SSL>
            <EnableSSL>true</EnableSSL>
            <!-- Transport Layer Security (TLS) version. Possible values: TLSV1, TLSV1_1, TLSV1_2. -->
            <TLSVersion>TLSV1_2</TLSVersion>
            <!-- The source of an SSL certificate. Possible values: FILE, STORE. The default value is STORE. -->
            <SSLCertificateSource>STORE</SSLCertificateSource>
            <SSLCertificateStoreSettings>
            <!-- The Common Name represents the server name protected by the SSL certificate (the fully
qualified HostName). -->
            <SSLCertificateCommonName></SSLCertificateCommonName>
            <!-- If specified, Windows machine certificate store is used. Otherwise, the user's certificate
store is used.
                        The default value  is true. -->
            <SSLCertificateUseMachineStore>true</SSLCertificateUseMachineStore>
                        <!-- Possible values: MY (Personal), ROOT (Trusted Root Certification Authorities),
TRUST (Enterprise Trust),
                        CA (Intermediate Certification Authorities), USERDS (Active Directory User Object). The
default value is MY. -->
                        <SSLCertificateStore>MY</SSLCertificateStore>
            </SSLCertificateStoreSettings>
            <SSLCertificateFileSettings>
            <!--Path to PKCS #12 (*.pfx) file containing the certificate and corresponding private key.
                        Path example: C:/Program Files/WebSpellChecker/AppServer/certificate.pfx -->
            <SSLCertificateFile></SSLCertificateFile>
            <SSLCertificatePassword></SSLCertificatePassword>
            </SSLCertificateFileSettings>
        </SSL>
```

This section applies to SSL setups in Linux-based environments

```
                <!-- Secure connection -->
                <SSL>
            <EnableSSL>true</EnableSSL>
            <!-- Transport Layer Security (TLS) version. Possible values: TLSV1, TLSV1_1, TLSV1_2. -->
            <TLSVersion>TLSV1_2</TLSVersion>
                         <!-- Path to certificate file (PEM format). Path example: /opt/WSC/AppServer/cert.pem
-->
            <SSLCertificateFile></SSLCertificateFile>
            <!-- Path to private key file (PEM format). Path example: /opt/WSC/AppServer/key.pem -->
            <SSLCertificateKeyFile></SSLCertificateKeyFile>
            <SSLCertificatePassword></SSLCertificatePassword>
            <!-- Contains the path to the file or directory containing the CA/root certificates. The default
value is empty. -->
            <SSLCertificateAuthorityLocation></SSLCertificateAuthorityLocation>
            <!-- specifies whether the builtin CA certificates from OpenSSL are used. The default value is
false. -->
            <SSLCertificateLoadDefaultCA>false</SSLCertificateLoadDefaultCA>
                </SSL>
```

4. Change the default value for the **EnableSSL** parameter to **true**.

```
<EnableSSL>true</EnableSSL>
```

5. When configuring an SSL connection, please note that SSL setup steps for Windows and Linux differ:

- for Windows, you can use either a .pfx file or export an SSL certificate from Windows Certificate Store.
- for Linux, you need to specify a path to a certificate file and a private key file in *.pem format.

See the details below.

5.1. Configuring **SSL connection on Windows**.

As was already mentioned, there are two options for how to configure an SSL connection using **FILE** or exporting from **STORE**. These options are defined in the **SSLCertificateSource** parameter. Depending on your preferences, you need to select one of the options. The default value is **STORE**.

```
<!-- The source of an SSL certificate. Possible values: FILE, STORE. The default value is STORE. -->
<SSLCertificateSource>STORE</SSLCertificateSource>
```

Let's review these two options and the steps required to configure them.

Option A. Configuring SSL connection using FILE.

1. Change SSLCertificateSource value from **STORE** to **FILE**:

```
<SSLCertificateSource>FILE</SSLCertificateSource>
```

2. Specify the path to *.pfx file containing the certificate and corresponding private key, for example, **C:/Program Files/WebSpellChecker /AppServer/certificate.pfx** in the tag below:

```
<SSLCertificateFile></SSLCertificateFile>
```

3. If your SSL certificate is password-protected, type the certificate password in the **SSLCertificatePassword** tag:

```
<SSLCertificatePassword>your_certificate_password</SSLCertificatePassword>
```

Option B. Configuring SSL connection by exporting SSL certificate from Windows Certificate Store.

Specify **SSLCertificateStoreSettings** parameters which is the group of parameters required to use an SSL certificate from **STORE**, namely:

- **SSLCertificateCommonName** which is a fully qualified HostName for which an SSL certificate is issued for; a string value, for example, webspellchecker.com;
- **SSLCertificateUseMachineStore** which is an option specifying if the machine store certificate is used.

5.2. Configuring **SSL connection on Linux**.

Unlike Windows, in Linux-based environments there is only option available for SSL connection configuration, namely, **FILE**.

To configure SSL connection on Linux:

1. Specify the path to certificate file *.pem format, for example, /opt/WSC/AppServer/cert.pem in the **SSLCertificateFile** tag below:

```
<SSLCertificateFile></SSLCertificateFile>
```

2. Specify the path to a private key file in *.pem format, for example, /opt/WSC/AppServer/key.pem in the tag below:

```
<SSLCertificateKeyFile></SSLCertificateKeyFile>
```

3. If your SSL certificate is password-protected, type the certificate password in the **SSLCertificatePassword** tag.

```
<SSLCertificatePassword>your_certificate_password</SSLCertificatePassword>
```

6. As soon as the modifications are completed, start AppServer for the changes to take effect.