Configuring SSL for SAS Web Server: Step-by-Step Guide

Securing communication between clients and the SAS Web Server is essential for protecting data integrity and privacy. Implementing SSL (Secure Sockets Layer) encrypts communication, ensuring that sensitive information like authentication credentials and data cannot be intercepted. Below is a step-by-step guide to configure SSL for the SAS Web Server.

Prerequisites

Before starting, ensure you have the following:

1. SSL Certificate: You can use a self-signed certificate for testing or a certificate from a trusted Certificate Authority (CA) for production.

2. Private Key: The private key that corresponds to the SSL certificate.

3. Intermediate and Root CA certificates (if applicable).

4. Administrative Access to the SAS Web Server configuration files.

Steps to Configure SSL for SAS Web Server

1. Locate the SAS Web Server Configuration Files

SAS Web Server is based on the Apache HTTP Server, so you'll modify its configuration to enable SSL. The configuration files are typically located in the following directory:

- UNIX/Linux: /SAS/config/Lev1/Web/WebServer/conf/

- Windows: C:\SAS\Config\Lev1\Web\WebServer\conf\

Within this directory, you will modify the httpd.conf file and possibly ssl.conf if it exists.

2. Obtain or Create an SSL Certificate

If you don't have an SSL certificate, you can create a self-signed certificate using the OpenSSL tool:

bash

openssl req -x509 -newkey rsa:2048 -keyout server.key -out server.crt -days 365 -nodes

This will generate:

- server.crt: The self-signed SSL certificate.

- server.key: The private key for the certificate.

For production, request a certificate from a trusted CA and ensure you also have the intermediate and root CA certificates if needed.

3. Modify the httpd.conf File

Open the httpd.conf file and ensure that the following lines are present to load the SSL module:

bash

LoadModule ssl_module modules/mod_ssl.so

Include conf/extra/httpd-ssl.conf

- ssl_module: This loads the SSL module for Apache.

- httpd-ssl.conf: This includes SSL-specific configurations.

4. Edit the httpd-ssl.conf File

Next, locate the httpd-ssl.conf file (typically located in conf/extra/). If it doesn't exist, you may need to create one.

Modify or add the following directives to specify the SSL certificate, key, and other settings:

bash

```
<VirtualHost _default_:443>
  ServerName your.server.com:443
  DocumentRoot "/SAS/config/Lev1/Web/WebServer/htdocs"

  SSLEngine on
  SSLCertificateFile "/path/to/your/server.crt"
  SSLCertificateKeyFile "/path/to/your/server.key"
  SSLCertificateChainFile "/path/to/your/chain.crt"    Optional, for intermediate certs

  <Directory "/SAS/config/Lev1/Web/WebServer/htdocs">
    SSLRequireSSL
    Options None
    AllowOverride None
  </Directory>

   Redirect HTTP to HTTPS
  RewriteEngine On
  RewriteCond %{HTTPS} !=on
```

```
  RewriteRule ^/(.) https://%{SERVER_NAME}/$1 [R,L]
</VirtualHost>
```

- SSLCertificateFile: Path to your SSL certificate.

- SSLCertificateKeyFile: Path to your private key.

- SSLCertificateChainFile: Path to your intermediate certificate chain (optional).

- ServerName: The hostname of your server (e.g., your.server.com).

5. Redirect HTTP Traffic to HTTPS

To ensure that all HTTP traffic is redirected to HTTPS, add the following rewrite rules within the VirtualHost block in the httpd.conf or httpd-ssl.conf:

bash

```
RewriteEngine On
RewriteCond %{HTTPS} !=on
RewriteRule ^/(.) https://%{SERVER_NAME}/$1 [R,L]
```

6. Modify the SAS Web Application Server (Optional)

In some cases, you may also need to configure the SAS Web Application Server (e.g., SASServer1_1) to communicate over HTTPS. This involves modifying the server.xml file located in the following directory:

bash

```
<SAS_CONFIG_DIR>/Lev1/Web/WebAppServer/SASServer1_1/conf/server.xml
```

Change the connector protocol to use SSL:

```
xml
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
  maxThreads="150" scheme="https" secure="true"
  keystoreFile="/path/to/keystore.jks" keystorePass="your_password"
  clientAuth="false" sslProtocol="TLS"/>
```

7. Restart the SAS Web Server

After making these changes, restart the SAS Web Server to apply the new SSL settings.

- On UNIX/Linux:

bash

./SASWebServer.sh restart

- On Windows:

bash

SASWebServer.bat restart

8. Verify the SSL Configuration

Once the server has restarted, verify that the SSL configuration is working by navigating to your server's HTTPS URL (e.g., https://your.server.com). Check for the SSL certificate in your browser to ensure that it is valid and correctly applied.

You can also use SSL testing tools like SSL Labs or OpenSSL to confirm the proper configuration.

9. Enable Strong Cipher Suites (Optional)

For enhanced security, you may want to specify which cipher suites are allowed. This can be done by adding the following directive to your httpd-ssl.conf file:

bash

```
SSLProtocol all -SSLv2 -SSLv3
SSLCipherSuite HIGH:!aNULL:!MD5
```

This disables the insecure SSLv2 and SSLv3 protocols and allows only strong cipher suites.

Conclusion

Configuring SSL for the SAS Web Server ensures secure communication between clients and the server, safeguarding sensitive information. By following the steps outlined above, you can set up SSL with either self-signed or CA-issued certificates, redirect HTTP traffic to HTTPS, and further strengthen security with appropriate cipher suites.