Micro Focus Security

ArcSight SmartConnectors

Format Preserving Encryption Environment Setup Guide

Document Release Date: October 22, 2018

Software Release Date: October 22, 2018

Legal Notices

Warranty

Restricted Rights Legend

Copyright Notice

Trademark Notices

Support

Contact Information

Phone	A list of phone numbers is available on the Micro Focus Security ArcSight Technical Support Page: https://softwaresupport.softwaregrp.com/support-contact-information
Support Web Site	https://softwaresupport.softwaregrp.com/
Protect 724 Community	https://community.softwaregrp.com/t5/ArcSight/ct-p/arcsight

Revision History

Date	Description
9/20/2018	Second revision: Added process to upgrade from Voltage Simple API Java 5.10 client to 5.20.
10/19/2017	First release of this technical note.

Contents

SmartConnectors with Format Preserving Encryption

Running a SmartConnector with Format Preserving Encryption enabled requires correct environment setup for the machine hosting the SmartConnector.

The setup primarily involves addressing two areas (1) ensuring proper connectivity between the connector machine and the SecureData appliance server and (2) ensuring the availability of an appliance certificate in all locations needed by the SmartConnector.

In addition, if you are not using Instant Connector Deployment with ArcSight Management Center, or you are upgrading a connector 7.6.0 or older to 7.7.0 or higher (via ArcSight Management Center or standalone), you will need to manually install the client for each connector host. Consult your SecureData Appliance documentation for instructions for your platform. Follow the instructions to verify that the connectivity is established from the SecureData Client to the SecureData Server and the test program runs as expected before proceeding with SmartConnector configuration for data encryption. Data encryption parameters are described in the configuration guide for each SmartConnector.

Network Connectivity

During connector configuration, during connector initialization at runtime, and also occasionally (even though not frequently) during events processing, the SmartConnector must connect to the SecureData appliance through the HTTPS protocol. Therefore, the HTTPS connectivity between the connector machine and the appliance server must be good.

The following sections describe steps to be followed to ensure connectivity for Linux and Windows operating systems.

Linux

- Edit the **`/etc/hosts`** file of the connector machine to include a line for the IP address and the hostname for the SecureData appliance. This step is required only if the DNS does not recognize the SecureData appliance FQDN.
- If your connector machine does not require a proxy to make an HTTPS connection to the SecureData appliance server, you do not need to set up a proxy. Verify that no global proxy is set for your machine. For a given user, you can open a fresh terminal, log in as the user, and enter **`env | grep -i proxy`** to determine whether a proxy has been inadvertently set by someone else.
- If your connector machine requires a proxy to make an HTTPS connection to the SecureData appliance server, set up a proper proxy. There is more than one way to set up a proxy. The System Administrator can choose the proper way to do it for the case in hand. However, keep in mind the following three important items:
  - If there are machines that do not require a proxy to be reached from the connector machine (such as ESM or Logger, for example), then the **`no_proxy/NO_PROXY`** environment variables must be set to bypass proxies for those hosts.
  - Make sure that the proxy-related variables are set globally, that is, for all users.
  - Verify that whatever proxy-related variables you have set are not transient (valid for one shell/terminal), and also verify they are actually in effect. You may need to reboot your machine to verify. Finally, on a fresh terminal, enter **`env | grep -i proxy`** to see all proxy-related environment variables. For a machine that has a number of hosts that require a proxy and a number of hosts that do not require a proxy, you should see something like:

```
no_proxy=localhost,.xxx.com,.aaa.bbb.com
NO_PROXY=localhost,.xxx.com,.aaa.bbb.com
https_proxy=dddd.ccccc.hh.com:8080
HTTPS_PROXY=dddd.ccccc.hh.com:8080
```

Windows

- Edit the `C:\Windows\System32\Drivers\etc\hosts` file of the connector machine to include a line for the IP address and the hostname for the SecureData appliance. This step is required only if the DNS does not recognize the SecureData appliance FQDN.
- If your connector machine does not require a proxy to make an https connection to the SecureData appliance server, do not set up a proxy. Verify that no global proxy is set for your machine. For a given user, you can open a fresh terminal, log in as the user, and enter `env | grep –i proxy` to determine whether a proxy has been inadvertently set by someone else.
- If your connector machine requires a proxy to reach certain hosts and does not require a proxy to reach some hosts, you can set these parameters from your browser's **Internet Options > Connections > LAN Settings >Advanced** tab. After setting the correct proxies and bypass list properly, open a command prompt as Administrator and enter the following command to import the Internet Options into the HTTP protocol connection:

```
netsh  winhttp  import  proxy  source=ie
```

Secure Data Appliance Certificates

To be able to make a successful connection and encrypt data, the server certificate must be present in all needed locations. These locations are highlighted in the following sections.

Windows

For SmartConnectors running in Windows, the server certificate must be in the following three locations. Perform these steps after installing the connector core software, but prior to selecting connector and adding parameter information.

- Connector certificate store - `../current/jre/lib/security/cacerts`. When using FIPS for the connector, the certificate must be placed into `..\current\user\agent\fips\bcfips_ks`

  You can import any certificate to this store by using the `../current/jre/bin/keytool` utility. For example, from the `$ARCSIGHT_HOME/current/bin` directory, execute the following command to import the certificate:
  `arcsight keytoolgui`

  Open the keystore in `$ARCSIGHT_HOME/jre/lib/security/cacerts` (the password will be `changeit`).

  From the **Menu** bar, select **Tools** and **Import Certificate**. Upload the certificate file.

  Trust the certificate.

- Trusted Root Certification Authority for the local computer. Use the Windows certificate import wizard to import the certificate.
- Trusted Root Certification Authority for the current user. Use the Windows certificate import wizard to import the certificate.

Finally, it is always advisable to have unchained server certificates. Windows certificates in trusted authority are subject to CRL verification and it may involve verifying the entire chain. The process, controlled by Windows, may require connecting outside the hosts to verify the chain of trust, resulting somewhat troublesome. If a component cannot be verified, Windows does not trust it and the connection to the SecureData appliance fails.

Linux

For SmartConnectors running on Linux, the server certificate must be in the following two locations. Perform these steps after installing the connector core software, but prior to selecting connector and adding parameter information.

- Connector certificate store - `../current/jre/lib/security/cacerts`. When using FIPS for the connector, the certificate must be placed into `..\current\user\agent\fips\bcfips_ks`

  You can import any certificate to this store using the `../current/jre/bin/keytool` utility. For example, from the `$ARCSIGHT_HOME/current/bin` directory, execute the following command to import the certificate:

  ```
  arcsight keytoolgui
  ```

  Open the keystore in `$ARCSIGHT_HOME/jre/lib/security/cacerts` (the password will be `changeit`).

  From the **Menu** bar, select **Tools** and **Import Certificate**. Upload the certificate file.

  Trust the certificate.

- The store used by the SecureData client. Assuming the client directory is `/opt/voltage`, the store would be `/opt/voltage/trustStore.` The certificate must be Base64 encoded, not DER; otherwise, the ./c_rehash command will fail. Copy the server certificate (in `*.pem` format) in the directory `/opt/voltage/trustStore`, and run the following command:

  ```
  /opt/voltage/trustStore/c_rehash .
  ```

  Note: Do not forget to type the `'.'`

Upgrade from the Voltage Simple API Java 5.10 client to 5.20

The following steps describe the process to upgrade from Voltage Simple API Java 5.10 client to 5.20.

1. Stop the connectors which are using the client you are going to upgrade.
2. For Linux machines: Delete the content of the current Voltage Client Installation Directory **(default: /opt/voltage).**
3. For Windows machines: Find the installer for the version that you want to uninstall and run it. Follow the wizard and select **Remove**, after the process is complete go to the installation folder and delete any remaining files or folders from the installation directory **(default: C:\voltage).**
4. Run the installer for the new version, choose exactly the same installation folder as the one from the previous client, if the installation folder is changed the connector will fail since it won't be able to find the components from the Voltage client.
5. On Linux, after installing the new version of the client, make sure you reimport the SecureData Server certificate in the client.
    1. Go to /opt/voltage/trustStore (assuming the installation directory for the Voltage Client is /opt/voltage) and copy the certificate with a *.pem extension.
    2. In the same folder, run the command: ./c_rehash .
       Note: Don't forget to type '.'. Keep in mind that the certificate must be Base64 encoded.
6. After the client is installed, and the connection to the Voltage server is available, the connector can be started. It should encrypt the data in the same way it did with the older version of the client.
   Note: The standalone installed connectors will expect the Voltage Client's install path to be **/opt/voltage** (Linux) or **C:\voltage** (Windows). If the connector was installed using the one-click functionality, users must be careful not to change the path were the connector will look for the Voltage Client.

For more information about the Voltage Simple API Java Client, please refer to the official Voltage Guides.