IBM Spectrum Symphony RFE 97111 Readme File

About IBM Spectrum Symphony 7.1.2 security enhancements

This fix includes the following security enhancements:

- Support to configure Secure Socket Layer (SSL) authentication between VEMKD and PEM, RS (repository server) and the RS client, and SD (session director) SOAP server and SD SOAP client.
- Secure Hass Algorithm 2 (SHA-2) compliance for Transport Layer Security (TLS) trusted certificate stores for middleware communication with client data.

Readme file for: IBM® Spectrum Symphony
Product/Component Release: 7.1.2
Fix ID: sym-7.1.2-build432057-jpmc
Publication date: 2 December 2016
Last modified date: 2 December 2016

1. Scope

Applicability	
Operating systems	RHEL 6.4 or above 64-bit RHEL 7.x 64-bit RHEL 7.2 SE 64-bit
Product version	IBM Spectrum Symphony 7.1.2

Limitations and known issues	
Limitations	• For VEMKD and PEM SSL connections: 1. Enabling SSL and Kerberos between VEMKD and PEM at the same time is not supported. 2. The CAPATH setting is not supported for VEMKD and PEM SSL connections. • For RS and RS client SSL connections, the IBM Spectrum Symphony grid synchronization and the RS region features are not supported. • For SD SOAP server and SD SOAP client SSL connections: 1. Enabling SSL between the SD SOAP server and SD SOAP client is applicable to IBM Spectrum Symphony workload only; MapReduce jobs are not supported. 2. The CAPATH and SERVER_AUTH settings are not supported for SD SOAP server and SD SOAP client SSL connections.
Known issues	None

2. Installation

1) Prerequisites

Before applying this fix, you must have IBM Spectrum Symphony 7.1.2 installed.

2) Packages

File name	Description
sym-7.1.2.0_x86_64_build432057.tar.gz	This package contains the fix applicable to the master and management hosts.
symcomp-7.1.2.0_x86_64_build432057.tar.gz	The package contains the fix applicable to the compute hosts.

Notes for IBM Spectrum Symphony Developer Edition and IBM Spectrum Symphony client:
The IBM Spectrum Symphony Developer Edition and client packages are not released separately from this fix. Refer to “Installation steps” section for how to install on IBM Spectrum Symphony Developer Edition and IBM Spectrum Symphony client hosts.

3) Installation procedure

a. Before installation

1. Shut down the IBM Spectrum Symphony cluster. Log on to the master host as the cluster administrator and run:

```
$soamcontrol app disable all
$egosh service stop all
$egosh ego shutdown all
```

2. Back up the following configuration files on management hosts:

- $EGO_CONFDIR/ego.conf
- $EGO_CONFDIR/../../eservice/esc/conf/services/sd.xml
- $EGO_CONFDIR/../../eservice/esc/conf/services/rs.xml

3. Back up the following files on management hosts:

- $EGO_TOP/3.4/linux-x86_64/etc/vemkd
- $EGO_TOP/3.4/linux-x86_64/etc/pem
- $EGO_TOP/3.4/linux-x86_64/etc/lim
- $EGO_TOP/3.4/linux-x86_64/etc/egosc
- $EGO_TOP/3.4/linux-x86_64/etc/execproxy
- $EGO_TOP/3.4/linux-x86_64/etc/rs
- $EGO_TOP/3.4/linux-x86_64/bin/egodeploy
- $EGO_TOP/3.4/linux-x86_64/bin/rsdeploy
- $EGO_TOP/3.4/linux-x86_64/bin/egosh
- $EGO_TOP/3.4/linux-x86_64/lib/libvem.so.3.4.0

- $EGO_TOP/3.4/linux-x86_64/lib/libsoamdeploy.so
- $EGO_TOP/3.4/linux-x86_64/lib/libsoam_resources_7.1.2.so
- $EGO_TOP/soam/7.1.2/linux-x86_64/lib64/libvem.so.3.4.0
- $EGO_TOP/soam/7.1.2/linux-x86_64/lib64/libsoambase.so
- $EGO_TOP/soam/7.1.2/linux-x86_64/lib64/libsoamdeploy.so
- $EGO_TOP/soam/7.1.2/linux-x86_64/lib64/libsoam_resources_7.1.2.so
- $EGO_TOP/soam/7.1.2/linux-x86_64/etc/sd
- $EGO_TOP/soam/7.1.2/linux-x86_64/etc/ssm
- $EGO_TOP/soam/7.1.2/linux-x86_64/etc/sim
- $EGO_TOP/soam/7.1.2/linux-x86_64/bin/soamdeploy
- $EGO_TOP/soam/7.1.2/linux-x86_64/bin/soamcontrol
- $EGO_TOP/soam/7.1.2/linux-x86_64/bin/soamlog
- $EGO_TOP/soam/7.1.2/linux-x86_64/bin/soammod
- $EGO_TOP/soam/7.1.2/linux-x86_64/bin/soamview
- $EGO_TOP/soam/7.1.2/linux-x86_64/bin/soamreg
- $EGO_TOP/soam/7.1.2/linux-x86_64/bin/soamunreg
- $EGO_TOP/soam/7.1.2/linux-x86_64/bin/soammigrate
- $EGO_TOP/soam/7.1.2/linux-x86_64/bin/soamlogon
- $EGO_TOP/soam/7.1.2/linux-x86_64/bin/soamlogoff
- $EGO_TOP/wlp/usr/servers/gui/apps/soam/7.1.2/soamgui/WEB-INF/classes/com/platform/soam/gui/api/SOAMJniUtility.class
- $EGO_TOP/wlp/usr/servers/gui/apps/soam/7.1.2/symgui/WEB-INF/classes/com/platform/gui/pmr/general/mapreduce/gui/api/SOAMJniUtility.class
- $EGO_TOP/soam/mapreduce/7.1.2/linux-x86_64/lib/hadoop-2.4.x/hadoop-mapreduce-client-core-2.4.1.jar
- $EGO_TOP/soam/mapreduce/7.1.2/linux-x86_64/lib/hadoop-2.6.0/hadoop-mapreduce-client-core-2.6.0.jar
- $EGO_TOP/soam/mapreduce/7.1.2/linux-x86_64/lib/hadoop-2.7.x/hadoop-mapreduce-client-core-2.7.2.jar
- $EGO_TOP/perf/3.4/etc/plc.sh
- $EGO_TOP/perf/soam/7.1.2/lib/perf_soam_loader.jar
- $EGO_TOP/perf/mapreduce/7.1.2/lib/perf_pmr_loader.jar
- $EGO_TOP/gui/3.4/lib/soamgui.jar
- $EGO_TOP/wlp/usr/servers/gui/apps/soam/7.1.2/symgui/WEB-INF/lib/soamgui.jar

4. Back up the following files on all compute hosts:
   - $EGO_TOP/3.4/linux-x86_64/etc/vemkd
   - $EGO_TOP/3.4/linux-x86_64/etc/pem
   - $EGO_TOP/3.4/linux-x86_64/etc/lim
   - $EGO_TOP/3.4/linux-x86_64/etc/egosc
   - $EGO_TOP/3.4/linux-x86_64/etc/execproxy
   - $EGO_TOP/3.4/linux-x86_64/etc/rs
   - $EGO_TOP/3.4/linux-x86_64/bin/egodeploy
   - $EGO_TOP/3.4/linux-x86_64/bin/rsdeploy
   - $EGO_TOP/3.4/linux-x86_64/bin/egosh
   - $EGO_TOP/3.4/linux-x86_64/lib/libvem.so.3.4.0
   - $EGO_TOP/3.4/linux-x86_64/lib/libsoamdeploy.so
   - $EGO_TOP/3.4/linux-x86_64/lib/libsoam_resources_7.1.2.so

- $EGO_TOP/soam/7.1.2/linux-x86_64/lib64/libvem.so.3.4.0
- $EGO_TOP/soam/7.1.2/linux-x86_64/lib64/libsoambase.so
- $EGO_TOP/soam/7.1.2/linux-x86_64/lib64/libsoamdeploy.so
- $EGO_TOP/soam/7.1.2/linux-x86_64/lib64/libsoam_resources_7.1.2.so
- $EGO_TOP/soam/7.1.2/linux-x86_64/etc/sd
- $EGO_TOP/soam/7.1.2/linux-x86_64/etc/ssm
- $EGO_TOP/soam/7.1.2/linux-x86_64/etc/sim
- $EGO_TOP/soam/7.1.2/linux-x86_64/bin/soamdeploy
- $EGO_TOP/soam/7.1.2/linux-x86_64/bin/soamcontrol
- $EGO_TOP/soam/7.1.2/linux-x86_64/bin/soamlog
- $EGO_TOP/soam/7.1.2/linux-x86_64/bin/soammod
- $EGO_TOP/soam/7.1.2/linux-x86_64/bin/soamview
- $EGO_TOP/soam/7.1.2/linux-x86_64/bin/soamreg
- $EGO_TOP/soam/7.1.2/linux-x86_64/bin/soamunreg
- $EGO_TOP/soam/7.1.2/linux-x86_64/bin/soammigrate
- $EGO_TOP/soam/7.1.2/linux-x86_64/bin/soamlogon
- $EGO_TOP/soam/7.1.2/linux-x86_64/bin/soamlogoff
- $EGO_TOP/soam/mapreduce/7.1.2/linux-x86_64/lib/hadoop-2.4.x/hadoop-mapreduce-client-core-2.4.1.jar
- $EGO_TOP/soam/mapreduce/7.1.2/linux-x86_64/lib/hadoop-2.6.0/hadoop-mapreduce-client-core-2.6.0.jar
- $EGO_TOP/soam/mapreduce/7.1.2/linux-x86_64/lib/hadoop-2.7.x/hadoop-mapreduce-client-core-2.7.2.jar

5. Back up the following files on IBM Spectrum Symphony Developer Edition hosts:
   - $SOAM_HOME/7.1.2/linux-x86_64/bin/soamdeploy
   - $SOAM_HOME/7.1.2/linux-x86_64/bin/soamcontrol
   - $SOAM_HOME/7.1.2/linux-x86_64/bin/soamlog
   - $SOAM_HOME/7.1.2/linux-x86_64/bin/soammod
   - $SOAM_HOME/7.1.2/linux-x86_64/bin/soamview
   - $SOAM_HOME/7.1.2/linux-x86_64/bin/soamreg
   - $SOAM_HOME/7.1.2/linux-x86_64/bin/soamunreg
   - $SOAM_HOME/7.1.2/linux-x86_64/bin/soammigrate
   - $SOAM_HOME/7.1.2/linux-x86_64/bin/soamlogon
   - $SOAM_HOME/7.1.2/linux-x86_64/bin/soamlogoff
   - $SOAM_HOME/7.1.2/linux-x86_64/etc/rs
   - $SOAM_HOME/7.1.2/linux-x86_64/lib64/libsoam_resources_7.1.2.so

6. Back up the following files on client hosts:
   - $SOAM_HOME/lib64/libsoam_resources_7.1.2.so
   - $SOAM_HOME/lib64/libsoambase.so

7. Clean up the GUI work directories on all management hosts:
```
$rm –rf $EGO_TOP/gui/work/*
$rm –rf $EGO_TOP/gui/workarea/*
```

8. Clean up the browser cache on all client hosts.

b. Installation steps

1. Copy the `sym-7.1.2.0_x86_64_build432057.tar.gz` file to $EGO_TOP directory on all management hosts, and decompress the package.

   Note: If you previously configured a shared directory for HA failover, then also copy `security` folder under `$EGO_TOP/perf/conf/` to the $PERF_CONFDIR directory.

2. Copy the `symcomp-7.1.2.0_x86_64_build432057.tar.gz` file to the $EGO_TOP directory on all compute hosts, and decompress the package.

3. Copy the `sym-7.1.2.0_x86_64_build432057.tar.gz` file to IBM Spectrum Symphony Developer Edition hosts, and copy the corresponding decompressed files backed up previously (in the “Before installation” section) into each IBM Spectrum Symphony Developer Edition host.

4. Copy the `sym-7.1.2.0_x86_64_build432057.tar.gz` file to IBM Spectrum Symphony Client hosts, and copy the corresponding decompressed files backed up previously (in the “Before installation” section) into each IBM Spectrum Symphony client host.

5. Verify that the permissions and ownership of the files under the $EGO_TOP directory are the same as they were before applying the fix. Update any file permissions or ownership as required.

c. After installation

1. Refer to the “Configuration" section within this readme file and configure the security features that you want to enable with this fix.

2. After completing all configuration steps in the “Configuration" section, ensure configuration changes take effect by starting the cluster:
   ```
   #egosh ego start all
   ```

3. Uninstalling

1) Uninstalling from management and compute hosts

1. Disable all applications.

2. Shut down the cluster.

3. Recover all the files you backed up during installation on all hosts.

4. Start the cluster.

2) Uninstalling from IBM Spectrum Symphony Developer Edition hosts

Recover all the files previously backed up on IBM Spectrum Symphony Developer Edition hosts.

3) Uninstalling from IBM Spectrum Symphony client hosts

Recover all the files previously backed up on IBM Spectrum Symphony client hosts.

4. Configuration

1) Enabling SSL communication between VEMKD and PEM

Before you use SSL communication between VEMKD and PEM:

- Enabling SSL and Kerberos between VEMKD and PEM at the same time is not supported.
- The CAPATH setting is not supported for VEMKD and PEM SSL connections. Do not set the CAPATH value within the `ego.conf` file when configuring SSL between VEMKD and PEM.

To enable SSL communication between VEMKD and PEM, edit `$EGO_CONFDIR/ego.conf` file on all **management** hosts to add the following settings:

1. **`EGO_PEM_TRANSPORT_SECURITY`**
   Enables or disables SSL connections between PEM and VEMKD. Set to `SSL` to enable SSL connections.

2. **`EGO_KD_PEM_TS_PARAMS`**
   The EGO_KD_PEM_TS_PARAMS configuration consists of CAFILE, CERTIFICATE, CIPHER, PRIVATE_KEY, and SERVER_AUTH settings.

   For example:
   ```
   EGO_KD_PEM_TS_PARAMS="SSL[CAFILE=$HOME/secuirty/cacert.pem,CERTIFICATE=$HOME/security/vemkd.pem,CIPHER=ECDHE-ECDSA-AES256-GCM-SHA384,PRIVATE_KEY=$HOME/secuirty/vemkd.key,SERVER_AUTH={PEM}]"
   ```

   If you do not configure this parameter, ensure EGO_DEFAULT_TS_PARAMS setting is correctly configured in `ego.conf` file.

3. **`EGO_PEM_TS_PARAMS`**
   The EGO_PEM_TS_PARAMS configuration consists of CERTIFICATE, CIPHER, PRIVATE_KEY, CAFILE and SERVER_AUTH settings.

   For example:
   ```
   EGO_PEM_TS_PARAMS="SSL[CERTIFICATE=$HOME/security/pem.pem,PRIVATE_KEY=$HOME/security/pem.key,CIPHER= ECDHE-ECDSA-AES256-GCM-SHA384,CAFILE=$HOME/security/cacert.pem,SERVER_AUTH={SEC_VEMKD}]"
   ```

   If you do not configure this parameter, ensure EGO_DEFAULT_TS_PARAMS setting is correctly configured in `ego.conf` file.

4. **`EGO_KD_PEM_TS_PORT`**
   Specifies the SSL port number for VEMKD (for example, port 32781).

5. **`EGO_PEM_TS_PORT`**
   Specifies the SSL port number for PEM (for example, port 32782).

Notes:

- After you restart your cluster, the configuration settings you make to the `$EGO_CONFDIR/ego.conf` file on the management host will be automatically propagated to the `ego.conf` file on the compute hosts.

  You cannot manually change the SSL parameters introduce by connection between PEM and VEMKD in the `ego.conf` file on compute hosts.

- The files specified for the CERTIFICATE, PRIVATE_KEY, CAFILE settings must be stored in the same directory on all compute hosts.

- Configuring EGO_KD_PEM_TS_PARAMS and EGO_PEM_TS_PARAMS to use the default parameter from EGO_DEFAULT_TS_PARAMS at the same time is not supported.

The following table outlines the `ego.conf` file settings to be added for SSL communication between VEMKD and PEM:

Setting	Behavior	Default value	Valid values
EGO_PEM_TRANSPORT_SECURITY =SSL	Enables or disables SSL connection between VEMKD and PEM.	None (comment out this parameter to disable SSL)	• SSL: enables SSL connection between VEMKD and PEM • No value: do not include this setting if you want to disable SSL connection between VEMKD and PEM.
EGO_KD_PEM_TS_PARAMS ="SSL[...]"	SSL configuration for VEMKD, if SSL between VEMKD and PEM is enabled.	None	The supported ciphers for VEMKD are as follows: • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-SHA256 If you do not configure the cipher using the EGO_KD_PEM_TS_PARAMS setting, then the default cipher DHE-RSA-AES256-SHA will be used.
EGO_PEM_TS_PARAMS	SSL configuration	None	The supported ciphers for PEM

="SSL[...]"	for PEM, if SSL between PEM and VEMKD is enabled.		are as follows: • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-SHA256
EGO_KD_PEM_TS_PORT=32781	VEMKD accepts SSL connection from PEM through this port.	None	Specifies the SSL port number for VEMKD. For example, port 32781.
EGO_PEM_TS_PORT=32782	PEM accepts SSL connection from VEMKD through this port	None	Specifies the SSL port number for PEM. For example, port 32782.

2) Enabling SSL communication between RS and the RS client

Before you use SSL communication between the RS and RS client:
Note that the RS client includes:

- Command line interface for **`soamdeploy`**, **`egodeploy`**, and **`rsdeploy`**.
- RESTful API for **`soamdeploy`** and **`egodeploy`**.
- Cluster management console pages that involve package operations.
- The session director (SD).

To enable SSL communication between RS and the RS client, edit the `$EGO_CONFDIR/../../eservice/esc/conf/services/rs.xml` file on all **management** hosts to add the following settings:

1. **`RS_RSSDK_TRANSPORT`**
   The communication driver for both RS and the RS client. The driver value for SSL is `TCPIPv4SSL`. If this setting is not defined, the protocol driver is `TCPIPv4` by default to indicate no SSL authentication.

2. **`RS_RSSDK_TRANSPORT_ARG`**
   Arguments for initializing the SSL communication library on the RS side.

   The RS_RSSDK_TRANSPORT_ARG settings consist of a CERTIFICATE, CIPHER, and PRIVATE_KEY.

   The format is the same as the EGO_DEFAULT_TS_PARAMS setting in the `ego.conf` file.

   For example:
   `SSL[CERTIFICATE=$HOME/security/user.pem,CIPHER=ECDHE-ECDSA-AES256-GCM-`

`SHA384,PRIVATE_KEY=$HOME/security/user.key]`

The RS_RSSDK_TRANSPORT_ARG setting is configured in conjunction with the RSSDK_TRANSPORT_ARG setting.

3. **`RSSDK_TRANSPORT_ARG`**
   Arguments for initializing the SSL communication library on the RS client side.

   The RSSDK_TRANSPORT_ARG configuration consists of a CIPHER, CAFILE (or CAPATH), and SERVER_AUTH settings.

   The format is the same as the EGO_CLIENT_TS_PARAMS setting in the `ego.conf` file.

   For example:
   `SSL[CIPHER=ECDHE-ECDSA-AES256-GCM-SHA384,`
   `CAFILE=$HOME/security/cacert.pem,SERVER_AUTH={SEC_EGO}]`

   The RSSDK_TRANSPORT_ARG setting is configured in conjunction with the RS_RSSDK_TRANSPORT_ARG setting.

The following table outlines the `rs.xml` file settings to be added for SSL communication between RS and the RS client:

Setting	**Behavior**	**Default value**	**Valid values**
RS_RSSDK_TRANSPORT= TCPIPv4SSL	Enables or disables SSL authentication between RS and the RS client.	TCPIPv4 (to indicate that SSL is disabled by default)	• TCPIPv4 (default) • TCPIPv4SSL
RS_RSSDK_TRANSPORT_ARG= "SSL [...]"	SSL configuration for RS if SSL between RS and RS client is enabled.	None	**Note:** CBC ciphers are forbidden as they are vulnerable to SSLv3 POODLE attacks. Supported ciphers for RS are as follows: • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-SHA256

RSSDK_TRANSPORT_ARG= "SSL[...]"	SSL configuration for the RS client if SSL between RS and RS client is enabled.	None	Supported ciphers for RS client are as follows: • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-SHA256 Provide a valid and consistent SERVER_AUTH value for the RSSDK_TRANSPORT_ARG setting. This value should be same as the common name defined when you generated the certificate.

Notes for RS and RS client setting for SSL support

- You can set the RS_RSSDK_TRANSPORT_ARG and RSSDK_TRANSPORT_ARG values in the `rs.xml` file directly or use `$EGO_DEFAULT_TS_PARAMS` and `$EGO_CLIENT_TS_PARAMS` environment variables so that the system retrieves values from the `ego.conf` file. If you use the default values from the `ego.conf` file, configure the EGO_DEFAULT_TS_PARAMS and EGO_CLIENT_TS_PARAMS values in the `$EGO_CONFDIR/ego.conf` file for management hosts. Configure the EGO_CLIENT_TS_PARAMS value on all management, compute, IBM Spectrum Symphony Developer Edition, and IBM Spectrum Symphony client hosts.

- If you configured only the RS_RSSDK_TRANSPORT setting as `TCPIPv4SSL`, this enables SSL and the client will not verify certificates on the server side.

- If you configured all three parameters (RS_RSSDK_TRANSPORT, RS_RSSDK_TRANSPORT_ARG, and RSSDK_TRANSPORT_ARG), ensure you set all three with appropriate values.

- Provide a valid CIPHER value for the RS_RSSDK_TRANSPORT_ARG and RSSDK_TRANSPORT_ARG settings. Use a consistent value between RS and the RS client.

3) Enabling SSL communication between the SD SOAP server and SD SOAP client

Before you use SSL communication between the SD SOAP server and SD SOAP client:

- Enabling SSL between the SD SOAP server and SD SOAP client is applicable to IBM Spectrum Symphony workload; MapReduce jobs are not supported.

Since the SD SOAP server does not support the SSL protocol for MapReduce clients, so exception will occur when the SD SOAP server is enabled for SSL and you run MapReduce jobs using **mrsh** command.

- The CAPATH and SERVER_AUTH settings are not supported for SD SOAP server and SD SOAP client SSL connections. Do not set the CAPATH and SERVER_AUTH values within the `ego.conf` file when configuring SSL between the SD SOAP server and SD SOAP client.

  The SD SOAP interface will not check the common name define in your certificates, so the SERVER_AUTH parameter will be ignored in the SD SOAP interface.

To enable SSL communication between the SD SOAP server and SD SOAP client:

1. Edit the `EGO_CONFDIR/../../eservice/esc/conf/services/sd.xml` file on **management** hosts to add the following settings:
   a. **SD_SOAP_TRANSPORT**
   The protocol driver. The driver value for SSL is `TCPIPv4SSL`. If this parameter is not defined, the protocol driver is `TCPIPv4` by default.

   b. **SD_SOAP_TRANSPORT_ARG**
   Arguments for initializing the SD SOAP server. Arguments consist of CERTIFICATE, CIPHER, PRIVATE_KEY. The format for the arguments is the same as the one used in the EGO_DEFAULT_TS_PARAMS setting. Alternatively, a variable, such as $EGO_DEFAULT_TS_PARAMS can be substituted in place of the arguments to get values from the `ego.conf` file on the SD host.

   For example:
   ```
   SSL[CERTIFICATE=$HOME/security/user.pem,CIPHER=ECDHE-ECDSA-AES256-GCM-SHA384,PRIVATE_KEY=$HOME/security/user.key]"
   ```

   Note that the SD_SOAP_TRANSPORT_ARG setting is configured in conjunction with the SDSOAPCLIENT_ARG**.**

   c. **SDSOAPCLIENT_ARG**
   Arguments for initializing the SD SOAP client. Arguments consists of CIPHER and CAFILE, SERVER_AUTH. The format for the arguments is the same as the one used in the EGO_CLIENT_TS_PARAMS setting. Alternatively, the $EGO_CLIENT_TS_PARAMS variable can be substituted in place of the arguments and get values from the `ego.conf` file on the client's local host.

   For example:
   ```
   SSL[CIPHER= ECDHE-ECDSA-AES256-GCM-SHA384,CAFILE=$HOME/security/cacert.pem,SERVER_AUTH={SEC_EGO}]
   ```

   Note that the SDSOAPCLIENT_ARG setting is configured in conjunction with the SD_SOAP_TRANSPORT_ARG**.**

2. Import your external certificate into the `$EGO_TOP/wlp/usr/shared/resources/security/serverKeyStore.jks` file on management host:
   Note: If HA is enabled, user should execute this command on candidate host as well.

   For example:
   ```
   $EGO_TOP/jre/3.4/linux-x86_64/bin/keytool -import -keystore $EGO_TOP
   ```

```
/wlp/usr/shared/resources/security/serverKeyStore.jks -file
$HOME/security/cacert.pem
```

The default password for keystore is `Liberty`. For details, see “Enabling SSL for the cluster management console using an external certificate” in IBM Knowledge Center (http://www.ibm.com/support/knowledgecenter/SSZUMP_7.1.2/manage_cluster/security_https_pmc_enabling_prod.html).

3. Import your certificate into the ${PERF_CONFDIR}/security/serverKeyStore.jks file on management host:

   For example:
   ```
   $EGO_TOP/jre/3.4/linux-x86_64/bin/keytool -import -keystore
   $PERF_CONFDIR/security/serverKeyStore.jks -file $HOME/security/cacert.pem
   ```

   The default password for the keystore is `changeit` for the JRE behavior.

The following table outlines the `sd.xml` file settings to be added for SSL communication between the SD SOAP server and SD SOAP client:

Setting	Behavior	Default value	Valid values
SD_SOAP_TRANSPORT= TCPIPv4SSL	Enables or disables SSL authentication between the SD SOAP server and SOA client.	TCPIPv4 (to indicate that SSL is disabled by default	• TCPIPv4 (default) • TCPIPv4SSL
SD_SOAP_TRANSPORT_ARG = "SSL [...]"	SSL configuration for the SD SOAP server, if SSL between the SD SOAP server and SOAP client is enabled.	None	The format for this parameter is the same as the EGO_DEFAULT_TS_PARAMS setting. See “IBM Spectrum Symphony and SSL” in IBM Knowledge Center for details. The supported ciphers for SD SOAP server are as follows: • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-SHA256
SDSOAPCLIENT_ARG ="SSL[...]"	SSL configuration for the SD SOAP client, if SSL between the SD	None	The format for this parameter is the same as the EGO_CLIENT_TS_PARAMS setting. See “IBM Spectrum Symphony and SSL” in IBM Knowledge Center for details.

Setting	Behavior	Default value	Valid values
	SOAP server and SOAP client is enabled.		The supported ciphers for SD SOAP client (GUI and PERF) are as follows: • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-SHA256

Note that the path of CERTIFICATE, CAFILE, PRIVATE_KEY used as $HOME/security in readme, user can change to any other directory you prefer.

5. Usage

1) Feature verification

- **SSL communication between VEMKD and PEM**:
  To verify if SSL is enabled between VEMKD and PEM, check messages in VEMKD. Here is an example excerpt from the VEMKD log:

```
SSL communication between VEMKD and PEM will be enabled with these
parameter settings:SSL[CAFILE=$HOME/security
/cacert.pem,CERTIFICATE=$HOME/security
/vemkd.pem,PRIVATE_KEY=$HOME/security/vemkd.key,CIPHER=ECDHE-ECDSA-
AES256-GCM-SHA384,SERVER_AUTH={PEM}].
```

- **SSL communication between RS and the RS client**:
  To verify if SSL is enabled between RS and the RS client, check message in RS log. Here is an example excerpt from the RS log:

```
platcomm.Router - Successfully started TCPIPv4SSL communication driver,
initialized with $EGO_DEFAULT_TS_PARAMS arguments and uses zero options.
```

  To verify if SSL is for the RS client, run RS client commands and check message in RS client log, for example soamdeploy log. Here is an example excerpt from the soamdeploy log:

```
platcomm.Router - Successfully started TCPIPv4SSL communication driver,
initialized with $EGO_CLIENT_TS_PARAMS arguments and uses zero options.
```

- **SSL communication between SD SOAP server and SD SOAP client:**
  To verify If SSL is enabled between the SD SOAP server and SD SOAP client, check messages in SD. Here is an example excerpt from the SD log:

```
sd.adminManager.SdAdminListener - SdAdminListener::init(): The system
will enable SSL for the SOAP server using certificate <$HOME/
security/user.pem>, private key <$HOME/security/user.key>, and cipher
```

<ECDHE-ECDSA-AES128-SHA256>.

2) Feature interactions

- **SSL communication between VEMKD and PEM:**
  IBM Spectrum Symphony supports SSL communication for many components. You can enable all or some of them. If enable all of them, ensure EGO_KD_PEM_TS_PARAMS and EGO_PEM_TS_PARAMS are correctly configure in `$EGO_CONFDIR/ego.conf`. The addition of SSL communication between VEMKD and PEM will not impact existing SSL communication.

- **SSL communication between RS and the RS client:**
  If RS SSL is enabled, the description for EGO client RS_DEPLOY will be modified. Here is an example description:

```
$ egosh client view RS_DEPLOY
------------------------------------------------------------
CLIENT NAME: RS_DEPLOY
DESCRIPTION: platcomm:TCPIPv4SSL+$EGO_CLIENT_TS_PARAMS||platform:30916,30925
TTL         : 15
LOCATION    : 38566@hostA
USER        : Admin
CHANNEL INFORMATION:
CHANNEL              STATE
12                   CONNECTED
```

- **SSL communication between the SD SOAP server and SD SOAP client:**
  If SD SOAP SSL is enabled, the description for EGO client SD_ADMIN will be modified.

  By default (without SSL), the description of SD_ADMIN for EGO client is in this format:
  `$hostname:$port`

  With SSL enabled, the description format changes as such:
  `soap:TCPIPv4SSL+$SDSOAPCLIENT_ARG ||$hostname:$port`
  where `$SDSOAPCLIENT_ARG` comes from the `sd.xml` file.

  Here is an example description:

```
$ egosh client view SD_ADMIN
------------------------------------------------------------
CLIENT NAME: SD_ADMIN
DESCRIPTION: soap:TCPIPv4SSL+$SDSOAPCLIENT_ARG||platform:17875
TTL         : 15
LOCATION    : 48795@hostA
USER        : Admin
CHANNEL INFORMATION:
CHANNEL              STATE
17                   CONNECTED
```

  If you use the SD_ADMIN EGO client description, your WSDL client code should be updated accordingly.

3) Best practices

This section provides configuration examples for SSL connections between VEMKD and PEM, RS and the RS client, and the SD SOAP server and SD SOAP client.

- **SSL communication between VEMKD and PEM:**

  1. Edit the the `$EGO_CONFDIR/ego.conf` file by adding following lines:

```
EGO_PEM_TRANSPORT_SECURITY=SSL
EGO_KD_PEM_TS_PORT=32781
EGO_PEM_TS_PORT=32782
EGO_KD_PEM_TS_PARAMS="SSL[CAFILE=$HOME/security/cacert.pem,CERTIF
ICATE=$HOME/security/vemkd.pem,CIPHER= ECDHE-ECDSA-AES256-GCM-
SHA384,PRIVATE_KEY=$HOME/security/vemkd.key,SERVER_AUTH=pem{SEC_P
EM}]"
EGO_PEM_TS_PARAMS="SSL[CERTIFICATE=$HOME/security/pem.pem,PRIVATE
_KEY=$HOME/security/pem.key,CIPHER= ECDHE-ECDSA-AES256-GCM-
SHA384,CAFILE=$HOME/security/cacert.pem,SERVER_AUTH=vemkd{SEC_VEM
KD}]"
```

  2. Ensure that the certificates specified in the `ego.conf` file exist on all management and compute hosts.

  3. Start the master host.

  4. Start the management and compute hosts.

  5. Run `egosh rg/egosh resource list -l` to check that the cluster is successful.

- **SSL communication between RS and the RS client:**

  1. Edit the `$EGO_CONFDIR/../../eservice/esc/conf/services/rs.xml` file by adding the RS_RSSDK_TRANSPORT, RS_RSSDK_TRANSPORT_ARG, and RSSDK_TRANSPORT_ARG settings in the X86_64 hostType section:

```
<ego:EnvironmentVariable
name="RS_RSSDK_TRANSPORT">TCPIPv4SSL</ego:EnvironmentVariable>

<ego:EnvironmentVariablename="RS_RSSDK_TRANSPORT_ARG">$EGO_DEFAUL
T_TS_PARAMS</ego:EnvironmentVariable>

<ego:EnvironmentVariable
name="RSSDK_TRANSPORT_ARG">$EGO_CLIENT_TS_PARAMS</ego:Environment
Variable>
```

  2. Edit the `$EGO_CONFDIR/ego.conf` file by adding the EGO_DEFAULT_TS_PARAMS and EGO_CLIENT_TS_PARAMS on master host:

```
EGO_DEFAULT_TS_PARAMS="SSL[CERTIFICATE=$HOME/security/user.pem,CI
PHER=ECDHE-ECDSA-AES256-GCM-
SHA384,PRIVATE_KEY=$HOME/security/user.key]"

EGO_CLIENT_TS_PARAMS="SSL[CIPHER= ECDHE-ECDSA-AES256-GCM-
SHA384,CAFILE=$HOME/security/cacert.pem,SERVER_AUTH={SEC_EGO}]"
```

  3. Edit local host's `ego.conf` file on all management hosts, compute hosts, and IBM Spectrum Symphony Developer Edition hosts to add the EGO_CLIENT_TS_PARAMS value:

```
EGO_CLIENT_TS_PARAMS="SSL[CIPHER= ECDHE-ECDSA-AES256-GCM-
SHA384,CAFILE=$HOME/security/cacert.pem,SERVER_AUTH={SEC_EGO}]"
```

  4. Restart the whole cluster to make the configurations effective.

- **SSL communication between the SD SOAP server and SD SOAP client:**

  1. Edit the `$EGO_CONFDIR/../../eservice/esc/conf/services/sd.xml` file on the management host by adding the following lines:

```
<ego:EnvironmentVariable
name="SD_SOAP_TRANSPORT">TCPIPv4SSL</ego:EnvironmentVariable>

<ego:EnvironmentVariable
name="SD_SOAP_TRANSPORT_ARG">SSL[CERTIFICATE=$HOME/security/user.
pem,CIPHER=ECDHE-ECDSA-AES128-
SHA256,PRIVATE_KEY=$HOME/security/user.key]</ego:EnvironmentVaria
ble>

<ego:EnvironmentVariable
name="SDSOAPCLIENT_ARG">SSL[CIPHER=ECDHE-ECDSA-AES128-
SHA256,CAFILE=$HOME/security/cacert.pem,SERVER_AUTH={SEC_EGO}]</e
go:EnvironmentVariable>
```

  2. Import your certificate into the `$EGO_TOP/wlp/usr/shared/resources/security/serverKeyStore.jks` file on management host:

  Note: If HA is enabled, also run this command on each candidate host.

```
$EGO_TOP/jre/3.4/linux-x86_64/bin/keytool -import -keystore
$EGO_TOP/wlp/usr/shared/resources/security/serverKeyStore.jks -
file $HOME/security/cacert.pem -storepass Liberty
```

  3. Import your certificate into the `${PERF_CONFDIR}/security/serverKeyStore.jks` file on management host:

```
$EGO_TOP/jre/3.4/linux-x86_64/bin/keytool -import -keystore
$PERF_CONFDIR/security/serverKeyStore.jks -file
$HOME/security/cacert.pem -storepass changeit
```

  4. Stop the EGOSC daemon and restart the SD, WEBGUI, and loader controller (PLC) services.

6. Troubleshooting

If SSL is not enabled successfully, follow below steps before checking the error logs.

1. Verify that your certificate was issued by a specific CA:

```
[root@host1 ssl]# openssl verify -CAfile cacert.pem user.pem
user.pem: OK
```

2. Check that the CERTIFICATE, PRIVATE_KEY, and CAFILE settings exist in the `ego.conf` file.

3. Check that the SERVER_AUTH setting is the same value as the common name (used when generating the certificate) when its type is HOST or {string}.

4. Check that the CIPHER setting is a supported cipher, and the cipher specified for server and client match.

- **SSL communication between VEMKD and PEM:**
  1. Ensure the EGO_PEM_TRANSPORT_SECURITY setting is set to `SSL`.

2. Check the values of the EGO_KD_PEM_TS_PARAMS and EGO_PEM_TS_PARAMS settings are correctly configured in `ego.conf`.

3. If all the above configurations are correct, refer to the `$EGO_TOP/kernel/log/vemkd.log.host_name` and `$EGO_TOP/kernel/log/pem.log.host_name` log files for additional troubleshooting.

- **SSL communication between RS and the RS client:**
  1. Ensure the RS_RSSDK_TRANSPORT setting is set to `TCPIPv4SSL`.

  2. Check the value of the RS_RSSDK_TRANSPORT_ARG setting is `$EGO_DEFAULT_TS_PARAMS`, or in this format: `"SSL[CERTIFICATE=$HOME/security/user.pem,CIPHER= ECDHE-ECDSA-AES256-GCM-SHA384,PRIVATE_KEY=$HOME/security/user.key]"`

  3. Check that the value of the RSSDK_TRANSPORT_ARG setting is `$EGO_CLIENT_TS_PARAMS`, or in this format: `"SSL[CIPHER= ECDHE-ECDSA-AES256-GCM-SHA384,CAFILE=$HOME/security/cacert.pem,SERVER_AUTH={SEC_EGO}]"`

  4. When the value of RS_RSSDK_TRANSPORT_ARG is `$EGO_DEFAULT_TS_PARAMS` in the `rs.xml` file, check that the EGO_DEFAULT_TS_PARAMS value is correctly configured in the `$EGO_CONFDIR/ego.conf` file on all management hosts.

  5. When the value of RSSDK_TRANSPORT_ARG is `$EGO_CLIENT_TS_PARAMS` in the `rs.xml` file, check that the EGO_CLIENT_TS_PARAMS value is correctly configured in the `$EGO_CONFDIR/ego.conf` file on all management, compute, and IBM Spectrum Symphony Developer Edition hosts.

  6. If all the above configurations are correct, refer to the `$EGO_TOP/eservice/rs/log/rs.`*`host_name`*`.log` file for additional troubleshooting.

  7. RS may report many ERROR logs related to SSL configuration or operation errors. These are triggered by the grid synchronization client EGO repository server agent for two reasons:

     The grid synchronization client needs to connect to RS. When SSL communication for RS and the RS client is enabled, grid synchronization uses a non-SSL connection, while RS uses an SSL connection, and therefore, the grid synchronization connection to RS will fail.

     When you set the RS_RSSDK_TRANSPORT_ARG and RSSDK_TRANSPORT_ARG values in the rs.xml file directly, ERROR messages show in the rs.log file.

     Here is an example of rs.xml file:

```
<ego:EnvironmentVariable name="RS_RSSDK_TRANSPORT">TCPIPv4SSL
</ego:EnvironmentVariable>

<ego:EnvironmentVariable name="RS_RSSDK_TRANSPORT_ARG">
   SSL[CERTIFICATE=$HOME/security/user.pem,CIPHER= ECDHE-ECDSA-
AES256-GCM-SHA384,PRIVATE_KEY=$HOME/security/user.key]
</ego:EnvironmentVariable>

<ego:EnvironmentVariable name="RSSDK_TRANSPORT_ARG">
   SSL[CIPHER= ECDHE-ECDSA-AES256-GCM-
SHA384,CAFILE=$HOME/security/cacert.pem,SERVER_AUTH={SEC_EGO}]
```

```
</ego:EnvironmentVariable>
```

Here is an example of rs.log file:

```
[root@ib15b01]# cat eservice/rs/log/rs.ib15b01.log |grep SSL
2016-11-29 07:50:21.288 GMT INFO [6417:140517980743488]
platcomm.Router - Successfully started TCPIPv4SSL communication
driver, initialized with
SSL[CERTIFICATE=$HOME/security/user.pem,CIPHER=ECDHE-ECDSA-AES256-
GCM-SHA384,PRIVATE_KEY=$HOME/security/user.key] arguments and uses
zero options.
2016-11-29 07:50:36.258 GMT ERROR [6417:140517785958144] platcommdrv
- Code[S61017]: TCPdriver.cpp:3959   Communication driver returns
SSL configuration or operation error: "Reactive_Stream::open()
SSL_ERROR_SYSCALL ssl_error=336130315:error:1408F10B:SSL
routines:SSL3_GET_RECORD:wrong version number".
2016-11-29 07:50:36.258 GMT ERROR [6417:140517785958144] platcommdrv
- Code[S61017]: TCPdriver.cpp:3964   Communication driver returns
SSL configuration or operation error: "Possible reasons: 1) the TLS
version between the server and client do not match, or one side does
not use TLS; 2) there is no shared cipher between the server and
client; 3) you have not specified a cipher, or have specified an not
supported cipher; 4) the server and client SSL driver names do not
match; 5) you have not specified a server or client SSL driver name;
or 6) the server side certificate cannot be verified with the value
specified for the CAFILE or CAPATH parameter.".
```

- **SSL communication between the SD SOAP server and SD SOAP client:**
  1. Ensure SD_SOAP_TRANSPORT is right and supported value: `TCPIPv4/TCPIPv4SSL`.
  2. Check that the SD_SOAP_TRANSPORT_ARG setting is `$EGO_DEFAULT_TS_PARAMS`, or in this format:
     `"SSL[CERTIFICATE=$HOME/security/user.pem,CIPHER= ECDHE-ECDSA-AES256-GCM-SHA384,PRIVATE_KEY=$HOME/security/user.key]"`
  3. Check that the SDSOAPCLIENT_ARG setting is `$EGO_CLIENT_TS_PARAMS`, or in this format:
     `"SSL[CIPHER= ECDHE-ECDSA-AES256-GCM-SHA384,CAFILE=$HOME/security/cacert.pem,SERVER_ATH={SEC_EGO}"`
  4. When the value of SOAP_TRANSPORT_ARG is `$EGO_DEFAULT_TS_PARAMS` in the `sd.xml` file, check that the EGO_DEFAULT_TS_PARAMS value is correctly configured in the `$EGO_CONFDIR/ego.conf` file on all management hosts.
  5. When the value of SDSOAPCLIENT_ARG is `$EGO_CLIENT_TS_PARAMS` in the `sd.xml` file, check that the EGO_CLIENT_TS_PARAMS value is correctly configured in the `$EGO_CONFDIR/ego.conf` file on all management, compute, and IBM Spectrum Symphony Developer Edition hosts.
  6. If all of the above configurations are correct, refer to the `{SOAM_HOME}/logs/sd.host_name.log` file for additional troubleshooting.

7. Copyright and trademark information