SSL for Analyzer

Introduction to SSL

SSL (Secure Sockets Layer) is a commonly used protocol for managing the security of message transmission on the Internet. For sites that deploy or collect sensitive information, SSL offers protection against unauthorized access to data, and assurance to your users that their information is being handled by a secure, trusted site.

- SSL enables a Web browser and a Web server to communicate securely with HTTPS (Secure Hypertext Transfer Protocol). **HTTPS** is a Web protocol built into most browsers—it encrypts and decrypts user page requests as well as the pages that are returned by the Web server. (In practical terms, this means that once SSL is enabled, the URLs used to access the site change to use *https* rather than *http*. For example *https://iseriesname:18682/analyzer.jsp*)
- SSL allows the Web browser to authenticate the Web server. The SSL protocol requires the Web server to have a digital certificate installed on it for an SSL connection to be made. SSL is an integral part of most Web browsers and Web servers.

Digital Certificates

Digital certificates are issued by a certificate authority such as Thawte Certification (*http://www.thawte.com/*) or VeriSign (*http://www.verisign.com*). Thawte is a provider of digital certifications focusing on the small business market, with a range of entry-level products, while VeriSign focuses on the Enterprise market. For those just looking to try out this functionality, test certificates are also available. (See “Using a Test Certificate” on p. 8.)

Note: The instructions in this document describe SSL setup using VeriSign as a certificate authority. However, you may use the certificate authority of your choice with instructions on that authority’s Web site.

An SSL certificate contains the following information:

- Domain for which the certificate was issued
- Owner of the certificate (also the person or entity with the right to use the domain)
- Physical location of the owner
- Validity dates of the certificate

How SSL Works

SSL relies on a key pair containing a public key and a private key that uniquely complement each other. When a message is encrypted by one key, only the other key can decrypt it.

Once SSL has been configured and enabled, the process for establishing a secure connection is as follows:

- The user logs on to the site using an URL beginning with *https* rather than *http*. For example *https://iseriesname:18682/analyzer.jsp*.
- The Web server responds by sending its digital certificate to the client, including the public key.
- The browser creates an encrypted session key from the server's public key. The session key is used to encrypt the information sent from the client to the server. Most browsers have built-in SSL support.
- After a successful authentication, the SSL connection is established between the browser and the Web server. Several things usually happen on the Web browser end:
  - From this point on, only encrypted information is exchanged between the browser and the Web server.
  - The browser displays a secure icon (for example, a locked padlock).
  - The browser does not display security warnings. Instead, it notifies the user that the Web site is secure. The user can view the site's digital certificate for more information.
- The Web server uses the private key to decrypt the encrypted information.

Configuring Your Analyzer Site to Use SSL

The following step-by-step instructions describe SSL setup using VeriSign as a certificate authority. However, you may use the certificate authority of your choice with instructions on that authority's Web site.

Before you begin, you must install Analyzer and JDK 1.4 (one of Analyzer's requirements). From here, the basic steps in setting up SSL are:

- Create a *keystore* file and Certificate Signing Request (*.csr*) on the iSeries
- Obtain a certificate.
- Save the certificate.
- Verify the CACERTS file.
- Import the certificate.
- Enable SSL in the Analyzer Server.
- Update Analyzer URLs for SSL.

Note: You may prefer to download a test certificate before fully launching SSL in production. To do this, follow the instructions in “Using a Test Certificate” on p. 8.

Creating a Keystore File and Certificate Signing Request (CSR)

You will use the keytool program to create a *keystore* file and *.csr.*

To Create a Keystore File

- ▶ Make sure Analyzer and JDK 1.4 are installed.
- ▶ On an iSeries command line, type STRQSH. This starts QShell.
- ▶ To open the directory in which you will create your files, type PWD. This means “present working directory” and will display the root directory. If you want to change the directory, use the CD command.
- ▶ Type KEYTOOL -GENKEY -KEYALG “RSA” -KEYSTORE KEYSTORE -ALIAS ANZSERVER

  where you replace *ANZSERVER* with the name of your Analyzer Server. It is important to use the same alias whenever you enter alias information for SSL.

  Note: The second *KEYSTORE* in the command is the name of the file in which the *keystore* password should be stored. You can choose any name, but we recommend using *KEYSTORE.*
- ▶ At the following prompts, enter information for your company:
  - ■ Enter keystore password. Choose any alphanumeric password with a minimum of 6 characters.
  - ■ What is your first and last name? Enter the name of your iSeries machine, such as *ourserver*. The name you enter depends on how you will reference your system in the URL. For instance, to reference it with *https://ourserver:18682/analyzer.jsp*, enter ourserver. This name must match the *iseriesname* you use in your Analyzer URL. Otherwise, you will get an error when using Analyzer with SSL.
  - ■ What is the name of your organizational unit? For example, enter IT or Support.
  - ■ What is the name of your organization? Enter the name of your company.
  - ■ What is the name of your City or Locality?
  - ■ What is the name of your State or Province? This must be spelled out, not abbreviated. For example, enter Minnesota.
  - ■ What is the two-letter country code for this unit? For example, enter US.
  - ■ Is <CN=prodserver, OU=SPSS, O=IT, L=Rochester, ST=Minnesota, C=US> correct? If the information you entered is correct, type Yes and press Enter.
  - ■ Enter key password for <ANZSERVER>. Choose any alphanumeric password with a minimum of 6 characters. For easier recall, we recommend using the same number as your *keystore* password. To choose the same number, press Enter.

- *Note:* As an alternative to answering the prompts, you can enter the following string, replacing the data with your information:

  ```
  KEYTOOL -GENKEY -KEYALG "RSA" -KEYSTORE KEYSTORE -ALIAS ANZSERVER -
  STOREPASS 123456 -DNAME "CN=John Doe, OU=ShowCase, O=IT, L=Rochester,
  ST=Minnesota, C=US"
  ```

- Write down your *keystore* and *key* passwords here or in a safe place. You will use them to complete SSL setup. _________________________________

- After the data is entered, a *keystore* streamfile is added to the IFS directory you chose with the PWD command.

To Create a CSR

- If you are not already in QShell, start it with the STRQSH command. To work with the directory in which you will save your output, type PWD and change the directory as needed.

- Enter the following keytool command:

  ```
  KEYTOOL -CERTREQ -KEYALG "RSA" -ALIAS ANZSERVER -FILE MY.HOST.COM.CSR
  -KEYSTORE KEYSTORE
  ```

  where you replace *ANZSERVER* with the name of your Analyzer Server, and *MY.HOST.COM* with a name for your *.csr* file (this can be anything you choose).

Obtaining a Certificate

The next step is to send the file to VeriSign (or another certificate signing authority) to obtain a certificate. (If you prefer to download a test certificate before fully launching SSL in production, follow the instructions in “Using a Test Certificate” on p. 8.)

- To begin the process, go to *http://www.verisign.com/client/enrollment/index.html* and enroll for a Class 1 Digital ID.

- When asked for the software vendor, select IBM HTTP.

- You will need to copy and paste your *.csr* file to this site.

- VeriSign will email you your certificate information. You will likely receive an email similar to the following:

```
-----BEGIN CERTIFICATE-----
 MIIDhjCCAu+gAwIBAgIQeO5I3hhbIHHJycO1HXvdwTANBgkqhkiG9w0
 BAQUFADBfMQswCQYDVQQGEwJVUzEXMBUGA1UEChMOVmVyaVNpZ24sIE
 luYy4xNzA1BgNVBAsTLkNsYXNzIDMgUHVibGljIFByaW1hcnkgQ2Vyd
 GlmaWNhdGlvbiBBdXRob3JpdHkwHhcNOTcwNDE3MDAwMDAwWhcNMTEx
 MDI0MjM1OTU5WjCBujEfMB0GA1UEChMWVmVyaVNpZ24gVHJ1c3QgTmV
 0d29yazEXMBUGA1UECxMOVmVyaVNpZ24sIEluYy4xMzAxBgNVBAsTKl
 ZlcmlTaWduIEludGVybmF0aW9uYWwgU2VydmVyIENBIC0gQ2xhc3MgM
 zFJMEcGA1UECxNAd3d3LnZlcmlzaWduLmNvbS9DUFMgSW5jb3JwLmJ5
 IFJlZi4gTElBQklMSVRZIExURC4oYyk5NyBWZXJpU2lnbjCBnzANBgk
 qhkiG9w0BAQEFAAOBjQAwgYkCgYEA2IKA6NYZAn0fhRg5JaJlK+G/1A
 XTvOY2O6rwTGxbtueqPHNFVbLxveqXQu2aNAoV1Klc9UAl3dkHwTKyd
 WzEyruj/lYncUOqY/UwPpMo5frxCTvzt01OOfdcSVq4wR3Tsor+cDCV
 -----END CERTIFICATE-----
```

Saving the Certificate

- Paste the text from VeriSign's email into a text editor (such as Notepad).
- Name the file and save it with a *.cer* extension in the same iSeries IFS location as your *keystore* and *.csr* files.

Verifying the CACERTS File

The *CACERTS* file is required so the server will trust VeriSign certificates.

- To verify that the *CACERTS* file exists in the correct directory, type the following command in an iSeries command line:

  ```
  WRKLNK '/QIBM/PRODDATA/JAVA400/JDK14/LIB/SECURITY'
  ```

- Select option 5 to display.
- If the *CACERTS* file does not exist, copy the one from /YourServerLibPath/JRE/LIB/SECURITY and paste it into the directory above. In the path, replace *YourServerLibPath* with the path where you installed your Analyzer Server.

Importing the Certificate

To import the certificate (*.cer* file) into your *keystore* file, use the following keytool command.

- If you are not already in QShell, start it with the STRQSH command. To work with the directory in which you will save your output, type PWD and change the directory as needed.

- Type KEYTOOL -IMPORT -ALIAS ANZSERVER -KEYSTORE KEYSTORE -TRUSTCACERTS -FILE MY.HOST.COM.CER

  where you replace *ANZSERVER* with the name of your Analyzer Server and *MY.HOST.COM.CER* with the name of your certificate file (*.cer*).

  Note: It is important to use the same alias whenever you enter alias information for SSL. If the *.csr* file was generated without the -ALIAS information (unlikely), you must find what default alias was used in the *.csr* before you run the -IMPORT command. To find the default alias, type the following command:

  KEYTOOL -LIST -RFC -KEYSTORE KEYSTORE

  Use the alias listed in this command in place of *ANZSERVER* (or your Analyzer Server) when you run the -IMPORT command.

Enabling SSL in the Analyzer Server

To enable SSL in the Analyzer Server, you will work with two *.xml* files: *secure-web-site.xml*, and *server.xml*.

To Create secure-web-site.xml

- In your mapped IFS drive, navigate to /ANZSERVER/ORION/CONFIG

  where *ANZSERVER* is the name of your Analyzer Server library.

- Create a file named *secure-web-site.xml* by making a copy of *default-web-site.xml* and renaming the copy. Do not delete or modify any information in *default-web-site.xml*.

- Using Notepad or another text editor, open *secure-web-site.xml* for editing.

- Locate the port="18682" entry. (The number is your Analyzer Server port number). Change the number to an **SSL port number.** (This is a number of your choice. We suggest 18443.)

- Add secure="true" as an attribute to the <web-site> tag.

- Manually add the following line:
  <ssl-config keystore="/user1/keystore" keystore-password="123456"/>

  and replace */user1/keystore* with the path to your *keystore* file. (This is the same location where you created your *keystore* file.) Then replace *123456* with the *keystore* password you chose in “To Create a Keystore File” on p. 3.

- When you are done, the file will look similar to the following.

Figure 1-1
Sample version of secure-web-site.xml

```
<?xml version="1.0" encoding="UTF-8"?>
 <web-site host="[ALL]" port="18443" secure="true"
 display-name="Default Orion WebSite">
 <default-web-app application="default" name="defaultWebApp"/>
 <access-log path="../log/default-web-access.log"/>
 <web-app application="Analyzer6" name="Analyzer6" root="/"
 load-on-startup="true" shared="true"/>
 <ssl-config keystore="/user1/keystore" keystore-password="123456"/>
 </web-site>
```

- Save and close the file.

To Edit server.xml

- In your mapped IFS drive, navigate to /ANZSERVER/ORION/CONFIG

  where *ANZSERVER* is the name of your Analyzer Server library.

- Using Notepad or another text editor, open *server.xml* for editing.
- Verify that the following section is in the file:

  `<web-site path="./default-web-site.xml"/>`

- **Add** the following section to the file:

  `<web-site path="./secure-web-site.xml" needs-client-auth="true"/>`

- Save and close the file.

Updating Analyzer URLs for SSL

- After you have edited the *.xml* files, restart the Analyzer Server with the STRANZ command.
- Now that you have enabled SSL, the URLs to the Analyzer Web pages must change. Instead of *http*, they will use *https*. Locate and update all URLs, using the following to access Analyzer:
  - To open the Analyzer launch page (administrators): *https://iseriesname:anzport/index.html*
  - To open the Java Web Client directly (users): *https://iseriesname:anzport/analyzer.jsp*
  - To open the Administration Tools Client directly (administrators): *https://iseriesname:anzport/administrator.jsp*

  In the examples, *iseriesname* is the name of your iSeries, and *anzport* is your Analyzer Server port number. For example: *https://prodserver:18682/analyzer.jsp*.

Using a Test Certificate

You may prefer to download a test certificate before fully launching SSL in production. Test certificates are available from VeriSign (*http://www.verisign.com*) and Thawte Certification (*http://www.thawte.com/*). Follow the instructions at the site. You will need your *.csr* file. The certificate authority will give you a *.cer* file. When you are ready for a production certificate, check with these sites (or other certificate authorities) for information and instructions.

To Install the Intermediate Certificate Authority Certificate

If you decide to use a test certificate, you must make sure the certificate is trusted. With VeriSign, you can do this by importing VeriSign's Intermediate Certificate Authority (CA) into your *keystore*. Embedding this root certificate in the product enables Analyzer to verify the validity of any certificates.

- ▶ Go to *http://www.verisign.com*.
- ▶ Enter Intermediate CA into the Search box.
- ▶ Scroll the results and click on the link that will help you install the Intermediate CA Certificate.
- ▶ The page will be titled *Installing the Intermediate CA Certificate*, and it will include a certificate in a block of text. (The text looks similar to the example in "Obtaining a Certificate" on p. 4.) Copy the text and paste it into Notepad or another text editor.
- ▶ Save the file with the name *IntermediateCA.cer* in an IFS file in the same location as your other *keystore* files.

To Import the Intermediate Certificate Authority Certificate

To import the Intermediate CA certificate into your *keystore* file, use the following keytool command:

- ▶ On an iSeries command line, type STRQSH to start QShell.
- ▶ To work with the directory in which you will save your output, type PWD and change the directory as needed.
- ▶ Type KEYTOOL -IMPORT -ALIAS INTERMEDIATECA -KEYSTORE KEYSTORE -TRUSTCACERTS -FILE INTERMEDIATECA.CER

The Intermediate CA certificate file is now ready for use. Continue with the instructions in "Saving the Certificate" on p. 5.