CONFIGURING 3RD PARTY & SELF SIGNED SSL CERTIFICATE ON IBM HTTP CERTIFICATE

Introduction

Securing a website is a big challenge in the world today. A number of various software and hardware related techniques have been introduced to cope with this issue. We came across this same problem during the deployment of one of our web applications where we wanted to implement security. For that we implemented 3rd party SSL certificate and configured it on IBM HTTP Server to make our web application more secure.

Solution

We have two options when we implement security on our website. Either we buy 3rd party SSL certificates or we can implement self signed certificates which come with WebSphere Application Server by default. 3rd party certificates are issued & verified by some authorized 3rd Party organizations like Verisign, Entrust etc.
Self signed certificates come up with Websphere Application Server

In our scenario, we will be configuring a 3rd party certificate which we have already bought from a 3rd Party organization.

IBM HTTP Server supports Secure Socket Layer (SSL) Version 2 and Version 3 and Transport Layer Security (TLS) Version 1. IBM HTTP Server is based on the Apache Web server, but for SSL configuration it requires the IBM-supplied SSL modules, rather than the OpenSSL modules. This document describes configuration of 3rd Party SSL Certificates on IBM HTTP Server, although it is possible to use another supported Web server.

Requesting certificate authority-signed personal certificates

In a production environment, use a personal certificate signed by a certificate authority (CA). The principal or the owner of the CA-signed personal certificate is authenticated by a CA when the CA signs the principal certificate. Since the certificate authorities (CAs) keep their private keys secure, the signed certificate is more trustworthy than a self-signed certificate. Certificate authorities are entities that issue valid certificates for other entities. Well-known CAs includes Verisign, Entrust, and GTE CyberTrust. You can request a test certificate or a production certificate from some of the CAs like Verisign.

Before you begin

The authentication process by a CA can take time. Commercial CAs often require up to a week to complete their authentication process. Even on-site CAs can take several minutes, if not hours, or even days, to complete their authentication process. Therefore, you must plan for the certificates that you need.

Considering the following points when you plan for the CA-signed certificate:

- On the certificate signing request that you send to the CA, specify the common name for the certificate. The common name is the primary, universal identity for the certificate. It should uniquely identify the principal that it represents. Verify that the common name is valid in the configured user registry for the WebSphere domain.
- Check the formatting of the address fields that your CA requires when planning the address for a certificate request.

Steps for this task

1. Create and send a certificate signing request (CSR) to the CA.
2. Visit the CA Web site and follow the instructions to request a test or production certificate.

Results

Once the request is accepted, the certificate authority verifies your identity and finally issues a signed certificate to you. The certificate is usually sent through e-mail.

What to do next

Once you receive the e-mail from the CA, follow the instructions to store your signed certificate as a file. Receive or store the certificate into the keystore file as a personal certificate.

Creating self-signed personal certificates

A self-signed personal certificate is a temporary digital certificate you issue to yourself, acting as the certificate authority (CA). Creating a self-signed certificate creates a private key and a public key within the key database file. The self-signed certificate is created in a keystore file and it is useful when you develop and test your application. You can also create a self-signed personal certificate from your cryptographic token device.

Before you begin

If you want to create a self-signed certificate for a keystore, you must have already created the keystore file. You can later extract the public key and add the key as a signer certificate to other truststore files.

Steps for this task

1. if it is not already running.
2. Click **New Self-Signed** from the tool bar or click **Create > New Self-Signed Certificate**.
3. Select the **X509** version and the key size that suits your application.
4. Enter the appropriate information for your self-signed certificate:

   Key Label

   Give the certificate a key label, which is used to uniquely identify the certificate within the keystore file. If you have only one certificate in each keystore file, you can assign any value to the label. However, it is good practice to use a unique label related to the server name.

Common Name

Enter the common name. This name is the primary, universal identity for the certificate; it should uniquely identify the principal that it represents. In a WebSphere environment, certificates frequently represent server principals, and the common convention is to use common names of the form *host_name* and *server_name*. The common name must be valid in the configured user registry for the secured WebSphere environment.

Organization

Enter the name of your organization.

Optional fields

Enter the organization unit (a department or division), location (city), state and province (if applicable), zip code (if applicable), and select the two-letter identifier of the country in which the server belongs. For a self-signed certificate, these fields are optional. However, commercial CAs might require them.

Validity period

Specify the lifetime of the certificate in days, or accept the default.

5. Click **OK**.

Results
Your key database file now contains a self-signed personal certificate.

Example

What to do next
If you need a test certificate signed by a certificate authority.

Receiving certificate authority-signed personal certificates

Once the certificate signing request (CSR) is accepted, a certificate authority (CA) processes the request and verifies your identity. Once approved, the CA sends the signed certificate back through e-mail. Store the signed certificate in a keystore database file. This procedure describes how to receive the CA-signed certificate into a keystore file using the key management utility (iKeyman). You use this utility the same way for both test certificates and production certificates. The primary difference between the two certificate types is the amount of time it takes for the CA to authenticate the principal your certificate represents. Test certificates are authenticated automatically based on some simple edit checks and returned to you within a few hours. Production certificates may take several days or a week to authenticate and return to you. If the CSR request is made for the cryptographic token, the certificate must be received into that token. If the request is made for the secondary key database of the token, the certificate must be received into that database.

Before you begin
Receive the signed certificate from the CA through e-mail. Follow the instructions from the CA to store the certificate into a file.

Steps for this task

1. if it is not already running.
2. Open the key database file from which you generated the request.
3. Type the password and click **OK**.
4. Select **Personal Certificates** from the pull-down list.
5. Click **Receive**.
6. Click **Data type** and select the data type of the new digital certificate, such as Base64-encoded ASCII data.
   Select the data type that matches the CA-signed certificate. If the CA sends the certificate as part of an E-mail message, you may first need to cut and paste the certificate into a separate file.
7. Type the certificate file name and location for the new digital certificate, or click **Browse** to locate the CA-signed certificate.
8. Click **OK**.
9. Type a label for the new digital certificate and click **OK**.

Results
The personal certificate list now displays the label you just gave for the new CA-signed certificate.

Example

What to do next
Once the CA-signed certificate is successfully received, you can extract or export the public key of the certificate to a file for distribution to the network.

Extracting public certificates for truststore files

Use this procedure to extract a public certificate, which includes its public key, from a keystore file. If a target truststore file already contains the signer certificate of the certificate authority (CA) that signed the certificate, you do not need to extract and add the certificate to the target truststore file. However, in general, you need to complete this procedure for a self-signed certificate.

Before you begin

Extracting a certificate from one keystore file and adding it to a truststore file is not the same as exporting the certificate and then importing it. Exporting a certificate copies all the certificate information, including its private key, and is normally only used if you want to copy a personal certificate into another keystore file as a personal certificate.

If a certificate is self-signed, extract the certificate and its public key from the keystore file and add it to the target truststore file.

If a certificate is CA-signed, verify that the CA certificate used to sign the certificate is listed as a signer certificate in the target truststore file. The keystore file must already exist and contain the certificate to be extracted.

Steps for this task

1. If it is not already running.
2. Open the keystore file from which the public certificate will be extracted.
3. Select **Personal Certificates**.
4. Click **Extract Certificate**.
5. Click **Base64-encoded ASCII data** under Data type.
6. Enter the **Certificate File Name** and **Location**.
7. Click **OK** to export the public certificate into the specified file.

Results

A certificate file that contains the public key of the signed personal certificate is now available for the target truststore file.

Adding certificates to the IBM HTTP Server

After you secure the channel between the IBM HTTP Server and the WebSphere Application Server, you must secure the channel between the IBM HTTP Server and the Web browsers that will be used to access Lotus Connections.

Configure the IBM HTTP Server to support SSL before you complete this procedure.

To secure the connection between the IBM HTTP Server and a requesting Web browser, you must import certificates into the IBM HTTP Server key store. There are different types of certificates that you can use. This procedure describes how to import the self-signed certificate that is shipped with the IBM Websphere Application Server into the IBM HTTP Server. This is just one of the methods you can use. You could also import a certificate purchased from a third-party Certificate Authority, or create and use a new self-signed certificate. See the IBM HTTP Server documentation to determine which key strategy is best for your environment.

To import the public IBM WebSphere Application Server certificate into the IBM HTTP Server, complete the following steps:

1. **From the IBM WebSphere Application Server Integrated Solutions Console, select** SSL certificate and key management **>** Key Stores and certificates**, and then select** NodeDefaultKeyStore **for a stand-alone deployment or** CellDefaultKeyStore **for a network deployment.**

2. **Click** Personal Certificates**, select the** default **check box, and then click** Extract**.**

3. **Give the extracted file a name and save it in a place you will remember.**

   Note: **The convention is to give the file an ARM extension.**

4. **Leave encoding set to Base64.**

5. **Click** OK**.**

6. **Open the** httpd.conf **configuration file from the /opt/IBM/HTTPServer/conf directory, and then edit it as follows:**

   a. **Find the directory in which the plugin-cfg.xml file is stored by searching for the WebSpherePluginConfig line. It should look something like this:**

   ```
   b.    WebSpherePluginConfig "C:\IBM\HTTPServer\Plugins\config\
    webserver1\plugin-cfg.xml"
   ```

   c. **Find the directory in which the plugin-key.kdb file is stored by searching for the term plugin-key.kdb in the plugin-cfg.xml file. For example:**

   ```
   d.    <Property
   e.     Name="keyring"
    Value="c:\IBM\HTTPServer\Plugins\config\webserver1\plugin-
   key.kdb" />
   ```

7. **From the** bin **directory of the IBM HTTP server, execute the** ikeyman.bat **file.**

8. **Click** KeyDatabaseFile **>** Open**, and then select a key database type of** CMS**. Specify** plugin-key.kdb **as the file name. Specify the file path to the KDB file. For example:**

   ```
   C:\IBM\HTTPServer\Plugins\config\webserver1\plugin-key.kdb
   ```

9. **Click** OK**, and then enter the password.**

   Note: **The default password from WebSphere Application Server is** WebAS **(case sensitive).**

10. **Click** Personal Certificates**, and then select** Signer Certificates**.**

11. **Click** Add**.**

12. **Find the file you exported with the *.arm extension, select it, and then click** OK**.**

13. **Save and exit.**

14. **Restart the IBM HTTP Server to apply the changes.**

REFERENCES :

IBM.COM:

http://www-01.ibm.com/support/docview.wss?rs=177&uid=swg21179559

http://www-01.ibm.com/support/docview.wss?rs=177&context=SSEQTJ&uid=swg21006430

http://www-01.ibm.com/support/docview.wss?rs=177&context=SSEQTJ&uid=swg21395799&loc=en_US&cs=UTF-8&lang=en

http://www-01.ibm.com/support/docview.wss?rs=177&context=SSEQTJ&uid=swg21114864

IBM Global Services
Route 100
Somers, NY 10589
U.S.A.
Produced in the United States of America
08-10