Spectrum Software, Inc.
11445 Johns Creek Pkwy.
Suite 300
Duluth, GA – 30097
www.spectrumscm.com

Subject: **SpectrumSCM SSL Users Guide**

Issue Date: **March 20th, 2003**

From: **William C. Brown**
corey@spectrumsoftware.net
(770)813-4952

1.0 Introduction: SpectrumSCM supports the use of SSL (Secure Sockets Layer) to encrypt communications between the SpectrumSCM server and clients. This allows the use of SpectrumSCM to securely access proprietary information across untrusted networks, such as the Internet. Furthermore, since SpectrumSCM uses SSL, a time-tested, reliable, industry-standard security methodology, obtaining what is needed to enable secure communications is not difficult.

Since SSL uses certificates for authentication, a certificate will need to be generated and signed by a certificate authority (such as Thawte or VeriSign) and installed for use by SpectrumSCM.

2.0 Generating an SSL Certificate: Although most CAs provide methods for generating SSL certificates, Java (versions 1.4 and later) also provide a method for generating certificates. Install the Java SDK on any workstation (available at www.javasoft.com), then enter the following command to create an RSA certificate, referenced by the alias of *sslcert*, and stored in a keystore named *certstore*.

```
Keytool –genkey – keystore certstore –keyalg RSA –alias
        sslcert –storepass <store passwd> -keypass <key password>
```

** Note ** the previous command is all one line and was only folded in this document for display purposes.

The keytool command will prompt for some information to complete the certificate. Example responses follow:

```
What is your first and last name?
        [Unknown]:   John Doe
What is the name of your organizational unit?
        [Unknown]:   IT
What is the name of your organization?
        [Unknown]:   Widgets Inc.
```

```
What is the name of your city or locality?
        [Unknown]:  Atlanta
What is the name of the State or Province?
        [Unknown]:  Georgia
What is the two-letter country code?
        [Unknown]:  US
```

The tool will then prompt for the user to confirm the information that was just entered by the user:

```
is <CN=John Doe, OU=IT, O=Widgets Inc., L=Atlanta, ST=GA, C=US> correct?
        [No]: Y
```

The keytool will generate a private-public key pair (secured by the password *keypw*), assign the provided attributes to the pair, and store the keys (the public key is wrapped inside a self signed certificate) in the keystore *certstore* (secored by the password *storepw*). A *keystore* is a file that holds key pairs and certificates for SSL certification.

Now direct keytool to create a Certificate Signing Request. This is a file that contains sslcert's public key.

```
Keytool – genkey –alias sslcert –keystore certstore –file sslcert.csr
```

This file is submitted to a CA, and for a nominal fee, they will return a certificate reply, which consists of the original sslcert's public key signed with the CA's private key. The following command replaces the self-signed certificate *sslcert* inside the keystore with a certificate signed by the CR.

```
Keytool –import -alias sslcert –keystore certstore –file sslcert.reply
```

The certificate, signed by the CA, is now suitable for use by SSL and SpectrumSCM. The keystore file *certstore* should be copied to SpectrumSCM's SCM_VAR/etc/security/ssl directory.

3.0 Using Certificates with SpectrumSCM: SpectrumSCM has a properties file, *scm.properties*, which contains key/value pairs that specify a number of application settings and configurations. The *scm.properties* file is located within the SCM_VAR/etc directory, which was specified when the application was first installed. SpectrumSCM provides two client interfaces: the application interface and the HTTP interface (SCMLite). The application interface is always active when the application is running, the HTTP interface is optional. Either or both interfaces may be secured.

SpectrumSCM's SSL properties are located at the end of the *scm.properties* file. These properties are divided into those for the application interface and those for the HTTP interface. The following is a review of the purpose of each SSL property for the application interface:

Scm.ssl.provider:

Specifies the provider which supports the SSL protocols. Use the provider bundled with SpectrumSCM, *com.sun.net.ssl.internal.ssl.Provider*

Scm.ssl.inUse:

Indicates whether SpectrumSCM should secure the application interface with SSL or not.

Scm.ssl.protocol:

Specifies the SSL protocol. TLS and SSLv3 are acceptable values. Generally specify TLS.

Scm.ssl.keymanager.algorithm:

Specifies the algorithm used to encode the keystore. Use **SunX509** with the provider mentioned above.

Scm.ssl.keystore.type:

Specifies the implementation of the keystore, which for the provider given above is **JKS**

Scm.ssl.keystore.filename:

Specifies the name of the keystore containing the SSL certificate to be used for securing the application interface. In the example above, the file name was *certstore*.

Scm.ssl.keystore.password:

Specifies the passeord to the keystore containing the SSL certificate. In our example: *storepass*.

Scm.ssl.key.alias:

Specifies the alias for the SSL certificate. In the example above, *sslcert*.

Scm.ssl.key.password:

Specifies the password for the SSL certificate. In the example above, *keypass*.

The SSL properties for the HTTP interface are very similar:

Scm.http.ssl.provider:

Specifies the provider which supports the SSL protocols. In most cases use the provider bundled with SpectruSCM: com.sun.net.ssl.internal.ssl.Provider.

Scm.http.ssl.inUse:

Indicates whether SpectrumSCM should secure the HTML interface with SSL.

Scm.http.ssl.protocol:

Specifies the SSL protocol. TLS and SSLv3 are acceptable values. Generally specify TLS.

Scm.http.ssl.keymanager.algorithm:

Specifies the algorithm used to encode the keystore. The provider mentioned above uses: SunX509.

Scm.http.ssl.keystore.type:

Specifies the implementation of the keystore. For the provider mention above use **JKS**.

Scm.http.ssl.keystore.filename:

Specifies the name of the keystore containing the SSL certificate to be used for securing the application interface. In the example above, the file name was *certstore*.

Scm.http.ssl.keystore.password:

Specifies the passeord to the keystore containing the SSL certificate. In our example: *storepass*.

Scm.http.ssl.key.alias:

Specifies the alias for the SSL certificate. In the example above, *sslcert*.

Scm.http.ssl.key.password:

Specifies the password for the SSL certificate. In the example above, *keypass*.

4.0 Running SpectrumSCM Securely: Once the scm.properties file has been edited to enable SSL on either or both interfaces, start the SpectrumSCM application, or restart it if it was already running. As confirmation, the following message should appear in the server log (located in the logs directory under the installation SCM_VAR/logs directory), if the application was successfully secured with SSL:

```
SSL enabled.
```

Or on the HTTP interface:

```
SCMLite SSL enabled.
```

Once the application interface has been secured, the SpectrumSCM User Interface must also be configured to use SSL. Either use the UIConfigurationWizard to modify the startup for the application mode or, if using the web interface, direct your browser to https://SCMserver:1100 instead of the usual address.

If the application is accessed via WebStart, modify the SpectrumSCM.jnlp file that was installed in the webservers document root directory and add the following line to the arguments section at the end of the file:

<argument>-ssl</argument>

Have the users recycle through starting the client to pick the change up in the jnlp file.