



**Hewlett Packard**  
Enterprise

# **HPE Security ArcSight Event Broker and Connectors**

Configuring TLS between Event Broker and SmartConnectors

April 14, 2017

## Legal Notices

### Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

The network information used in the examples in this document (including IP addresses and hostnames) is for illustration purposes only.

HPE Security ArcSight products are highly flexible and function as you configure them. The accessibility, integrity, and confidentiality of your data is your responsibility. Implement a comprehensive security strategy and follow good security practices.

This document is confidential.

### Restricted Rights Legend

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

### Copyright Notice

© Copyright 2017 Hewlett Packard Enterprise Development, LP

Follow this link to see a complete statement of copyrights and acknowledgements:

<https://community.softwaregrp.com/t5/Discussions/Third-Party-Copyright-Notices-and-License-Terms/td-p/1589228>

## Support

### Contact Information

<b>Phone</b>	A list of phone numbers is available on the Technical Support Page: <a href="https://softwaresupport.softwaregrp.com/support-contact-information">https://softwaresupport.softwaregrp.com/support-contact-information</a>
<b>Support Web Site</b>	<a href="https://softwaresupport.softwaregrp.com/">https://softwaresupport.softwaregrp.com/</a>
<b>ArcSight Product Documentation</b>	<a href="https://community.softwaregrp.com/t5/ArcSight-Product-Documentation/ct-p/productdocs">https://community.softwaregrp.com/t5/ArcSight-Product-Documentation/ct-p/productdocs</a>

### Revision History

Date	Description
04/14/2017	Initial release.

# Contents

- Chapter 1: What is FIPS? ..... 4
  
- Chapter 2: Client Authentication ..... 5
  - Configure an Event Broker Destination with Client Authentication in FIPS Mode ..... 5
    - Step 1: On the Connector Server ..... 5
    - Step 2: On the Event Broker Server ..... 7
    - Step 3: On the Connector Server ..... 8
    - Step 4: On the Event Broker Server ..... 8
    - Step 5: On the Connector Server ..... 8
    - Step 6: On the Event Broker Server ..... 12
  - Configure an Event Broker Destination with Client Authentication in Non-FIPS Mode ..... 13
    - Step 1: On the Connector Server ..... 13
    - Step 2: On the Event Broker Server ..... 14
    - Step 3: On the Connector Server ..... 15
    - Step 4: On the Event Broker Server ..... 15
    - Step 5: On the Connector Server ..... 15
    - Step 6: On the Event Broker Server ..... 18
  - Configure an Event Broker Destination without Client Authentication in FIPS Mode ..... 19
    - On the Event Broker Server ..... 19
    - On the SmartConnector Server ..... 19
  
- Send Documentation Feedback ..... 22

# Chapter 1: What is FIPS?

Under the Information Technology Management Reform Act (Public Law 104-106), the Secretary of Commerce approves standards and guidelines that are developed by the National Institute of Standards and Technology (NIST) for Federal computer systems. These standards and guidelines are issued by NIST as Federal Information Processing Standards (FIPS) for use government-wide. NIST develops FIPS when there are compelling Federal government requirements such as for security and interoperability and there are no acceptable industry standards or solutions.

FIPS Suite B includes cryptographic algorithms for hashing, digital signatures, and key exchange. The entire suite of cryptographic algorithms is intended to protect both classified and unclassified national security systems and information.

**Note:** When FIPS-compliant connectors connect to a non-FIPS-compliant destination, the solution is not considered FIPS compliant. When the destination is installed in FIPS Suite B compliant mode, the SmartConnectors must also be installed in FIPS Suite B compliant mode.

# Chapter 2: Client Authentication

Follow these instructions to enable Client Authentication for Event Broker and SmartConnectors with Event Broker destinations. For additional Event Broker configuration, see the *Administrator's Guide for Event Broker* and "Event Broker" in the *Smart Connector User Guide* on [Protect 724](#).

**Note:** The following examples:

- use the default password. See the appendix for FIPS Compliant SmartConnectors in the *SmartConnector User Guide* on [Protect 724](#) to set a non-default password.
- assume that you are on the Linux platform. For Windows platforms, use backslashes when entering commands.
- assume you are using a command prompt window to enter Windows commands. Do not use Windows PowerShell.

## Configure an Event Broker Destination with Client Authentication in FIPS Mode

Follow these steps to configure an Event Broker (EB) destination from the SmartConnector with client authentication in FIPS mode.

### Step 1: On the Connector Server

1. Prepare the connector:
  - **If the connector is not yet installed:** Run the installer. After core software has been installed, you will see a window that lets you select **Add a Connector** or **Select Global Parameters**. Check **Select Global Parameters**, and on the window displayed, select **Set FIPS mode**. Set to **Enabled**.
  - **If the connector is already installed:** Run the installer. Select **Set Global Parameters** and set **Set FIPS Mode** to **Enabled**.
2. Navigate to the connector's current directory, for example:

```
cd <install_dir>/current
```
3. Apply the following workaround for a Java keytool issue:
  - a. Create a new file, `agent.security`, at `<install_dir>/current/user/agent` (or at `<install_dir>\current\user\agent` on Windows platforms).
  - b. Add the following content to the file and save:

```
security.provider.1=org.bouncycastle.jcajce.provider  
.BouncyCastleFipsProvider  
security.provider.2=com.sun.net.ssl.internal.ssl.Provider BCFIPS  
security.provider.3=sun.security.provider.Sun
```

c. Move the `lib/agent/fips/bcprov-jdk14-119.jar` file to the current directory.

4. Set the environment variables for static values used by keytool:

```
export CURRENT=<full path to this "current" folder>  
export BC_OPTS="-storetype BCFKS -providertype BCFIPS  
-J-Djava.security.egd=file:/dev/urandom  
-J-Djava.ext.dirs=${CURRENT}/jre/lib/ext:${CURRENT}/lib/agent/fips  
-J-Djava.security.properties=${CURRENT}/user/agent/agent.security"  
export EB=<event broker hostname>_<event broker port>  
export STORES=${CURRENT}/user/agent/stores
```

**On Windows platforms:**

```
set CURRENT=<full path to this "current" folder>  
set BC_OPTS=-storetype BCFKS -providertype BCFIPS  
-J-Djava.ext.dirs=%CURRENT%\jre\lib\ext;%CURRENT%\lib\agent\fips  
-J-Djava.security.properties=%CURRENT%\user\agent\agent.security  
set EB=<event broker hostname>_<event broker port>  
set STORES=%CURRENT%\user\agent\stores
```

5. Create the `user/agent/stores` directory if it does not already exist, for example:

```
mkdir ${STORES}
```

**On Windows platforms:**

```
mkdir %STORES%
```

6. Create the connector key pair, for example (the connector FQDN, OU, O, L, ST, and C values must be changed for your company and location):

```
jre/bin/keytool ${BC_OPTS} -genkeypair -alias ${EB} -keystore  
${STORES}/${EB}.keystore.bcfips -dname "cn=<Connector  
FQDN>,OU=Arcsight,O=HP,L=Sunnyvale,ST=CA,C=US" -validity 365
```

**On Windows platforms:**

```
jre\bin\keytool %BC_OPTS% -genkeypair -alias %EB% -keystore  
%STORES%\%EB%.keystore.bcfips -dname "cn=<Connector  
FQDN>,OU=Arcsight,O=HP,L=Sunnyvale,ST=CA,C=US" -validity 365
```

When prompted, enter the password. Note the password; you will need it again in a later step. Press **Enter** to use the same password for the key. If you want to match the default value in the properties file, use the password change it.

7. List the key store entries. There should be one private key.

```
jre/bin/keytool ${BC_OPTS} -list -keystore ${STORES}/${EB}.keystore.bcfips  
-storepass ${STORE_PASSWD}
```

**On Windows platforms:**

```
jre\bin\keytool %BC_OPTS% -list -keystore %STORES%\%EB%.keystore.bcfips  
-storepass <key store password>
```

8. Create a Certificate Signing Request (CSR), for example:

```
jre/bin/keytool ${BC_OPTS} -certreq -alias ${EB} -keystore  
${STORES}/${EB}.keystore.bcfips -file ${STORES}/${EB}-cert-req -storepass  
${STORE_PASSWD}
```

**On Windows platforms:**

```
jre\bin\keytool %BC_OPTS% -certreq -alias %EB% -keystore  
%STORES%\%EB%.keystore.bcfips -file %STORES%\%EB%-cert-req -storepass  
%STORE_PASSWD%
```

## Step 2: On the Event Broker Server

1. When EB is first installed, it's setup to use self-signed certificates. To replace the self-signed certificates, obtain your company's root CA certificate, and an intermediate certificate and key pair. Place them in /tmp with the following names:

```
/tmp/intermediate.cert.pem
```

```
/tmp/intermediate.key.pem
```

```
/tmp/ca.cert.pem
```

Use the following command to add them to Event Broker:

```
/opt/arcsight/kubernetes/scripts/arcsight-cert-util.sh write --re-  
key=/tmp/intermediate.key.pem --re-crt=/tmp/intermediate.cert.pem --re-  
ca=/tmp/ca.cert.pem
```

**Note:** If EB was already deployed when running the above command, all Kafka broker nodes must be restarted. Either redeploy EB, or delete each Kafka pod. For example:

To list the pods:

```
kubectl get pods --all-namespaces
```

To delete the first Kafka pod:

```
kubectl delete pod eb-kafka-0 -n eventbroker1
```

Repeat for all the kafka pods.

```
2. export CA_CERT=/tmp/ca.cert.pem export INTERMEDIATE_CA_
   CRT=/tmp/intermediate.cert.pem
```

```
export INTERMEDIATE_CA_KEY=/tmp/intermediate.key.pem
```

```
export FIPS_CA_TMP=/opt/fips_ca_tmp export EB=<event broker hostname>_<event
broker port>
```

```
3. Create a temp location on the eb master server: mkdir $FIPS_CA_TMP
```

## Step 3: On the Connector Server

Copy the `${STORES}/${EB}-cert-req` file (`%STORES%\%EB%-cert-req` on Windows platforms) from the connector to the Event Broker directory created above, `/opt/fips_ca_tmp`.

## Step 4: On the Event Broker Server

Create the signed certificate, for example:

```
/bin/openssl x509 -req -CA $INTERMEDIATE_CA_CRT -CAkey $INTERMEDIATE_CA_KEY -
in ${EB} -cert-req -out ${EB} -cert-signed -days 365 -CAcreateserial -sha256
```

## Step 5: On the Connector Server

1. Copy the `${EB}-cert-signed` certificate from the Event Broker to the connector's `${STORES}` directory. (On the Windows platform, copy the `%EB%-cert-signed` certificate to the connector's `%STORES%` directory.)
2. Copy the `ca.cert.pem` certificate from the Event Broker to the connector's `${STORES}` directory. (On the Windows platform, copy the certificate to the `%STORES%` directory.)
3. Copy the `intermediate.cert.pem` certificate from the Event Broker to the connector's `${STORES}` directory. (On the Windows platform, copy the certificate to the `%STORES%` directory.)
4. Import the CA certificate to the trust store, for example:

```
jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/ca.cert.pem -alias
CARoot -keystore ${STORES}/${EB}.truststore.bcfips -storepass ${STORE_
PASSWD}
```

### On Windows platforms:

```
jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\ca.cert.pem -alias
CARoot -keystore %STORES%\%EB%.truststore.bcfips -storepass %STORE_
PASSWD%
```

5. Import the intermediate certificate to the trust store, for example:

```
jre/bin/keytool ${BC_OPTS} -importcert -file  
${STORES}/intermediate.cert.pem -alias INTCARoot -keystore  
${STORES}/${EB}.truststore.bcfips -storepass <trust store password>
```

**On Windows platforms:**

```
jre\bin\keytool %BC_OPTS% -importcert -file  
%STORES%\intermediate.cert.pem -aliasINTCARoot -keystore  
%STORES%\%EB%.truststore.bcfips -storepass <trust store password>
```

6. Import the CA certificate to the key store, for example:

```
jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/ca.cert.pem -alias  
CARoot -keystore ${STORES}/${EB}.keystore.bcfips -storepass ${STORE_  
PASSWD}
```

**On Windows platforms:**

```
jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\ca.cert.pem -alias  
CARoot -keystore %STORES%\%EB%.keystore.bcfips -storepass %STORE_  
PASSWD%
```

7. When prompted, enter **yes** to trust the certificate.
8. Import the signed certificate to the key store, for example:

```
jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/${EB}-cert-signed  
-alias ${EB} -keystore ${STORES}/${EB}.keystore.bcfips -storepass ${STORE_  
PASSWD}
```

**On Windows platforms:**

```
jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\%EB%-cert-signed  
-alias %EB% -keystore %STORES%\%EB%.keystore.bcfips -storepass %STORE_  
PASSWD%
```

If successful, this command will return the message, *Certificate reply was installed in keystore.*

9. Import the intermediate certificate to the key store, for example:

```
jre/bin/keytool ${BC_OPTS} -importcert -file  
${STORES}/intermediate.cert.pem -alias  
INTCARoot -keystore ${STORES}/${EB}.keystore.bcfips -storepass <key store  
password>
```

**On Windows platforms:**

```
jre\bin\keytool %BC_OPTS% -importcert -file  
%STORES%\intermediate.cert.pem -alias  
INTCARoot -keystore %STORES%\%EB%.keystore.bcfips -storepass <key store  
password>
```

If successful, this command will return the message, *Certificate reply was installed in keystore.*

10. Navigate to the `bin` directory and run agent setup. Install a connector with Event Broker as the destination, for example:

```
cd <installation dir>/current/bin  
./runagentsetup.sh
```

**On Windows platforms:**

```
cd <installation dir>\current\bin  
runagentsetup.bat
```

- a. When completing the Event Broker destination fields, use the same values as in Step 8 for the path and password.
  - b. Set **Use SSL/TLS** to **true**.
  - c. Set **Use SSL/TLS Authentication** to **true**.
11. Cleanup. Delete the following files:

**Caution:** The following files should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Event Broker. These files are very sensitive and should not be distributed to other machines.

```
rm ${STORES}/intermediate.cert.pem  
rm ${STORES}/intermediate.key.pem
```

## Configuring TLS between Event Broker and SmartConnectors

```
rm ${STORES}/${EB}-cert-signed  
rm ${STORES}/${EB}-cert-req
```

**On Windows platforms:**

```
del %STORES%\intermediate.cert.pem  
del %STORES%\intermediate.key.pem  
del %STORES%\%EB%-cert-signed  
del %STORES%\%EB%-cert-req
```

12. Move the `bcprov-jdk14-119.jar` file back to the `lib/agent/fips` directory (or `lib\agent\fips` on Windows platforms).

## Step 6: On the Event Broker Server

To cleanup the Event Broker server, delete the temporary folder where the certificate was signed and the certificate and key files in `/tmp`.

**Caution:** The temporary folder should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Event Broker. These files are very sensitive and should not be distributed to other machines.

# Configure an Event Broker Destination with Client Authentication in Non-FIPS Mode

Follow these steps to configure an Event Broker (EB) destination from the SmartConnector with client authentication, but in non-FIPS mode.

## Step 1: On the Connector Server

1. Prepare the SmartConnector:

- **If the connector is not yet installed:** Run the installer. After core software has been installed, you will see a window that lets you select **Add a Connector** or **Select Global Parameters**. Check **Select Global Parameters**, and on the window displayed, select **Set FIPS mode**. Set to **Enabled**.
- **If the connector is already installed:** Run the installer. Select **Set Global Parameters** and set **Set FIPS Mode** to **Enabled**.

2. Navigate to the connector's current directory, for example:

```
cd <install dir>/current
```

**On Windows platforms:**

```
cd <install dir>\current
```

3. Set the environment variables for the static values used by keytool, for example:

```
export CURRENT=<full path to this "current" folder>  
export EB=<eb hostname>_<eb port>  
export STORES=${CURRENT}/user/agent/stores
```

**On Windows platforms:**

```
set CURRENT=<full path to this "current" folder>  
set EB=<eb hostname>_<eb port>  
set STORES=%CURRENT%\user\agent\stores
```

4. Create the user/agent/stores directory if does not already exist, for example:

```
mkdir ${STORES}
```

**On Windows platforms:**

```
mkdir %STORES%
```

5. Create the connector key pair, for example:

```
jre/bin/keytool -genkeypair -alias ${EB} -keystore  
${STORES}/${EB}.keystore.jks -dname "cn=<Connector  
FQDN>,OU=Arcsight,O=HP,L=Sunnyvale,ST=CA,C=US" -validity 365
```

**On Windows platforms:**

```
jre\bin\keytool -genkeypair -alias %EB% -keystore  
%STORES%\%EB%.keystore.jks -dname "cn=<Connector  
FQDN>,OU=Arcsight,O=HP,L=Sunnyvale,ST=CA,C=US" -validity 365
```

When prompted, enter the password. Note the password; you will need it again in a later step. Press Enter to use the same password for the key.

6. List the key store entries. There should be one private key.

```
jre/bin/keytool -list -keystore ${STORES}/${EB}.keystore.jks -storepass  
${STORE_PASSWD}
```

**On Windows platforms:**

```
jre\bin\keytool -list -keystore %STORES%\%EB%.keystore.jks -storepass  
%STORE_PASSWD%
```

7. Create a Certificate Signing Request (CSR), for example:

```
jre/bin/keytool -certreq -alias ${EB} -keystore  
${STORES}/${EB}.keystore.jks -file ${STORES}/${EB}-cert-req -storepass  
${STORE_PASSWD}
```

**On Windows platforms:**

```
jre\bin\keytool -certreq -alias %EB% -keystore  
%STORES%\%EB%.keystore.jks -file %STORES%\%EB%-cert-req -storepass  
%STORE_PASSWD%
```

## Step 2: On the Event Broker Server

1. When EB is first installed, it's setup to use self-signed certificates. To replace the self-signed certificates, obtain your company's root CA certificate, and an intermediate certificate and key pair. Place them in /tmp with the following names:

```
/tmp/intermediate.cert.pem
```

```
/tmp/intermediate.key.pem
```

```
/tmp/ca.cert.pem
```

Use the following command to add them to Event Broker:

```
/opt/arc sight/kubernetes/scripts/arc sight-cert-util.sh write --re  
key=/tmp/intermediate.key.pem --re-crt=/tmp/intermediate.cert.pem --re-  
ca=/tmp/ca.cert.pem
```

**Note:** If EB was already deployed when running the above command, all Kafka broker nodes must be restarted. Either redeploy EB, or delete each Kafka pod. For example:

To list the pods:

```
kubectl get pods --all-namespaces
```

To delete the first Kafka pod:

```
kubectl delete pod eb-kafka-0 -n eventbroker1
```

Repeat for all the kafka pods.

2. 

```
export CA_CERT=/tmp/ca.cert.pem export  
INTERMEDIATE_CA_CERT=/tmp/intermediate.cert.pem export  
INTERMEDIATE_CA_KEY=/tmp/intermediate.key.pem  
export CERT_CA_TMP=/opt/fips_ca_tmp  
export EB=<event broker hostname>_<event broker port>
```
3. Create a temp location on the eb master server:  

```
mkdir $CERT_CA_TMP
```

## Step 3: On the Connector Server

Copy the `${STORES}/${EB}-cert-req` file (`%STORES%\%EB%-cert-req` on Windows platforms) from the connector to the Event Broker directory created above.

## Step 4: On the Event Broker Server

Create the signed certificate, for example:

```
/bin/openssl x509 -req -CA ${INTERMEDIATE_CA_CERT} -CAkey ${INTERMEDIATE_CA_KEY} -in ${EB} -cert-req -out ${EB} -cert-signed -days 365 -CAcreateserial -sha256
```

## Step 5: On the Connector Server

1. Copy the `${EB}-cert-signed` certificate from the Event Broker to the connector's `${STORES}` directory. (On the Windows platform, copy the `%EB%-cert-signed` certificate to the connector's `%STORES%` directory.)
2. Copy the `ca.cert.pem` certificate from the Event Broker to the connector's `${STORES}` directory. (On the Windows platform, copy the certificate to the `%STORES%` directory.)
3. Copy the `intermediate.cert.pem` certificate from the Event Broker to the connector's `${STORES}` directory. (On the Windows platform, copy the certificate to the `%STORES%` directory.)
4. Import the CA certificate to the trust store, for example:

```
jre/bin/keytool -importcert -file ${STORES}/${CA_CERT} -alias CARoot  
-keystore ${STORES}/${EB}.truststore.jks -storepass ${STORE_PASSWD}
```

**On Windows platforms:**

```
jre\bin\keytool -importcert -file %STORES%\${CA_CERT} -alias CARoot  
-keystore %STORES%\%EB%.truststore.jks -storepass %STORE_PASSWD%
```

5. Import the intermediate certificate to the trust store, for example:

```
jre/bin/keytool -importcert -file ${STORES}/intermediate.cert.pem -alias  
INTCARoot -keystore ${STORES}/${EB}.truststore.jks -storepass <trust store  
password>
```

**On Windows platforms:**

```
jre\bin\keytool -importcert -file %STORES%\intermediate.cert.pem -  
aliasINTCARoot -keystore %STORES%\%EB%.truststore.jks -storepass <trust  
store password>
```

6. When prompted, enter **yes** to trust the certificate.
7. Import the CA certificate to the key store, for example:

```
jre/bin/keytool -importcert -file ${STORES}/${CA_CERT} -alias CARoot -  
keystore ${STORES}/${EB}.keystore.jks -storepass ${STORE_PASSWD}
```

**On Windows platforms:**

```
jre\bin\keytool -importcert -file %STORES%\${CA_CERT} -alias CARoot -  
keystore %STORES%\%EB%.keystore.jks -storepass %STORE_PASSWD%
```

8. Import the intermediate certificate to the key store, for example:

```
jre/bin/keytool -importcert -file ${STORES}/intermediate.cert.pem -alias  
INTCARoot -keystore ${STORES}/${EB}.keystore.jks -storepass <key store  
password>
```

**On Windows platforms:**

```
jre\bin\keytool -importcert -file %STORES%\intermediate.cert.pem -alias  
INTCARoot -keystore %STORES%\%EB%.keystore.jks -storepass <key store  
password>
```

If successful, this command will return the message, *Certificate reply was installed in keystore.*

9. When prompted, enter **yes** to trust the certificate.
10. Import the signed certificate to the key store, for example:

```
jre/bin/keytool -importcert -file ${STORES}/${EB}-cert-signed -alias ${EB}  
-keystore ${STORES}/${EB}.keystore.jks -storepass ${STORE_PASSWD}
```

**On Windows platforms:**

```
jre\bin\keytool -importcert -file %STORES%\%EB%-cert-signed -alias %EB%  
-keystore %STORES%\%EB%.keystore.jks -storepass %STORE_PASSWD%
```

If successful, this command will return the message, Certificate reply was installed in keystore.

11. Note the key store and trust store paths:

```
echo ${STORES}/${EB}.truststore.jks  
echo ${STORES}/${EB}.keystore.jks
```

**On Windows platforms:**

```
echo %STORES%\%EB%.truststore.jks  
echo %STORES%\%EB%.keystore.jks
```

12. Navigate to the bin directory and run agent setup. Install a connector with Event Broker as the destination, for example:

```
cd <installation dir>/current/bin  
./runagentsetup.sh
```

**On Windows platforms:**

```
cd <installation dir>\current\bin  
runagentsetup.bat
```

- a. When completing the Event Broker destination fields, use the same values as in Step 8 for the path and password.
- b. Set **Use SSL/TLS** to **true**.
- c. Set **Use SSL/TLS Authentication** to **true**.

13. Cleanup. Delete the following files:

**Caution:** The following files should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Event Broker. These files are very sensitive and should not be distributed to other machines.

```
rm ${STORES}/intermediate.cert.pem  
rm ${STORES}/intermediate.key.pem  
rm ${STORES}/${EB}-cert-signed  
rm ${STORES}/${EB}-cert-req
```

**On Windows platforms:**

```
del %STORES%\intermediate.cert.pem  
del %STORES%\intermediate.key.pem  
del %STORES%\%EB%-cert-signed  
del %STORES%\%EB%-cert-req
```

## Step 6: On the Event Broker Server

To cleanup the Event Broker server, delete the temporary folder where the certificate was signed and the certificate and key files in /tmp.

**Caution:** The temporary folder should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Event Broker. These files are very sensitive and should not be distributed to other machines.

# Configure an Event Broker Destination without Client Authentication in FIPS Mode

Follow these steps to configure an Event Broker destination from the SmartConnector without client authentication in FIPS mode.

## On the Event Broker Server

When EB is first installed, it's setup to use self-signed certificates. To replace the self-signed certificates, obtain your company's root CA certificate, and an intermediate certificate and key pair. Place them in /tmp with the following names:

```
/tmp/intermediate.cert.pem
```

```
/tmp/intermediate.key.pem
```

```
/tmp/ca.cert.pem
```

Use the following command to add them to Event Broker:

```
/opt/arcsight/kubernetes/scripts/arcsight-cert-util.sh write --re-key=/tmp/intermediate.key.pem --re-crt=/tmp/intermediate.cert.pem --re-ca=/tmp/ca.cert.pem
```

**Note:** If EB was already deployed when running the above command, all Kafka broker nodes must be restarted. Either redeploy EB, or delete each Kafka pod. For example:

To list the pods:

```
kubectl get pods --all-namespaces
```

To delete the first Kafka pod:

```
kubectl delete pod eb-kafka-0 -n eventbroker1
```

Repeat for all the kafka pods.

## On the SmartConnector Server

1. Prepare the SmartConnector:

- **If the connector is not yet installed:** Run the installer. After core software has been installed, you will see a window that lets you select **Add a Connector** or **Select Global Parameters**. Check **Select Global Parameters**, and on the window displayed, select **Set FIPS mode**. Set to **Enabled**.

- **If the connector is already installed:** Run the installer. Select **Set Global Parameters** and set **Set FIPS Mode** to **Enabled**.

2. Navigate to the connector's current directory, for example:

```
cd <install dir>/current
```

3. Set the environment variables for the static values used by keytool, for example:

```
export CURRENT=<full path to this "current" folder>
export BC_OPTS="-storetype BCFKS -providertype BCFIPS -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath
${CURRENT}/lib/agent/fips/bc-fips-1.0.0.jar
-J-Djava.security.egd=file:/dev/urandom"
export EB=<event broker hostname>_<event broker port>
export STORES=${CURRENT}/user/agent/stores
export STORE_PASSWD=changeit
```

**On Windows platforms:**

```
set CURRENT=<full path to this "current" folder>
set BC_OPTS="-storetype BCFKS -providertype BCFIPS
-J-Djava.ext.dirs=%CURRENT%\jre\lib\ext;%CURRENT%\lib\agent\fips
-J-Djava.security.properties=%CURRENT%\user\agent\agent.security"
set EB=<event broker hostname>_<event broker port>
set STORES=%CURRENT%\user\agent\stores
set STORE_PASSWD=changeit
```

4. Create the user/agent/stores directory if does not already exist, for example:

```
mkdir ${STORES}
```

**On Windows platforms:**

```
mkdir %STORES%
```

5. Copy the ca.cert.pem and the intermediate.cert.pem certificates from the Event Broker to the connector's \${STORES} directory. (On the Windows platform, copy the certificate to the %STORES% directory.)

6. Import the CA certificate to the trust store, for example:

```
jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/ca.cert.pem -alias
CARoot -keystore ${STORES}/${EB}.truststore.bcfips -storepass ${STORE_
PASSWD}
```

**On Windows platforms:**

```
jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\ca.cert.pem -alias  
CARoot -keystore %STORES%\%EB%.truststore.bcfips -storepass %STORE_  
PASSWD%
```

7. Import the intermediate certificate to the trust store, for example: `jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/intermediate.cert.pem -alias INTCARoot -keystore ${STORES}/${EB}.truststore.bcfips -storepass <trust store password>`

8. When prompted, enter **yes** to trust the certificate.

9. Note the trust store path:

```
echo ${STORES}/${EB}.truststore.bcfips
```

**On Windows platforms:**

```
echo %STORES%\%EB%.truststore.bcfips
```

10. Navigate to the `bin` directory and run agent setup. Install a connector with Event Broker as the destination, for example:

```
cd <installation dir>/current/bin  
./runagentsetup.sh
```

**On Windows platforms:**

```
cd <installation dir>\current\bin  
runagentsetup.bat
```

- a. When completing the Event Broker destination fields, use the value from Step 7 for the trust store path and the password used in Step 6 for the trust store password.
  - b. Set **Use SSL/TLS** to **true**.
  - c. Set **Use SSL/TLS Authentication** to **false**.
11. Cleanup. Delete the certificate file, for example:

**Caution:** The following file should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Event Broker. These files are very sensitive and should not be distributed to other machines.

```
rm ${STORES}/ca.crt
```

**On Windows platforms:**

```
del %STORES%\ca.crt
```

# Send Documentation Feedback

If you have comments about this document, you can [contact the documentation team](#) by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

**Feedback on Configuring TLS between Event Broker and SmartConnectors (Event Broker and Connectors )**

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to [arc-doc@hpe.com](mailto:arc-doc@hpe.com).

We appreciate your feedback!