opentext™

ArcSight SmartConnectors

Software Version: CE 24.4

SmartConnector Installation and User Guide

Document Release Date: October 2024
Software Release Date: October 2024

Legal Notices

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Copyright Notice

Trademark Notices

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number
- Document Release Date, which changes each time the document is updated
- Software Release Date, which indicates the release date of this version of the software

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

https://www.microfocus.com/support-and-services/documentation

Contents

Overview of SmartConnectors

SmartConnectors intelligently collect a large amount of heterogenous raw event data from security devices in an enterprise network, process the data into ArcSight security events, and transport data to destination devices, which receives the event data from the connectors. The values such as severity, priority, and time zone are normalized into a common format and the data structure is normalized into a common schema. This allows you to find, sort, compare, and analyze all events using the same event fields.

SmartConnectors are built on a connector framework, which offers advanced features such as throttling, bandwidth management, caching, state persistence, filtering, encryption, and event enrichment, to ensure reliability, completeness, and security of log collection, while also optimizing the network usage.

The granular normalization of log data allows for the deterministic correlation that detects the latest threats including Advanced Persistent Threats and prepares data to be fed into machine learning models. SmartConnector technology supports over 400 different device types, such as routers, e-mail servers, anti-virus products, firewalls, intrusion detection systems (IDS), access control servers, VPN systems, anti-DoS appliances, operating system logs, and other sources that detect and report security or audit information.

SmartConnectors leverage ArcSight's industry-standard Common Event Format (CEF) for both OpenText and certified device vendors. This partner ecosystem keeps growing not only with the number of supported devices but also with the level of native adoption of CEF from device vendors.

SmartConnector Features

Connectors both receive and retrieve information from network devices. If the device sends information, the connector becomes a receiver. But, if the device does not send information, the connector can retrieve it.

SmartConnectors are also available to forward events between ArcSight systems such as Transformation Hub and ESM, enabling the creation of multi-tier monitoring and logging architectures for large organizations and Managed Service Providers.

Connectors perform the following tasks:

- Collect all the data from a source device, which eliminates the need to return to the device during an investigation or audit.
- Parse individual events and normalize event values such as severity, priority, and time zone into a common schema (format) for use by the ESM Manager.
- Filter out data that is not needed for analysis, thus saving network bandwidth and storage space (optional).
- Filter and aggregate events to reduce the volume sent to the Manager, ArcSight Logger, or other destinations, which reduces event processing time and increases efficiency of ArcSight.
- Categorize events by using a common, human-readable format, saving time, and making it easier to use the event categories to build filters, rules, reports, and data monitors.
- Add device and event information to it to complete the message and send it to the configured destination.
- Pass processed events to the ESM Manager.

After the connectors normalize and send events to the ESM Manager, the events are stored in the centralized ESM database. ESM then filters and cross-correlates these events with rules to generate meta-events. The meta-events then are automatically sent to administrators with corresponding Knowledge Base articles that contain information supporting their enterprise's policies and procedures.

Depending on the network device, some connectors can issue commands to devices. These actions can be executed manually or through automated actions from rules and some data monitors.

Specific connector configuration guides document device-to-ESM event mapping information for individual vendor devices, as well as specific installation parameters and configuration information.

Data collection

Connectors are specifically developed to work with network and security products by using multiple techniques such as simple log forwarding and parsing, direct installation on native devices, SNMP, and syslog.

The connectors support the following data collection and event reporting formats:

- Log File Readers (including text and log file)
- Syslog
- SNMP
- Database
- XML
- Proprietary protocols, such as OPSEC

The ArcSight ESM Console, ESM Manager, and connectors communicate using HTTP over Secure Sockets Layer (SSL also referred to as HTTPS).

Different connectors are available for the following types of vendor devices:

- Network and host-based IDS and IPS
- VPN, Firewall, router, and switch devices
- Vulnerability management and reporting systems
- Access and identity management
- Operating systems, Web servers, content delivery, log consolidators, and aggregators

For more information about the types of SmartConnectors, see "Types of SmartConnectors" on page 72.

Data encryption

Connectors provide SecureData format-preserving encryption to adhere to the regulatory requirement, which mandates that data leaving the connector machine to another destination must be encrypted. This feature is supported only on Linux and Windows 64-bit platforms. For more information about the format preserving parameters for connectors, refer to the Configuration Guide for the specific connector.

You can enable data encryption either during installation or while configuring a connector. You must provide the URL of the encryption server, the identity and shared secret configured for

SecureData, and the fields to be encrypted when configuring the connector. If a proxy is enabled for the machine, you need a proxy host and port for an HTTP connection.

Important:

- If you enable encryption, you cannot change any of the encryption parameters later. To change any parameters, you must reinstall the connector.
- To enable encryption on a connector that is already installed, use the wizard to select the **Modify Connector Parameters** option.
- In deployments where multiple connectors are chained or cascaded before reaching the destination, the encryption must only be enabled at the very first connector.
- Encryption of address fields including the IP addresses and MAC addresses are not supported.
- If the input data to be encrypted is in digits, then it must be at least three characters long.
- Additional data fields cannot be selected for encryption.
- For event data transfer, although the connector and the destination can be set to FIPS-compliant mode, if encryption is enabled, the communication between the connector and the secure server is not FIPS-compliant.
- Derived event fields cannot be chosen for encryption. If any of the derived fields need encryption, include the parent field for encryption.
- For optimum performance, the number of encrypted fields must be limited to 20.

Event filtering and aggregation

Filtering

You can add filter conditions to sort the events passed to the destination according to specific criteria during SmartConnector installation and configuration. For example, you can use filters to sort out events with certain characteristics, from specific network devices, or generated by vulnerability scanners. The events that do not meet the Connector filtering criteria are not forwarded.

To remove events that are not of interest or include only events that are of interest to your organization before they are ingested, you can use Customized Events Filtering.

For more information about configuring Filtering, see Managing SmartConnector Filter Conditions.

Aggregation

The Connector can be configured to aggregate (summarize and merge) events that have the same values in a specified set of fields, either for a specified number of times or within a specified time limit.

Connector aggregation compiles events with matching values into a single event. The aggregated event contains only the values that are common to events, and the earliest start time and latest end time. This reduces the number of individual events that must be evaluated. An event that repeats every 500 ms, for example, can be represented by a single event that is generated every 10 seconds, producing a 20:1 event compression. Individual connectors can be configured to aggregate events, thus reducing event traffic to the ESM Manager and the storage requirements in the ESM database.

For example, if the connector is configured to aggregate events with a certain Source IP and Port, Destination IP and Port, and Device Action whenever the events occur 10 times in 30 seconds. If 10 events with these matching values are received by the connector within that time frame, they are grouped into a single event with an aggregated event count of 10.

If the 30-seconds time frame expires and the connector receives only two matching events, the connector creates a single aggregated event with an aggregated event count of two. If 900 matching events are generated during 30 seconds, the connector creates 90 aggregated events, each with an aggregated event count of 10.

Firewalls are a good candidate for aggregation because of the volume of events with similar data coming in from multiple devices.

Unique Generator aa ID

Globally unique event ID (GEID) is an optional feature that can be enabled by updating certain parameters. Ideally, each event passing through an ArcSight product must be assigned a GEID.

The Generator ID is a value between 1 to 16383 and is used to create GEIDs in a sequential order that can register up to one million instances per second. Previous SmartConnector versions must be upgraded so that the events are properly assigned with GEIDs. GEIDs cannot be unassigned.

If you do not specify a value for Unique Generator ID:

- The GEID generated by the connector sets **zero** as the default value.
- The connector wizard displays a message, indicating that the Unique Generator ID has not been set.

- The **agent.log** file displays a message, indicating that the Unique Generator ID has not been set.
- When you create the **silent-properties** file, the value for the **containeroptionsconfig.agent.generator.id** property will be empty.
- Events will not be processed when **Amazon S3** is configured as one of the destinations or if **Recon** is selected as the value for the **Check Event Integrity Method** parameter for any destination.

Data mapping to vendor events

Connectors collect the vendor-specific event fields logged by a network device. Before these events are forwarded to their configured destination the events are mapped to the ArcSight data fields within the connector, based on the ArcSight ESM schema.

For specific mappings between the connector data fields and supported vendor-specific event definitions, see the configuration guide, available on SmartConnectors Grand List - (A-Z), for the device-specific connector. For example: for the SmartConnector for Cisco PIX/ASA Syslog mappings, see the Configuration Guide for Cisco PIX/ ASA Syslog SmartConnector.

General mappings for ArcSight Common Event Format connectors are documented in the Implementing ArcSight Common Event Format (CEF) guide.

FIPS compliance

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 mode is supported on local, and remote SmartConnectors.

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

FIPS Suite B

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.

FIPS compliant Connectors

The following connectors are FIPS compliant:

- All syslog connectors
- All file reader connectors
- All SNMP connectors
- Most database connectors (except Oracle Audit DB and when using SQL Server drivers with encryption)
- Cisco Secure IPS SDEE connectors
- Sourcefire Defense Center eStreamer connector
- Check Point OPSEC NG connector

FIPS non-compliant SmartConnectors

The following SmartConnectors are not FIPS compliant:

- Database connectors using SQL Server drivers with encryption
- Connectors using Oracle drivers

SmartConnectors Not certified as FIPS compliant

The following connectors are not certified as FIPS compliant:

- API connectors with proprietary internal mechanisms
- Web Services and Cloud connectors

Types of SmartConnectors

Depending on your requirement, you can select any of the following SmartConnector types:

- API Connectors
- Database Connectors

- File Connectors
- FlexConnectors
- Microsoft Windows Event Log Connectors
- Model Import Connectors
- Other connectors
- Scanner Connectors
- SNMP Connectors
- Syslog Connectors

API Connectors

API connectors use a standard or proprietary API to pull events from devices. In most cases, a certificate must be imported from the device to authenticate connector access to the device. There are also several configuration steps required on the device side. For more information, refer to the respective connector configuration guides.

Database Connectors

Database connectors support event collection from databases. They use SQL queries to periodically poll for events. Connectors support major database types, including MS SQL, MS Access, MySQL, Oracle, DB2, Postgres, and Sybase.

The database user must have adequate permission to access and read the database. For Audit database connectors, such as SQL Server Audit DB and Oracle Audit DB, system administrator permission is required.

Some database connectors such as the Microsoft SQL Server Multiple Instance DB connector support multiple database events. Connectors such as the connector for McAfee Vulnerability Manager DB collect events from scanner databases.

Note: Refer to FIPS Compliance Limitation to understand the limitations for some of the database SmartConnectors.

File Connectors

File connectors are normally installed on the device machine, but when the monitored files are accessible through network shares or NFS mounts, the connectors can be installed on remote machines as well.

Types of File Connectors:

- **Real Time**

  Real Time log file connectors read normal log files in which lines are separated by a new line character or fixed length records, in which a file consists of only one line but contain multiple records of fixed length.

  These connectors can continue to follow a log file that retains its name or changes its name based on the current date and other factors. Depending on the number of files monitored, Real Time connectors can be of type that monitors a single log file or of type that monitors multiple log files.

- **Folder Follower**

  Folder follower connectors monitor files copied to a folder. There are connectors that monitor a single log file in a folder and connectors that monitor log files recursively.

  Depending on the device type, connectors support **.txt** and **.xml** file types. Most of the scanner file connectors, such as Nessus, and NeXpose are in **.xml** format.

  The type of log file connector is not usually part of the connector name unless both types of connector exist for a particular device.

  Some connectors require a trigger file to let the connector know when the file is complete and ready for processing. This file typically has the same file name with a different extension. Files are renamed by default to increments such as **.processed**, **.processed.1**, and so on.

FlexConnectors

FlexConnectors allow you to create custom connectors that can read and parse information from third-party devices and map that information to the ArcSight event schema. When creating a custom connector, you define a set of properties (a configuration file) that identify the format of the log file or other source that is imported into the ESM Manager or Logger.

The FlexConnector framework is a software development kit (SDK) that lets you create a connector tailored to the devices on your network and their specific event data. For more information about FlexConnectors and how to use them, see the FlexConnector Developer's Guide.

Microsoft Windows Event Log Connectors

Microsoft Windows Event Log Connectors connect to local or remote Windows machines inside a single domain or in multiple domains, to retrieve and process security and system events.

System administrators use Windows Event Log to troubleshoot errors. Each entry in the event log contains information related to the severity of Error, Warning, Information, and Success Audit or Failure Audit messages.

There are following types of default Windows Event Logs:

- Application log, which tracks events that occur in a registered application.
- Security log, which tracks security changes and possible breaches in security.
- System log, which tracks system events.

The following connectors are available for Microsoft Windows Event Log:

- SmartConnector for Microsoft Windows Event Log
- SmartConnector for Microsoft Windows Event Log – Native

  For more information about the Native connector, see the configuration guide for the SmartConnector for Microsoft Windows Event Log - Native.

  For mappings, see SmartConnector for Microsoft Windows Event Log Native Windows Security Event Mappings document.

These connectors provide support for partial event parsing based on the Windows event header for all System and Application events. It also provides support for a FlexConnector-like framework that lets users create and deploy their parsers to parse event description for all System and Application events.

Some individual Windows Event Log applications are supported by the connectors for Microsoft Windows Event Log, for which Windows Event Log application or system support has been developed. See the configuration guides for specific connectors for a list of application and system events supported.

Model Import Connectors

Rather than collecting and forwarding events from devices, Model Import Connectors import user data from an Identity Management system into ArcSight ESM. For more information, see the individual configuration guides for Model Import Connectors on ArcSight Enterprise Security Manager (ESM) Documentation.

Model Import Connectors extract the user identity information from the database and populate the following lists in ESM with the data:

- Identity Roles Session List
- Identity Information Session List
- Account-to-Identity Map Active List

These lists are populated dynamically, which means that, as the identity data changes in the Identity Manager, the data in the lists are updated when you refresh the session list.

Other Connectors

Connectors that Use Multiple Mechanisms

Some connectors use multiple mechanisms. For example, the connector for Oracle Audit Database monitors both the database tables and audit files.

Connectors that Use TCP in Special Formats

Examples of connectors that use TCP in special formats are :

- **IP NetFlow (NetFlow/J-Flow):** Retrieves data over TCP in a Cisco-defined binary format.
- **ArcSight Streaming Connector:** Retrieves data over TCP from Logger in an ArcSight-proprietary format.

Scanner Connectors

There are two types of scanner connectors, those whose results are retained within a file, and those retrieved from a database.

Results for XML scanner connectors are retained in a file, making them log file connectors. Other scanners deposit their scanned events in a database and are treated as database connectors, and require the installation parameters used by the database connectors.

Scan reports are converted into base events, which for ESM destinations, can be viewed on the Console. The aggregated meta events are not displayed in the Console. Meta events create assets, asset categories, open ports, and vulnerabilities on the Console.

Scanner connectors can run in either of the following modes:

- **Interactive mode**

  In the Interactive mode, a graphical user interface shows the reports or log files available for import from the configured log directory. You can select the reports to send to the connector by selecting the **Send for individual log files** check box and clicking **Send to ArcSight**.
- **Automatic mode**

  Automatic mode is used in conjunction with an automated procedure to periodically run scans. The procedure, or shell script, must execute the scanner periodically and save a

report in **.cef** format. After the scan completes and the report is saved, an empty file called **<reportname>.cef_ready** must be created, which indicates to the connector that the **.cef** report is ready for importing. The connector continues to search for **.cef_ready** files and processes the corresponding **.cef** reports. The processed reports are renamed to **<original report file>.cef_processed**.

Parameter values required for scanner installation depends on whether you are installing a file or a database connector. File connectors require the absolute path to and name of the log file is required.

SNMP Connectors

SNMP Traps contain variable bindings, each of which holds a different piece of information for the event. They are usually sent over UDP to port 162, although the port can be changed.

SNMP connectors listen on port 162 by default or any other configured port and process the received traps. They can receive multiple trap types from the device but process traps only from one device with a unique Enterprise object identifier (OID).

SNMP is based on UDP, so there is a minor possibility of events being lost over the network.

Although there are several SNMP connectors for individual connectors, most SNMP support is provided by the SmartConnector for SNMP Unified. Parsers use the knowledge of the Management Information Base (MIB) to map the event fields, but, unlike some other SNMP-based applications, the connector itself does not require the MIB to be loaded.

Syslog Connectors

Syslog messages are free-form log messages prefixed with a Syslog header consisting of a numerical code (facility + severity), timestamp, and host name. Unlike file connectors, a Syslog connector can receive and process events from multiple devices. There is a unique regular expression that identifies the device.

TCP is a supported protocol for Syslog connectors. If UDP is used, there might be a possibility of missing Syslog messages over the network.

Depending on the mechanism with which the device logs are made available to the smartconnector, select the type of smartconnector to install:

- **Syslog Deamon SmartConnector** or **Syslog NG Deamon SmartConnector** - If the device writes logs to a port.
- **Syslog File SmartConnector** - If the device writes the log to a pipe or if the device writes log to a file.

SmartConnector Types	Available Parsers
• **Syslog Deamon:** The Syslog Daemon SmartConnector is a syslogd-compatible daemon designed to work in operating systems that have no syslog daemon in their default configuration, such as Microsoft Windows. They listen for Syslog messages on a configurable port, using port 514 by default. The default protocol is UDP, but other protocols such as Raw TCP are also supported. It is the only Syslog option supported for Windows platforms. • **Syslog File:** Supports the following types of logs: ◦ **Logs written to Pipe** When there is an existing syslog daemon syslogd is configured to write to a named pipe, and the SmartConnector reads from it to receive events. They require syslog configuration to send messages with a certain Syslog facility and severity. It is especially useful when storage is a factor. The Solaris platform tends to under-perform when using Syslog Pipe connectors. The operating system requires	AirMagnet Enterprise Syslog

SmartConnector Types	Available Parsers
	Apache HTTP Server Syslog

SmartConnector Types	Available Parsers
	Arbor Networks Peakflow Syslog

SmartConnector Types	Available Parsers
	ArcSight Common Event Format Syslog

SmartConnector Types	Available Parsers
	Barracuda Email Security Gateway Syslog

SmartConnector Types	Available Parsers
	Barracuda Firewall NG F-Series Syslog

SmartConnector Types	Available Parsers
	Barracuda Web Appliance Firewall Syslog

SmartConnector Types	Available Parsers
	Blue Coat Proxy SG Syslog

SmartConnector Types	Available Parsers
	BroadWeb NetKeeper Syslog

SmartConnector Types	Available Parsers
	Brocade BigIron Syslog

SmartConnector Types	Available Parsers
	Check Point Syslog

SmartConnector Types	Available Parsers
	Cisco ASA Syslog

SmartConnector Types	Available Parsers
	Cisco Catalyst OS Syslog

SmartConnector Types	Available Parsers
	Cisco IOS Syslog

SmartConnector Types	Available Parsers
	Cisco IronPort Email Security Appliance Syslog

SmartConnector Types	Available Parsers
	Cisco IronPort Web Security Appliance Syslog

SmartConnector Types	Available Parsers
	Cisco ISE Syslog

SmartConnector Types	Available Parsers
	Cisco Meraki Syslog

SmartConnector Types	Available Parsers
	Cisco Mobility Services Engine Syslog

SmartConnector Types	Available Parsers
	Cisco NX-OS Syslog

SmartConnector Types	Available Parsers
	Cisco Secure ACS Syslog

SmartConnector Types	Available Parsers
	Cisco Wireless LAN Controller Syslog

SmartConnector Types	Available Parsers
	Citrix NetScaler Syslog

SmartConnector Types	Available Parsers
	Dell SonicWALL Firewall Syslog

SmartConnector Types	Available Parsers
	F5 BIG-IP Syslog

SmartConnector Types	Available Parsers
	Fortinet Fortigate Syslog

SmartConnector Types	Available Parsers
	HoneyD Syslog

SmartConnector Types	Available Parsers
	HPE Aruba Mobility Controller Syslog

SmartConnector Types	Available Parsers
	HPE c7000 Virtual Connect Module Syslog
	HPE H3C Syslog
	HPE Integrated Lights-Out Syslog
	HP Printers Syslog
	HPE ProCurve Syslog
	HPE-UX Syslog
	IBM AIX Audit Syslog
	IBM Security Access Manager Syslog
	Infoblox NIOS Syslog
	Ingrian DataSecure Syslog
	Intersect Alliance SNARE Syslog
	ISC Bind Syslog
	ISC DHCP Syslog
	Juniper Firewall ScreenOS Syslog
	Juniper IDP Series Syslog
	Juniper JUNOS Syslog
	Juniper Network and Security Management Syslog
	Linux Audit Syslog
	McAfee Email Gateway Syslog
	McAfee Firewall Enterprise Syslog
	McAfee Network Security Manager Syslog
	McAfee Web Gateway Syslog
	Microsoft IIS Syslog
	NetApp Filer Syslog
	Netscout Arbor Security Syslog
	NitroSecurity Syslog
	Nortel Contivity Switch (VPN) Syslog
	Oracle Audit Syslog
	Oracle Solaris Basic Security Module Syslog
	Proofpoint Enterprise Protect and Enterprise Privacy Syslog
	Pulse Secure Pulse Connect Secure Syslog

SmartConnector Types	Available Parsers
	Radware DefensePro Syslog
	Sabernet NT Syslog
	Sendmail Syslog
	Snort Syslog
	Symantec Endpoint Protection Syslog
	Symantec Messaging Gateway Syslog
	TippingPoint SMS Syslog
	TippingPoint SMS Syslog Extended
	Top Layer Attack Mitigator Syslog
	Type80 SMA_RT Syslog
	UNIX OS Syslog
	VarySys PacketAlarm IPS Syslog
	VMware ESXi Server Syslog
	Vormetric CoreGuard Syslog

Other Syslog connectors are:

Raw Syslog: They are always used with the Raw Syslog destination. Raw Syslog connectors generally do not parse events. But, they take the Syslog string and copy it in the rawEvent field as-is . The Raw Syslog destination type takes the **rawEvent** field and sends it as-is by using UDP, Raw TCP, or TLS protocol, that is selected. The event flow is streamlined to eliminate components that do not add value. For example, with the Raw Syslog transport, the category fields in the event are ignored, so the categorization components are skipped. If you are transporting data to ArcSight Logger, you can use specific configuration parameters to provide minimal normalization of the Syslog data (for source and timestamp).

ArcSight CEF CISCO FireSight Syslog: Retrieves events and payload information from FireSIGHT DB by using the event ID and Sensor Name as input.

ArcSight CEF Encrypted Syslog UDP: Allows connector-to-connector communication through an encrypted channel by decrypting events previously encrypted through the CEF Encrypted Syslog (UDP) destination. The CEF connector lets ESM connect to aggregate, filter, correlate, and analyze events from applications and devices that deliver their logs in the CEF standard, by using the Syslog transport protocol.

UNIX supports all types of Syslog connectors. If a syslog process is already running, you can end the process or run the connector on a different port. The connector for UNIX OS Syslog provides the base parser for all Syslog sub-connectors.

For Syslog connector deployment information, see the connector Configuration Guide for UNIX OS Syslog. For device-specific configuration information and field mappings, see the connector configuration guide for the specific device. Each Syslog sub-connector has its own configuration guide.

Types of destinations

You can configure a connector to send events to one or more destinations. A destination is a Manager or device that can receive events from a connector. In addition to the selections configured during connector configuration, events can also be sent to additional or failover destinations.

Depending on your requirement, you can select any of the following destinations:

ArcSight Manager (encrypted)

If SmartConnectors are configured to use ArcSight Manager as a destination, they send events to the ESM Manger.

When connectors send events to ESM Manager, it stores the events in a relational database, processes them using its correlation engine, and makes them visible to the Console or Web interfaces. This is the commonly destination used.

For more information about the parameters to be selected during installation, see ArcSight Manager Parameters.

ArcSight Logger SmartMessage (encrypted)

Logger is a log management solution that is optimized for extremely high event throughput. Logger logs or stores time-stamped text messages, called events, at high sustained input rates. Events consist of a receipt time, a source (host name or IP address), and an un-parsed message portion. Logger compresses raw data, but also can retrieve it in an unmodified form for forensics-quality litigation reporting. Unlike ESM, Logger does not normalize events.

CIf SmartConnectors are configured to use ArcSight Logger SmartMessage as a destination, they send CEF events to Logger using an encrypted, optionally compressed channel called SmartMessage. Logger also can receive CEF syslog events from connectors.

To subscribe event data from a specific SmartConnector, do the following:

- Configure all the SmartConnectors to publish events to the same topic. Configure the Logger's Transformation Hub receiver to subscribe to this event topic. For more information, see the Administrator's Guide for Transformation Hub.
- Configure each SmartConnectors to publish events to different topics and then configure the Transformation Hub receiver on the Logger to subscribe to multiple event topics.

For more information about the parameters to be selected during installation, see ArcSight Logger SmartMessage Parameters.

You can also configure the SmartMessage transport to be persistent to achieve higher throughput for Logger destinations. For more information, see Configuring Persistent SmartMessage Transport.

ArcSight Logger SmartMessage Pool (encrypted)

You can specify a pool of logger devices as a single destination while the events are distributed among the loggers in the pool. Each batch of events processed by the connector is sent to the next logger in the pool in a round-robin fashion. You can either add the pool members one by one or use a CSV file that contains the predefined information for logger secure pool. You can also export and save the data entered in the panel into a CSV file.

For more information about the parameters to be selected during installation, see ArcSight Logger SmartMessage Pool Parameters.

Related Topics:

- Configuring Persistent SmartMessage Transport
- ArcSight Logger SmartMessage Pool (encrypted) Destination Parameters

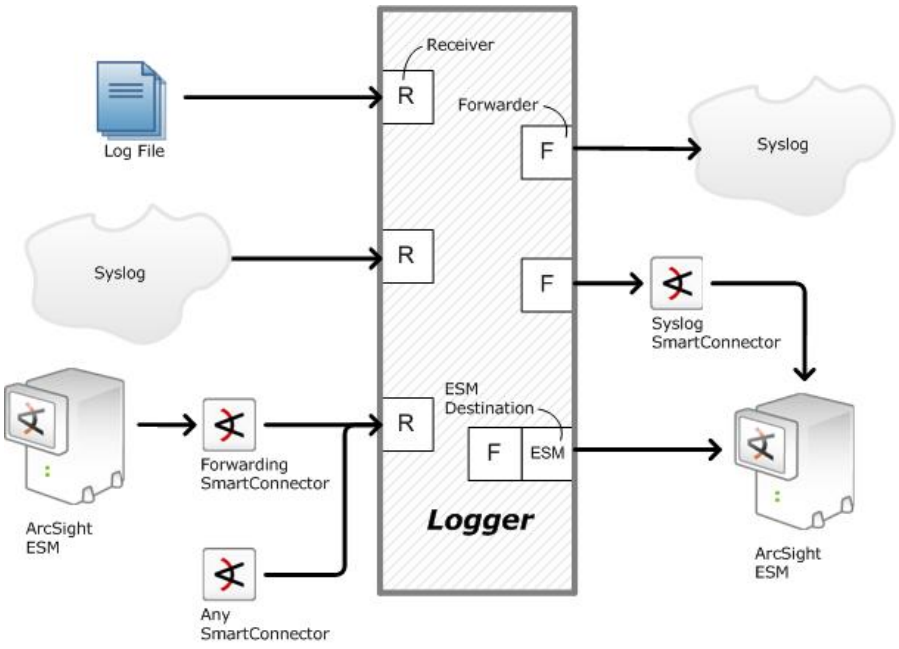
Sending Events from Logger to a Manager

Logger's most basic function is to store a large volume of security events. Logger can send a subset of these events to a Manager. It sends syslog or ArcSight Common Event Format (CEF) events directly to ESM through a built-in Connector called an ESM Destination. An ESM Destination appears as a Connector on a Console. For more information about ESM Destinations, see the *ArcSight Logger Administrator's Guide*.

SmartMessage is ArcSight technology used by Logger to provide a secure channel between Connectors and Logger. SmartMessage provides an end-to-end encrypted secure channel. At one end is a Connector, receiving events from the devices it supports; on the other end is SmartMessage Receiver on Logger.

Note: Use Syslog connector to forward events from Logger to ESM. If a different method such as Netcat is used, the events are forwarded to Logger, but not to ESM.

Logger Receivers (R) and Forwarders (F)



Note: The SmartMessage secure channel uses HTTPS (secure sockets layer protocol) to send encrypted events to Logger. This is similar to, but different from, the encrypted binary protocol used between Connectors and the ESM Manager.

Use port 443 (rather than ArcSight traditional port 8443) because the secure channel uses HTTPS.

Sending Events to Both Logger and a Manager

1. Set up the SmartMessage Receiver on Logger (see the configuration guide for the connector being installed).
2. Install the connector component (see the Connector Configuration Guide for your device).
3. Register the connector with an active ESM Manager and test that the connector is up and running.
4. Using the `$ARCSIGHT_HOME\current\bin\runagentsetup` script, restart the connector configuration program.
5. Select **Add, modify, or remove destinations** and click **Next**.
6. Select **Add destination** and click **Next**.
7. Select **ArcSight Logger SmartMessage (encrypted)** and click **Next**.
8. Enter the destination parameters and click **Next**:

Parameter	Description
Host Name/IP	The destination host name or IP address.
Port	The destination port 443 for Logger Appliance or 9000 for Software Logger.
Receiver Name	The destination receiver name. This setting should match the Receiver name you created in step 1 so that Logger can listen to events from this Connector.
Compression Mode	Select the option to enable or disable data compression. Default is **Disabled**.

9. If you have not already imported the certificate, the Logger certificate message is displayed to import the certificate to connector.
10. Select the **Import the certificate to connector from destination** option and click **Next**.
11. Navigate through the subsequent windows until you receive a message that confirms the configuration was successful. Click **Exit**, then click **Next** to exit the wizard.
12. Restart the connector for changes to take effect.

Sending Events to Logger

1. Set up the SmartMessage Receiver on Logger (see the *ArcSight Logger Administrator's Guide* for detailed instructions).
2. Install the connector component (see the Connector Configuration Guide for your device).
3. Using the `$ARCSIGHT_HOME\current\bin\runagentsetup` script, restart the connector configuration program.
4. Navigate through the windows, select **ArcSight Logger SmartMessage (encrypted)**, and then click **Next**.
5. Enter the destination parameter details and click **Next**.

Parameter	Description
Host Name/IP	The destination host name or IP address.
Port	The destination port 443 for Logger Appliance or 9000 for Software Logger.

Parameter	Description
Receiver Name	The destination receiver name. This setting should match the Receiver name you created in step 1 so that Logger can listen to events from this Connector.
Compression Mode	Select the option to enable or disable data compression. Default is **Disabled**.
CEF Version	Select any of the following options from the drop-down menu: • **0.1** - The Device Address, Source Address, Destination Address, and Agent Address fields will always be IPv4 or will be omitted. If there are any IPv6 addresses, they will be in the **Device Custom IPv6 Address** fields. The **Bytes In** and **Bytes Out** fields are limited to the size of an integer (up to 2^31-1). **Note**: Select **0.1** if you are not sure the chosen destination can handle CEF 1.0, which supports both IPv4 and IPv6modes. • **1.0** - Any of the address fields can be either IPv4 or IPv6 and the **Bytes In** and **Bytes Out** fields can be long values (up to 2^63-1).

6. If you have not already imported the certificate, the Logger certificate message is displayed to import the certificate to connector.
7. Select the **Import the certificate to connector from destination** option and click **Next**.
8. Navigate through the subsequent windows until you receive a message that confirms the configuration was successful. Click **Exit** and click **Next** to exit the wizard.
9. Restart the connector for changes to take effect.

Forwarding Events from ESM to Logger

The ArcSight Forwarding Connector can read events from an ESM Manager and forward them to Logger using ArcSight's Common Event Format (CEF).

Note: The Forwarding Connector is a separate installable file, named similarly to this: `ArcSight-6.x.x.<build>.x-SuperConnector-<platform>.exe`.

Use Forwarding Connector build 4810 or later for compatibility with Logger 1.5 or later.

1. Install the connector component (see the Connector Configuration Guide for your device).
2. Using the `$ARCSIGHT_HOME\current\bin\runagentsetup` script, restart the connector configuration program.
3. Navigate through the windows, select **ArcSight Logger SmartMessage (encrypted)**, and then click **Next**.
4. Enter the destination parameter details and click **Next**.

Parameter	Description
Host Name/IP	The destination host name or IP address.
Port	The destination port 443 for Logger Appliance or 9000 for Software Logger.
Receiver Name	The destination receiver name. This setting should match the Receiver name you created in step 1 so that Logger can listen to events from this Connector.
Compression Mode	Select the option to enable or disable data compression. Default is **Disabled**.

5. If you have not already imported the certificate, the Logger certificate message is displayed to import the certificate to connector.
6. Select the **Import the certificate to connector from destination** option and click **Next**.
7. Navigate through the subsequent windows until you receive a message that confirms the configuration was successful. Click **Exit** and click **Next** to exit the wizard.
8. Restart the connector for changes to take effect.

To configure the Forwarding Connector to send CEF output to Logger and send events to another Manager at the same time, see Sending Events to Both Logger and ESM.

ArcSight SaaS

If **ArcSight SaaS** is configured as a destination, all security events are sent in **Avro** format to Amazon MSK that is managed by ArcSight's SaaS offering.

For more information about the destination parameters to be selected during installation, see ArcSight SaaS.

It is mandatory that Admin must always get the registration URL before starting the connector install.

The registration URL for the **ArcSight SaaS** destination can be used only once. You can neither add failover destination for the **ArcSight SaaS** destination, nor modify the destination parameters.

When the access is revoked, events are no longer sent to Amazon MSK. A message indicating the same will be displayed in the logs. If you need to send events, then you must re-register the **ArcSight SaaS** destination with a new registration URL. For more information, see "Re-registering a destination" on page 168.

Note: If you re-register the **ArcSight SaaS** destination, all cached events in the connector will be lost. For more information, see "Events are not sent from SmartConnector to ArcSight SaaS" on page 228.

Transformation Hub

If SmartConnectors are configured to use Transformation Hub as a destination, they send events to Transformation Hub's Kafka cluster, from where the events are further distributed to real-time analysis and data warehousing systems.

The Transformation Hub destination is used to send events to a Transformation Hub cluster in in Avro, binary, or CEF format, which can then further distribute events to real-time analysis and data warehousing systems. Any application that supports retrieving data from Transformation Hub can receive these events (for example, ESM, ArcSight Investigate, Hadoop and Logger).

The SmartConnector Acknowledgments ("acks") ensure that Transformation Hub received the event before the SmartConnector removes it from its local queue. Acknowledgments do not indicate that consumers, such as Logger, have received the event data, only that Transformation Hub itself has. You can disable acknowledgments, enable to receive acknowledgment only from the primary replica, or enable every replica to acknowledge the event.

Supported SmartConnector versions encode their own IP address as meta-data in the Kafka message for consumers that require that information such as Logger Device Groups.

For instructions about setting up FIPS with Transformation Hub and SmartConnectors, see Configuring Security Mode for Smart Connectors with Transformation Hub Destinations.

For the content format Avro:

The SmartConnector uses Avro schema to emit the Avro output. Avro schema resides in the Schema Registry of Transformation Hub. The SmartConnector makes an HTTPS call to Transformation Hub to get and save the schema in its `user/agent` folder. The SmartConnector captures the Schema Registry details during the installation and fetches schema during its every restart.

Ensure that you use the compatible version of SmartConnector with Transformation Hub in order to emit Avro output as follows:

SmartConnector Version	Default Avro Schema Version	Transformation Hub Version
8.4	1.2.0	3.6 and 3.6.1

8.3	1.2.0	3.6
8.2	1.1.1	3.5

Note: You must install or upgrade Transformation Hub before upgrading SmartConnector.

To use a SmartConnector with the non-compatible version of Transformation Hub, perform the following steps after installing the SmartConnector:

1. Open the `$ARCSIGHT_HOME/current/user/agent/agent.properties` file.
2. Modify the **schema.registry.schema.version** parameter value to the required schema version. The currently supported versions are: 1.1.1 and 1.2.0

   For example: For SmartConnector 8.4 to work with Transformation Hub 3.5, set the property value to 1.1.1 as follows: **schema.registry.schema.version=1.1.1**
3. Restart the SmartConnector.

For the Content Types CEF 0.1 and CEF 1.0:

The key is sent on events with the Connectors IP address and a flag. The flag format is a single byte value. For ESM, the key is the agent ID.

The key format is: one byte flags + (4 or 16 bytes) IP (v 4 or v 6) address. Based on the value of the IP version bit, 4 or 16 additional bytes should be examined. This is used in case the key is made longer in a non-breaking fashion in the future.

Bit position	Meaning
0	IP version: 0 = IPv4 1 = IPv6
1	Key version: **Must be 0**. If there are future versions of key that are not backward compatible with this definition, it changes to 1.
2-7	Key version: **Must be 0**. Reserved for future.

For CEF 0.1 and 1.0, the events are delivered to Transformation Hub in their own messages, which are distributed to the partitions of the topic defined in Transformation Hub in a round-robin manner. For ESM, the events are sent in batches in a binary format. TLS encryption is supported, as is client certificate authentication.

When TLS is enabled by setting the **Kafka Broker on SSL/TLS** parameter to **true** during destination configuration, a Java KeyStore-format (.jks) file containing the certificates of the Transformation Hub's Kafka cluster, or a certificate that has signed them, will be required. The

location of this Trust Store file will be required during destination configuration. See Kafka documentation at https://kafka.apache.org/documentation.html#security_ssl for instructions.

Also, when client certificate authentication is enabled by setting the **Use SSL/TLS Client Authentication** parameter to **true**, a .jks file containing the private key and certificate to use must be provided. The Transformation Hub cluster must have the certificate (or a certificate that has signed it) in its trust store. The location of the keystore file and authentication information is to be provided in the **SSL/TLS Keystore File Path**, **SSL/TLS Keystore Password**, and **SSL/TLS Key Password** parameters. The key and keystore passwords are created when you set up Transformation Hub.

For more information about the parameters to be selected during installation, see Transformation Hub Parameters.

Amazon MSK

If **Amazon MSK** is configured as a destination, connectors will ingest events into the Amazon MSK server. The connector generates **Avro** output by using static Avro schema which is bundled with the Connector package. For more information about schema, refer to Avro Documentation.

For more information about the destination parameters to be selected during installation, see Amazon MSK.

Amazon S3

If SmartConnectors are configured to use **Amazon S3** (Amazon Simple Storage Service) as a destination, they send security events in the Avro format to Amazon S3. The Connector generates Avro output by using static Avro schema which is bundled with the Connector package. The Avro output is generated in the snappy compressed format. The TLSv1.2 protocol is used to secure file upload to S3 bucket. For more information, refer to Avro Documentation.

The **Amazon S3** destination is also supported for all the cloud-native Connectors, such as AWS Security Hub, AWS CloudWatch, and Azure Event Hub.

For more information about the parameters to be selected during installation, see Amazon S3 Parameters.

Microsoft Azure Event Hub

If SmartConnectors are configured to use **Microsoft Azure Event Hub** as a destination, they send events in Common Event Format (CEF) through a Kafka broker to Microsoft Azure Event Hub.

Note: Event Hub must enable a Kafka endpoint.

Azure Event Hub requires SSL or TLS for communication purposes and uses Shared Access Signatures (SAS) for authentication. In the same way, this requirement must be met for a Kafka endpoint within Event Hubs. To be compatible with Kafka, Event Hub uses SASL PLAIN for authentication and SASL SSL for transport security.

For more information about the parameters to be selected during installation, see Microsoft Azure Event Hub Parameters

CEF File

The Common Event Format (CEF) is an open log management standard that improves the interoperability of security-related information from different security and network devices and applications. The CEF destination allows you to capture security events in a CEF file rather than forwarding them to a Manager.

For more information about the parameters to be selected during installation, see CEF File Parameters.

CEF Syslog

If SmartConnectors are configured to use **CEF Syslog** as a destination, they send events in CEF (converted to bytes using the UTF-8 character encoding), by using UDP, TCP, or TLS protocol.

The TCP and UDP protocols can be used to send events to Logger, where data is received using a TCP or UDP Receiver. One receiver can receive events from more than one connector. The protocols can also be used to send events to a Syslog Daemon connector or non-ArcSight syslog receivers.

The TLS protocol sends events through a secure channel (an option that does not apply to Logger), and allows for one-way or two-way authentication. This data can be received by any application that supports TLS syslog reception, which includes ArcSight's Syslog NG Daemon connector.

For more details about the Syslog NG Connector, see the SmartConnector for Syslog NG Daemon.

For more information about the parameters to be selected during installation, see CEF Syslog Parameters.

CEF Encrypted Syslog (UDP)

If SmartConnectors are configured to use **CEF Encrypted (UDP)** as a destination, they send events in Common Event Format (CEF) using the UDP protocol, providing symmetric-key encryption. This option allows for a “Shared Secret” key that requires configuration to encrypt the data. This data can be decrypted on the receiver side by the CEF Encrypted Syslog (UDP) connector.

To decrypt the data on the receiving side, ensure that you have installed and configured the ArcSight CEF Encrypted Syslog (UDP) connector.

For more information about installing the connector and decrypting the data, see the SmartConnector for ArcSight CEF Encrypted Syslog (UDP) documentation.

For more information about the parameters to be selected during installation, see CEF Encrypted Syslog (UDP)

CSV File

Use this destination to capture events that a connector sends to ESM Manager into a CSV file. Typical ArcSight configurations do not require the use of external files to communicate events to the ESM Manager.

Event data is written to a file in Excel-compatible comma-separated values (CSV) format, with comments prefixed by ‘#.’ A connector can be configured to preface the data with a comment line that describes the fields found on a subsequent line.

Event data is written to files in the specified folder and can be configured to rotate periodically.

Following are the contents of an example event file:

#event.eventName,event.attackerAddress,event.targetAddress

"Port scan detected","1.1.1.1","2.2.2.2"

"Worm ""Code red"" detected","1.1.1.1","2.2.2.2"

"SQL Slammer detected","1.1.1.1","2.2.2.2"

"Email virus detected","1.1.1.1","2.2.2.2"

Rotating Event Data

Events are appended to the current file until the rotation time interval expires, at which time a new current file is created and the previous current file is renamed. One hour is a typical

rotation time interval.

Event files are named using the time stamp of their creation, and all files, except for the current file, have the text '.done.csv' appended. For example, a typical CSV file set configured to rotate every hour might consist of files named as follows:

```
2007-01-28-10-55-33.csv
```

```
2007-01-28-09-55-33.csv.done
```

```
2007-01-28-08-55-33.csv.done
```

Using the properties file, the configuration of your CSV Connector can be customized to filter and aggregate events as desired.

A Connector can also be configured to send events to a CSV file and an ESM Manager at the same time.

For more information about the parameters to be selected during installation, see CSV File Parameters

Raw Syslog

Although normalized data is faster and easier to parse and access, many IT professionals prefer having the raw data available for review, forensics, and litigation. This destination sends raw syslog events through the UDP, TCP, or TLS protocol.

It works in conjunction with the Raw Syslog connector, which captures raw, unparsed security events for further processing. If you are transporting data to ArcSight Logger, you can use specific configuration parameters to provide minimal normalization of the syslog data (for source and timestamp). For more information, see the *SmartConnector for Raw Syslog Daemon Configuration* Guide.

Note: Connections to Qualys Cloud Platform require TLS 1.1 or higher.

For more information about the parameters to be selected during installation, see Raw Syslog Parameters.

Overview of SmartConnectors

SmartConnectors intelligently collect a large amount of heterogenous raw event data from security devices in an enterprise network, process the data into ArcSight security events, and transport data to destination devices, which receives the event data from the connectors. The values such as severity, priority, and time zone are normalized into a common format and the data structure is normalized into a common schema. This allows you to find, sort, compare, and analyze all events using the same event fields.

SmartConnectors are built on a connector framework, which offers advanced features such as throttling, bandwidth management, caching, state persistence, filtering, encryption, and event enrichment, to ensure reliability, completeness, and security of log collection, while also optimizing the network usage.

The granular normalization of log data allows for the deterministic correlation that detects the latest threats including Advanced Persistent Threats and prepares data to be fed into machine learning models. SmartConnector technology supports over 400 different device types, such as routers, e-mail servers, anti-virus products, firewalls, intrusion detection systems (IDS), access control servers, VPN systems, anti-DoS appliances, operating system logs, and other sources that detect and report security or audit information.

SmartConnectors leverage ArcSight's industry-standard Common Event Format (CEF) for both OpenText and certified device vendors. This partner ecosystem keeps growing not only with the number of supported devices but also with the level of native adoption of CEF from device vendors.

SmartConnector Features

Connectors both receive and retrieve information from network devices. If the device sends information, the connector becomes a receiver. But, if the device does not send information, the connector can retrieve it.

SmartConnectors are also available to forward events between ArcSight systems such as Transformation Hub and ESM, enabling the creation of multi-tier monitoring and logging architectures for large organizations and Managed Service Providers.

Connectors perform the following tasks:

- Collect all the data from a source device, which eliminates the need to return to the device during an investigation or audit.
- Parse individual events and normalize event values such as severity, priority, and time zone into a common schema (format) for use by the ESM Manager.
- Filter out data that is not needed for analysis, thus saving network bandwidth and storage space (optional).
- Filter and aggregate events to reduce the volume sent to the Manager, ArcSight Logger, or other destinations, which reduces event processing time and increases efficiency of ArcSight.
- Categorize events by using a common, human-readable format, saving time, and making it easier to use the event categories to build filters, rules, reports, and data monitors.
- Add device and event information to it to complete the message and send it to the configured destination.
- Pass processed events to the ESM Manager.

After the connectors normalize and send events to the ESM Manager, the events are stored in the centralized ESM database. ESM then filters and cross-correlates these events with rules to generate meta-events. The meta-events then are automatically sent to administrators with corresponding Knowledge Base articles that contain information supporting their enterprise's policies and procedures.

Depending on the network device, some connectors can issue commands to devices. These actions can be executed manually or through automated actions from rules and some data monitors.

Specific connector configuration guides document device-to-ESM event mapping information for individual vendor devices, as well as specific installation parameters and configuration information.

Data collection

Connectors are specifically developed to work with network and security products by using multiple techniques such as simple log forwarding and parsing, direct installation on native devices, SNMP, and syslog.

The connectors support the following data collection and event reporting formats:

- Log File Readers (including text and log file)
- Syslog
- SNMP
- Database
- XML
- Proprietary protocols, such as OPSEC

The ArcSight ESM Console, ESM Manager, and connectors communicate using HTTP over Secure Sockets Layer (SSL also referred to as HTTPS).

Different connectors are available for the following types of vendor devices:

- Network and host-based IDS and IPS
- VPN, Firewall, router, and switch devices
- Vulnerability management and reporting systems
- Access and identity management
- Operating systems, Web servers, content delivery, log consolidators, and aggregators

For more information about the types of SmartConnectors, see "Types of SmartConnectors" on page 72.

Data encryption

Connectors provide SecureData format-preserving encryption to adhere to the regulatory requirement, which mandates that data leaving the connector machine to another destination must be encrypted. This feature is supported only on Linux and Windows 64-bit platforms. For more information about the format preserving parameters for connectors, refer to the Configuration Guide for the specific connector.

You can enable data encryption either during installation or while configuring a connector. You must provide the URL of the encryption server, the identity and shared secret configured for

SecureData, and the fields to be encrypted when configuring the connector. If a proxy is enabled for the machine, you need a proxy host and port for an HTTP connection.

Important:

- If you enable encryption, you cannot change any of the encryption parameters later. To change any parameters, you must reinstall the connector.
- To enable encryption on a connector that is already installed, use the wizard to select the **Modify Connector Parameters** option.
- In deployments where multiple connectors are chained or cascaded before reaching the destination, the encryption must only be enabled at the very first connector.
- Encryption of address fields including the IP addresses and MAC addresses are not supported.
- If the input data to be encrypted is in digits, then it must be at least three characters long.
- Additional data fields cannot be selected for encryption.
- For event data transfer, although the connector and the destination can be set to FIPS-compliant mode, if encryption is enabled, the communication between the connector and the secure server is not FIPS-compliant.
- Derived event fields cannot be chosen for encryption. If any of the derived fields need encryption, include the parent field for encryption.
- For optimum performance, the number of encrypted fields must be limited to 20.

Event filtering and aggregation

Filtering

You can add filter conditions to sort the events passed to the destination according to specific criteria during SmartConnector installation and configuration. For example, you can use filters to sort out events with certain characteristics, from specific network devices, or generated by vulnerability scanners. The events that do not meet the Connector filtering criteria are not forwarded.

To remove events that are not of interest or include only events that are of interest to your organization before they are ingested, you can use Customized Events Filtering.

For more information about configuring Filtering, see Managing SmartConnector Filter Conditions.

Aggregation

The Connector can be configured to aggregate (summarize and merge) events that have the same values in a specified set of fields, either for a specified number of times or within a specified time limit.

Connector aggregation compiles events with matching values into a single event. The aggregated event contains only the values that are common to events, and the earliest start time and latest end time. This reduces the number of individual events that must be evaluated. An event that repeats every 500 ms, for example, can be represented by a single event that is generated every 10 seconds, producing a 20:1 event compression. Individual connectors can be configured to aggregate events, thus reducing event traffic to the ESM Manager and the storage requirements in the ESM database.

For example, if the connector is configured to aggregate events with a certain Source IP and Port, Destination IP and Port, and Device Action whenever the events occur 10 times in 30 seconds. If 10 events with these matching values are received by the connector within that time frame, they are grouped into a single event with an aggregated event count of 10.

If the 30-seconds time frame expires and the connector receives only two matching events, the connector creates a single aggregated event with an aggregated event count of two. If 900 matching events are generated during 30 seconds, the connector creates 90 aggregated events, each with an aggregated event count of 10.

Firewalls are a good candidate for aggregation because of the volume of events with similar data coming in from multiple devices.

Unique Generator aa ID

Globally unique event ID (GEID) is an optional feature that can be enabled by updating certain parameters. Ideally, each event passing through an ArcSight product must be assigned a GEID.

The Generator ID is a value between 1 to 16383 and is used to create GEIDs in a sequential order that can register up to one million instances per second. Previous SmartConnector versions must be upgraded so that the events are properly assigned with GEIDs. GEIDs cannot be unassigned.

If you do not specify a value for Unique Generator ID:

- The GEID generated by the connector sets **zero** as the default value.
- The connector wizard displays a message, indicating that the Unique Generator ID has not been set.

- The **agent.log** file displays a message, indicating that the Unique Generator ID has not been set.
- When you create the **silent-properties** file, the value for the **containeroptionsconfig.agent.generator.id** property will be empty.
- Events will not be processed when **Amazon S3** is configured as one of the destinations or if **Recon** is selected as the value for the **Check Event Integrity Method** parameter for any destination.

Data mapping to vendor events

Connectors collect the vendor-specific event fields logged by a network device. Before these events are forwarded to their configured destination the events are mapped to the ArcSight data fields within the connector, based on the ArcSight ESM schema.

For specific mappings between the connector data fields and supported vendor-specific event definitions, see the configuration guide, available on SmartConnectors Grand List - (A-Z), for the device-specific connector. For example: for the SmartConnector for Cisco PIX/ASA Syslog mappings, see the Configuration Guide for Cisco PIX/ ASA Syslog SmartConnector.

General mappings for ArcSight Common Event Format connectors are documented in the Implementing ArcSight Common Event Format (CEF) guide.

FIPS compliance

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 mode is supported on local, and remote SmartConnectors.

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

FIPS Suite B

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.

FIPS compliant Connectors

The following connectors are FIPS compliant:

- All syslog connectors
- All file reader connectors
- All SNMP connectors
- Most database connectors (except Oracle Audit DB and when using SQL Server drivers with encryption)
- Cisco Secure IPS SDEE connectors
- Sourcefire Defense Center eStreamer connector
- Check Point OPSEC NG connector

FIPS non-compliant SmartConnectors

The following SmartConnectors are not FIPS compliant:

- Database connectors using SQL Server drivers with encryption
- Connectors using Oracle drivers

SmartConnectors Not certified as FIPS compliant

The following connectors are not certified as FIPS compliant:

- API connectors with proprietary internal mechanisms
- Web Services and Cloud connectors

Types of SmartConnectors

Depending on your requirement, you can select any of the following SmartConnector types:

- API Connectors
- Database Connectors

- File Connectors
- FlexConnectors
- Microsoft Windows Event Log Connectors
- Model Import Connectors
- Other connectors
- Scanner Connectors
- SNMP Connectors
- Syslog Connectors

API Connectors

API connectors use a standard or proprietary API to pull events from devices. In most cases, a certificate must be imported from the device to authenticate connector access to the device. There are also several configuration steps required on the device side. For more information, refer to the respective connector configuration guides.

Database Connectors

Database connectors support event collection from databases. They use SQL queries to periodically poll for events. Connectors support major database types, including MS SQL, MS Access, MySQL, Oracle, DB2, Postgres, and Sybase.

The database user must have adequate permission to access and read the database. For Audit database connectors, such as SQL Server Audit DB and Oracle Audit DB, system administrator permission is required.

Some database connectors such as the Microsoft SQL Server Multiple Instance DB connector support multiple database events. Connectors such as the connector for McAfee Vulnerability Manager DB collect events from scanner databases.

Note: Refer to FIPS Compliance Limitation to understand the limitations for some of the database SmartConnectors.

File Connectors

File connectors are normally installed on the device machine, but when the monitored files are accessible through network shares or NFS mounts, the connectors can be installed on remote machines as well.

Types of File Connectors:

- **Real Time**

  Real Time log file connectors read normal log files in which lines are separated by a new line character or fixed length records, in which a file consists of only one line but contain multiple records of fixed length.

  These connectors can continue to follow a log file that retains its name or changes its name based on the current date and other factors. Depending on the number of files monitored, Real Time connectors can be of type that monitors a single log file or of type that monitors multiple log files.

- **Folder Follower**

  Folder follower connectors monitor files copied to a folder. There are connectors that monitor a single log file in a folder and connectors that monitor log files recursively.

  Depending on the device type, connectors support **.txt** and **.xml** file types. Most of the scanner file connectors, such as Nessus, and NeXpose are in **.xml** format.

  The type of log file connector is not usually part of the connector name unless both types of connector exist for a particular device.

  Some connectors require a trigger file to let the connector know when the file is complete and ready for processing. This file typically has the same file name with a different extension. Files are renamed by default to increments such as **.processed**, **.processed.1**, and so on.

FlexConnectors

FlexConnectors allow you to create custom connectors that can read and parse information from third-party devices and map that information to the ArcSight event schema. When creating a custom connector, you define a set of properties (a configuration file) that identify the format of the log file or other source that is imported into the ESM Manager or Logger.

The FlexConnector framework is a software development kit (SDK) that lets you create a connector tailored to the devices on your network and their specific event data. For more information about FlexConnectors and how to use them, see the FlexConnector Developer's Guide.

Microsoft Windows Event Log Connectors

Microsoft Windows Event Log Connectors connect to local or remote Windows machines inside a single domain or in multiple domains, to retrieve and process security and system events.

System administrators use Windows Event Log to troubleshoot errors. Each entry in the event log contains information related to the severity of Error, Warning, Information, and Success Audit or Failure Audit messages.

There are following types of default Windows Event Logs:

- Application log, which tracks events that occur in a registered application.
- Security log, which tracks security changes and possible breaches in security.
- System log, which tracks system events.

The following connectors are available for Microsoft Windows Event Log:

- SmartConnector for Microsoft Windows Event Log
- SmartConnector for Microsoft Windows Event Log – Native

  For more information about the Native connector, see the configuration guide for the SmartConnector for Microsoft Windows Event Log - Native.

  For mappings, see SmartConnector for Microsoft Windows Event Log Native Windows Security Event Mappings document.

These connectors provide support for partial event parsing based on the Windows event header for all System and Application events. It also provides support for a FlexConnector-like framework that lets users create and deploy their parsers to parse event description for all System and Application events.

Some individual Windows Event Log applications are supported by the connectors for Microsoft Windows Event Log, for which Windows Event Log application or system support has been developed. See the configuration guides for specific connectors for a list of application and system events supported.

Model Import Connectors

Rather than collecting and forwarding events from devices, Model Import Connectors import user data from an Identity Management system into ArcSight ESM. For more information, see the individual configuration guides for Model Import Connectors on ArcSight Enterprise Security Manager (ESM) Documentation.

Model Import Connectors extract the user identity information from the database and populate the following lists in ESM with the data:

- Identity Roles Session List
- Identity Information Session List
- Account-to-Identity Map Active List

These lists are populated dynamically, which means that, as the identity data changes in the Identity Manager, the data in the lists are updated when you refresh the session list.

Other Connectors

Connectors that Use Multiple Mechanisms

Some connectors use multiple mechanisms. For example, the connector for Oracle Audit Database monitors both the database tables and audit files.

Connectors that Use TCP in Special Formats

Examples of connectors that use TCP in special formats are :

- **IP NetFlow (NetFlow/J-Flow):** Retrieves data over TCP in a Cisco-defined binary format.
- **ArcSight Streaming Connector:** Retrieves data over TCP from Logger in an ArcSight-proprietary format.

Scanner Connectors

There are two types of scanner connectors, those whose results are retained within a file, and those retrieved from a database.

Results for XML scanner connectors are retained in a file, making them log file connectors. Other scanners deposit their scanned events in a database and are treated as database connectors, and require the installation parameters used by the database connectors.

Scan reports are converted into base events, which for ESM destinations, can be viewed on the Console. The aggregated meta events are not displayed in the Console. Meta events create assets, asset categories, open ports, and vulnerabilities on the Console.

Scanner connectors can run in either of the following modes:

- **Interactive mode**

  In the Interactive mode, a graphical user interface shows the reports or log files available for import from the configured log directory. You can select the reports to send to the connector by selecting the **Send for individual log files** check box and clicking **Send to ArcSight**.
- **Automatic mode**

  Automatic mode is used in conjunction with an automated procedure to periodically run scans. The procedure, or shell script, must execute the scanner periodically and save a

report in **.cef** format. After the scan completes and the report is saved, an empty file called **<reportname>.cef_ready** must be created, which indicates to the connector that the **.cef** report is ready for importing. The connector continues to search for **.cef_ready** files and processes the corresponding **.cef** reports. The processed reports are renamed to **<original report file>.cef_processed**.

Parameter values required for scanner installation depends on whether you are installing a file or a database connector. File connectors require the absolute path to and name of the log file is required.

SNMP Connectors

SNMP Traps contain variable bindings, each of which holds a different piece of information for the event. They are usually sent over UDP to port 162, although the port can be changed.

SNMP connectors listen on port 162 by default or any other configured port and process the received traps. They can receive multiple trap types from the device but process traps only from one device with a unique Enterprise object identifier (OID).

SNMP is based on UDP, so there is a minor possibility of events being lost over the network.

Although there are several SNMP connectors for individual connectors, most SNMP support is provided by the SmartConnector for SNMP Unified. Parsers use the knowledge of the Management Information Base (MIB) to map the event fields, but, unlike some other SNMP-based applications, the connector itself does not require the MIB to be loaded.

Syslog Connectors

Syslog messages are free-form log messages prefixed with a Syslog header consisting of a numerical code (facility + severity), timestamp, and host name. Unlike file connectors, a Syslog connector can receive and process events from multiple devices. There is a unique regular expression that identifies the device.

TCP is a supported protocol for Syslog connectors. If UDP is used, there might be a possibility of missing Syslog messages over the network.

Depending on the mechanism with which the device logs are made available to the smartconnector, select the type of smartconnector to install:

- **Syslog Deamon SmartConnector** or **Syslog NG Deamon SmartConnector** - If the device writes logs to a port.
- **Syslog File SmartConnector** - If the device writes the log to a pipe or if the device writes log to a file.

SmartConnector Types	Available Parsers
• **Syslog Deamon:** The Syslog Daemon SmartConnector is a syslogd-compatible daemon designed to work in operating systems that have no syslog daemon in their default configuration, such as Microsoft Windows. They listen for Syslog messages on a configurable port, using port 514 by default. The default protocol is UDP, but other protocols such as Raw TCP are also supported. It is the only Syslog option supported for Windows platforms. • **Syslog File:** Supports the following types of logs: ◦ **Logs written to Pipe** When there is an existing syslog daemon syslogd is configured to write to a named pipe, and the SmartConnector reads from it to receive events. They require syslog configuration to send messages with a certain Syslog facility and severity. It is especially useful when storage is a factor. The Solaris platform tends to under-perform when using Syslog Pipe connectors. The operating system requires	AirMagnet Enterprise Syslog

SmartConnector Types	Available Parsers
	Apache HTTP Server Syslog

SmartConnector Types	Available Parsers
	Arbor Networks Peakflow Syslog

SmartConnector Types	Available Parsers
	ArcSight Common Event Format Syslog

SmartConnector Types	Available Parsers
	Barracuda Email Security Gateway Syslog

SmartConnector Types	Available Parsers
	Barracuda Firewall NG F-Series Syslog

SmartConnector Types	Available Parsers
	Barracuda Web Appliance Firewall Syslog

SmartConnector Types	Available Parsers
	Blue Coat Proxy SG Syslog

SmartConnector Types	Available Parsers
	BroadWeb NetKeeper Syslog

SmartConnector Types	Available Parsers
	Brocade BigIron Syslog

SmartConnector Types	Available Parsers
	Check Point Syslog

SmartConnector Types	Available Parsers
	Cisco ASA Syslog

SmartConnector Types	Available Parsers
	Cisco Catalyst OS Syslog

SmartConnector Types	Available Parsers
	Cisco IOS Syslog

SmartConnector Types	Available Parsers
	Cisco IronPort Email Security Appliance Syslog

SmartConnector Types	Available Parsers
	Cisco IronPort Web Security Appliance Syslog

SmartConnector Types	Available Parsers
	Cisco ISE Syslog

SmartConnector Types	Available Parsers
	Cisco Meraki Syslog

SmartConnector Types	Available Parsers
	Cisco Mobility Services Engine Syslog

SmartConnector Types	Available Parsers
	Cisco NX-OS Syslog

SmartConnector Types	Available Parsers
	Cisco Secure ACS Syslog

SmartConnector Types	Available Parsers
	Cisco Wireless LAN Controller Syslog

SmartConnector Types	Available Parsers
	Citrix NetScaler Syslog

SmartConnector Types	Available Parsers
	Dell SonicWALL Firewall Syslog

SmartConnector Types	Available Parsers
	F5 BIG-IP Syslog

SmartConnector Types	Available Parsers
	Fortinet Fortigate Syslog

SmartConnector Types	Available Parsers
	HoneyD Syslog

SmartConnector Types	Available Parsers
	HPE Aruba Mobility Controller Syslog

SmartConnector Types	Available Parsers
	HPE c7000 Virtual Connect Module Syslog
	HPE H3C Syslog
	HPE Integrated Lights-Out Syslog
	HP Printers Syslog
	HPE ProCurve Syslog
	HPE-UX Syslog
	IBM AIX Audit Syslog
	IBM Security Access Manager Syslog
	Infoblox NIOS Syslog
	Ingrian DataSecure Syslog
	Intersect Alliance SNARE Syslog
	ISC Bind Syslog
	ISC DHCP Syslog
	Juniper Firewall ScreenOS Syslog
	Juniper IDP Series Syslog
	Juniper JUNOS Syslog
	Juniper Network and Security Management Syslog
	Linux Audit Syslog
	McAfee Email Gateway Syslog
	McAfee Firewall Enterprise Syslog
	McAfee Network Security Manager Syslog
	McAfee Web Gateway Syslog
	Microsoft IIS Syslog
	NetApp Filer Syslog
	Netscout Arbor Security Syslog
	NitroSecurity Syslog
	Nortel Contivity Switch (VPN) Syslog
	Oracle Audit Syslog
	Oracle Solaris Basic Security Module Syslog
	Proofpoint Enterprise Protect and Enterprise Privacy Syslog
	Pulse Secure Pulse Connect Secure Syslog

SmartConnector Types	Available Parsers
	Radware DefensePro Syslog
	Sabernet NT Syslog
	Sendmail Syslog
	Snort Syslog
	Symantec Endpoint Protection Syslog
	Symantec Messaging Gateway Syslog
	TippingPoint SMS Syslog
	TippingPoint SMS Syslog Extended
	Top Layer Attack Mitigator Syslog
	Type80 SMA_RT Syslog
	UNIX OS Syslog
	VarySys PacketAlarm IPS Syslog
	VMware ESXi Server Syslog
	Vormetric CoreGuard Syslog

Other Syslog connectors are:

Raw Syslog: They are always used with the Raw Syslog destination. Raw Syslog connectors generally do not parse events. But, they take the Syslog string and copy it in the rawEvent field as-is . The Raw Syslog destination type takes the **rawEvent** field and sends it as-is by using UDP, Raw TCP, or TLS protocol, that is selected. The event flow is streamlined to eliminate components that do not add value. For example, with the Raw Syslog transport, the category fields in the event are ignored, so the categorization components are skipped. If you are transporting data to ArcSight Logger, you can use specific configuration parameters to provide minimal normalization of the Syslog data (for source and timestamp).

ArcSight CEF CISCO FireSight Syslog: Retrieves events and payload information from FireSIGHT DB by using the event ID and Sensor Name as input.

ArcSight CEF Encrypted Syslog UDP: Allows connector-to-connector communication through an encrypted channel by decrypting events previously encrypted through the CEF Encrypted Syslog (UDP) destination. The CEF connector lets ESM connect to aggregate, filter, correlate, and analyze events from applications and devices that deliver their logs in the CEF standard, by using the Syslog transport protocol.

UNIX supports all types of Syslog connectors. If a syslog process is already running, you can end the process or run the connector on a different port. The connector for UNIX OS Syslog provides the base parser for all Syslog sub-connectors.

For Syslog connector deployment information, see the connector Configuration Guide for UNIX OS Syslog. For device-specific configuration information and field mappings, see the connector configuration guide for the specific device. Each Syslog sub-connector has its own configuration guide.

Types of destinations

You can configure a connector to send events to one or more destinations. A destination is a Manager or device that can receive events from a connector. In addition to the selections configured during connector configuration, events can also be sent to additional or failover destinations.

Depending on your requirement, you can select any of the following destinations:

ArcSight Manager (encrypted)

If SmartConnectors are configured to use ArcSight Manager as a destination, they send events to the ESM Manger.

When connectors send events to ESM Manager, it stores the events in a relational database, processes them using its correlation engine, and makes them visible to the Console or Web interfaces. This is the commonly destination used.

For more information about the parameters to be selected during installation, see ArcSight Manager Parameters.

ArcSight Logger SmartMessage (encrypted)

Logger is a log management solution that is optimized for extremely high event throughput. Logger logs or stores time-stamped text messages, called events, at high sustained input rates. Events consist of a receipt time, a source (host name or IP address), and an un-parsed message portion. Logger compresses raw data, but also can retrieve it in an unmodified form for forensics-quality litigation reporting. Unlike ESM, Logger does not normalize events.

CIf SmartConnectors are configured to use ArcSight Logger SmartMessage as a destination, they send CEF events to Logger using an encrypted, optionally compressed channel called SmartMessage. Logger also can receive CEF syslog events from connectors.

To subscribe event data from a specific SmartConnector, do the following:

- Configure all the SmartConnectors to publish events to the same topic. Configure the Logger's Transformation Hub receiver to subscribe to this event topic. For more information, see the Administrator's Guide for Transformation Hub.
- Configure each SmartConnectors to publish events to different topics and then configure the Transformation Hub receiver on the Logger to subscribe to multiple event topics.

For more information about the parameters to be selected during installation, see ArcSight Logger SmartMessage Parameters.

You can also configure the SmartMessage transport to be persistent to achieve higher throughput for Logger destinations. For more information, see Configuring Persistent SmartMessage Transport.

ArcSight Logger SmartMessage Pool (encrypted)

You can specify a pool of logger devices as a single destination while the events are distributed among the loggers in the pool. Each batch of events processed by the connector is sent to the next logger in the pool in a round-robin fashion. You can either add the pool members one by one or use a CSV file that contains the predefined information for logger secure pool. You can also export and save the data entered in the panel into a CSV file.

For more information about the parameters to be selected during installation, see ArcSight Logger SmartMessage Pool Parameters.

Related Topics:

- Configuring Persistent SmartMessage Transport
- ArcSight Logger SmartMessage Pool (encrypted) Destination Parameters

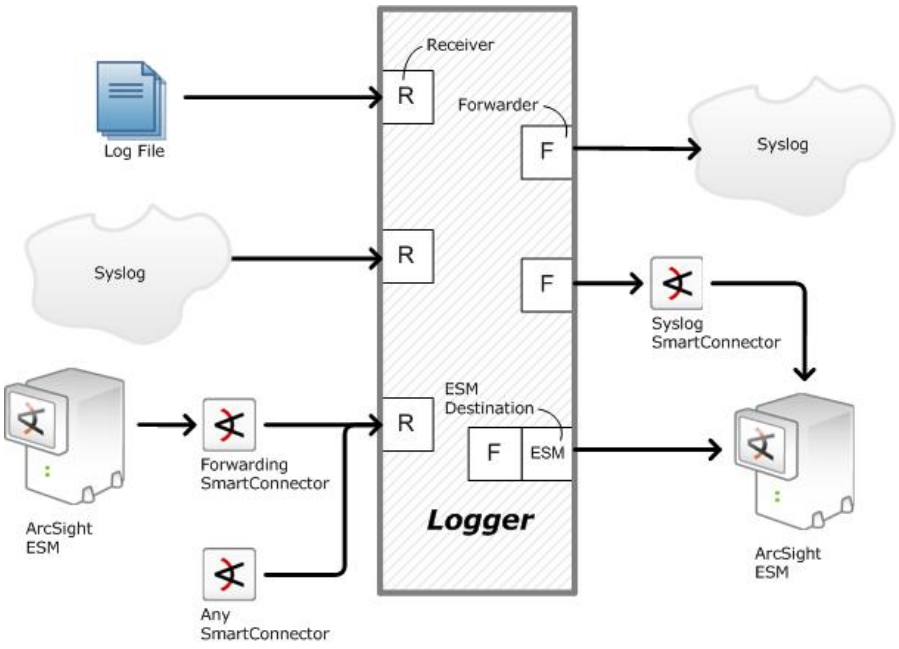
Sending Events from Logger to a Manager

Logger's most basic function is to store a large volume of security events. Logger can send a subset of these events to a Manager. It sends syslog or ArcSight Common Event Format (CEF) events directly to ESM through a built-in Connector called an ESM Destination. An ESM Destination appears as a Connector on a Console. For more information about ESM Destinations, see the *ArcSight Logger Administrator's Guide*.

SmartMessage is ArcSight technology used by Logger to provide a secure channel between Connectors and Logger. SmartMessage provides an end-to-end encrypted secure channel. At one end is a Connector, receiving events from the devices it supports; on the other end is SmartMessage Receiver on Logger.

Note: Use Syslog connector to forward events from Logger to ESM. If a different method such as Netcat is used, the events are forwarded to Logger, but not to ESM.

Logger Receivers (R) and Forwarders (F)



Note: The SmartMessage secure channel uses HTTPS (secure sockets layer protocol) to send encrypted events to Logger. This is similar to, but different from, the encrypted binary protocol used between Connectors and the ESM Manager.

Use port 443 (rather than ArcSight traditional port 8443) because the secure channel uses HTTPS.

Sending Events to Both Logger and a Manager

1. Set up the SmartMessage Receiver on Logger (see the configuration guide for the connector being installed).
2. Install the connector component (see the Connector Configuration Guide for your device).
3. Register the connector with an active ESM Manager and test that the connector is up and running.
4. Using the `$ARCSIGHT_HOME\current\bin\runagentsetup` script, restart the connector configuration program.
5. Select **Add, modify, or remove destinations** and click **Next**.
6. Select **Add destination** and click **Next**.
7. Select **ArcSight Logger SmartMessage (encrypted)** and click **Next**.
8. Enter the destination parameters and click **Next**:

Parameter	Description
Host Name/IP	The destination host name or IP address.
Port	The destination port 443 for Logger Appliance or 9000 for Software Logger.
Receiver Name	The destination receiver name. This setting should match the Receiver name you created in step 1 so that Logger can listen to events from this Connector.
Compression Mode	Select the option to enable or disable data compression. Default is **Disabled**.

9. If you have not already imported the certificate, the Logger certificate message is displayed to import the certificate to connector.
10. Select the **Import the certificate to connector from destination** option and click **Next**.
11. Navigate through the subsequent windows until you receive a message that confirms the configuration was successful. Click **Exit**, then click **Next** to exit the wizard.
12. Restart the connector for changes to take effect.

Sending Events to Logger

1. Set up the SmartMessage Receiver on Logger (see the *ArcSight Logger Administrator's Guide* for detailed instructions).
2. Install the connector component (see the Connector Configuration Guide for your device).
3. Using the `$ARCSIGHT_HOME\current\bin\runagentsetup` script, restart the connector configuration program.
4. Navigate through the windows, select **ArcSight Logger SmartMessage (encrypted)**, and then click **Next**.
5. Enter the destination parameter details and click **Next**.

Parameter	Description
Host Name/IP	The destination host name or IP address.
Port	The destination port 443 for Logger Appliance or 9000 for Software Logger.

Parameter	Description
Receiver Name	The destination receiver name. This setting should match the Receiver name you created in step 1 so that Logger can listen to events from this Connector.
Compression Mode	Select the option to enable or disable data compression. Default is **Disabled**.
CEF Version	Select any of the following options from the drop-down menu: • **0.1** - The Device Address, Source Address, Destination Address, and Agent Address fields will always be IPv4 or will be omitted. If there are any IPv6 addresses, they will be in the **Device Custom IPv6 Address** fields. The **Bytes In** and **Bytes Out** fields are limited to the size of an integer (up to 2^31-1). **Note**: Select **0.1** if you are not sure the chosen destination can handle CEF 1.0, which supports both IPv4 and IPv6modes. • **1.0** - Any of the address fields can be either IPv4 or IPv6 and the **Bytes In** and **Bytes Out** fields can be long values (up to 2^63-1).

6. If you have not already imported the certificate, the Logger certificate message is displayed to import the certificate to connector.
7. Select the **Import the certificate to connector from destination** option and click **Next**.
8. Navigate through the subsequent windows until you receive a message that confirms the configuration was successful. Click **Exit** and click **Next** to exit the wizard.
9. Restart the connector for changes to take effect.

Forwarding Events from ESM to Logger

The ArcSight Forwarding Connector can read events from an ESM Manager and forward them to Logger using ArcSight's Common Event Format (CEF).

Note: The Forwarding Connector is a separate installable file, named similarly to this: `ArcSight-6.x.x.<build>.x-SuperConnector-<platform>.exe`.

Use Forwarding Connector build 4810 or later for compatibility with Logger 1.5 or later.

1. Install the connector component (see the Connector Configuration Guide for your device).
2. Using the `$ARCSIGHT_HOME\current\bin\runagentsetup` script, restart the connector configuration program.
3. Navigate through the windows, select **ArcSight Logger SmartMessage (encrypted)**, and then click **Next**.
4. Enter the destination parameter details and click **Next**.

Parameter	Description
Host Name/IP	The destination host name or IP address.
Port	The destination port 443 for Logger Appliance or 9000 for Software Logger.
Receiver Name	The destination receiver name. This setting should match the Receiver name you created in step 1 so that Logger can listen to events from this Connector.
Compression Mode	Select the option to enable or disable data compression. Default is **Disabled**.

5. If you have not already imported the certificate, the Logger certificate message is displayed to import the certificate to connector.
6. Select the **Import the certificate to connector from destination** option and click **Next**.
7. Navigate through the subsequent windows until you receive a message that confirms the configuration was successful. Click **Exit** and click **Next** to exit the wizard.
8. Restart the connector for changes to take effect.

To configure the Forwarding Connector to send CEF output to Logger and send events to another Manager at the same time, see Sending Events to Both Logger and ESM.

ArcSight SaaS

If **ArcSight SaaS** is configured as a destination, all security events are sent in **Avro** format to Amazon MSK that is managed by ArcSight's SaaS offering.

For more information about the destination parameters to be selected during installation, see ArcSight SaaS.

It is mandatory that Admin must always get the registration URL before starting the connector install.

The registration URL for the **ArcSight SaaS** destination can be used only once. You can neither add failover destination for the **ArcSight SaaS** destination, nor modify the destination parameters.

When the access is revoked, events are no longer sent to Amazon MSK. A message indicating the same will be displayed in the logs. If you need to send events, then you must re-register the **ArcSight SaaS** destination with a new registration URL. For more information, see "Re-registering a destination" on page 168.

Note: If you re-register the **ArcSight SaaS** destination, all cached events in the connector will be lost. For more information, see "Events are not sent from SmartConnector to ArcSight SaaS" on page 228.

Transformation Hub

If SmartConnectors are configured to use Transformation Hub as a destination, they send events to Transformation Hub's Kafka cluster, from where the events are further distributed to real-time analysis and data warehousing systems.

The Transformation Hub destination is used to send events to a Transformation Hub cluster in in Avro, binary, or CEF format, which can then further distribute events to real-time analysis and data warehousing systems. Any application that supports retrieving data from Transformation Hub can receive these events (for example, ESM, ArcSight Investigate, Hadoop and Logger).

The SmartConnector Acknowledgments ("acks") ensure that Transformation Hub received the event before the SmartConnector removes it from its local queue. Acknowledgments do not indicate that consumers, such as Logger, have received the event data, only that Transformation Hub itself has. You can disable acknowledgments, enable to receive acknowledgment only from the primary replica, or enable every replica to acknowledge the event.

Supported SmartConnector versions encode their own IP address as meta-data in the Kafka message for consumers that require that information such as Logger Device Groups.

For instructions about setting up FIPS with Transformation Hub and SmartConnectors, see Configuring Security Mode for Smart Connectors with Transformation Hub Destinations.

For the content format Avro:

The SmartConnector uses Avro schema to emit the Avro output. Avro schema resides in the Schema Registry of Transformation Hub. The SmartConnector makes an HTTPS call to Transformation Hub to get and save the schema in its `user/agent` folder. The SmartConnector captures the Schema Registry details during the installation and fetches schema during its every restart.

Ensure that you use the compatible version of SmartConnector with Transformation Hub in order to emit Avro output as follows:

SmartConnector Version	Default Avro Schema Version	Transformation Hub Version
8.4	1.2.0	3.6 and 3.6.1

8.3	1.2.0	3.6
8.2	1.1.1	3.5

Note: You must install or upgrade Transformation Hub before upgrading SmartConnector.

To use a SmartConnector with the non-compatible version of Transformation Hub, perform the following steps after installing the SmartConnector:

1. Open the `$ARCSIGHT_HOME/current/user/agent/agent.properties` file.
2. Modify the **schema.registry.schema.version** parameter value to the required schema version. The currently supported versions are: 1.1.1 and 1.2.0

   For example: For SmartConnector 8.4 to work with Transformation Hub 3.5, set the property value to 1.1.1 as follows: **schema.registry.schema.version=1.1.1**
3. Restart the SmartConnector.

For the Content Types CEF 0.1 and CEF 1.0:

The key is sent on events with the Connectors IP address and a flag. The flag format is a single byte value. For ESM, the key is the agent ID.

The key format is: one byte flags + (4 or 16 bytes) IP (v 4 or v 6) address. Based on the value of the IP version bit, 4 or 16 additional bytes should be examined. This is used in case the key is made longer in a non-breaking fashion in the future.

Bit position	Meaning
0	IP version: 0 = IPv4 1 = IPv6
1	Key version: **Must be 0**. If there are future versions of key that are not backward compatible with this definition, it changes to 1.
2-7	Key version: **Must be 0**. Reserved for future.

For CEF 0.1 and 1.0, the events are delivered to Transformation Hub in their own messages, which are distributed to the partitions of the topic defined in Transformation Hub in a round-robin manner. For ESM, the events are sent in batches in a binary format. TLS encryption is supported, as is client certificate authentication.

When TLS is enabled by setting the **Kafka Broker on SSL/TLS** parameter to **true** during destination configuration, a Java KeyStore-format (.jks) file containing the certificates of the Transformation Hub's Kafka cluster, or a certificate that has signed them, will be required. The

location of this Trust Store file will be required during destination configuration. See Kafka documentation at https://kafka.apache.org/documentation.html#security_ssl for instructions.

Also, when client certificate authentication is enabled by setting the **Use SSL/TLS Client Authentication** parameter to **true**, a .jks file containing the private key and certificate to use must be provided. The Transformation Hub cluster must have the certificate (or a certificate that has signed it) in its trust store. The location of the keystore file and authentication information is to be provided in the **SSL/TLS Keystore File Path**, **SSL/TLS Keystore Password**, and **SSL/TLS Key Password** parameters. The key and keystore passwords are created when you set up Transformation Hub.

For more information about the parameters to be selected during installation, see Transformation Hub Parameters.

Amazon MSK

If **Amazon MSK** is configured as a destination, connectors will ingest events into the Amazon MSK server. The connector generates **Avro** output by using static Avro schema which is bundled with the Connector package. For more information about schema, refer to Avro Documentation.

For more information about the destination parameters to be selected during installation, see Amazon MSK.

Amazon S3

If SmartConnectors are configured to use **Amazon S3** (Amazon Simple Storage Service) as a destination, they send security events in the Avro format to Amazon S3. The Connector generates Avro output by using static Avro schema which is bundled with the Connector package. The Avro output is generated in the snappy compressed format. The TLSv1.2 protocol is used to secure file upload to S3 bucket. For more information, refer to Avro Documentation.

The **Amazon S3** destination is also supported for all the cloud-native Connectors, such as AWS Security Hub, AWS CloudWatch, and Azure Event Hub.

For more information about the parameters to be selected during installation, see Amazon S3 Parameters.

Microsoft Azure Event Hub

If SmartConnectors are configured to use **Microsoft Azure Event Hub** as a destination, they send events in Common Event Format (CEF) through a Kafka broker to Microsoft Azure Event Hub.

Note: Event Hub must enable a Kafka endpoint.

Azure Event Hub requires SSL or TLS for communication purposes and uses Shared Access Signatures (SAS) for authentication. In the same way, this requirement must be met for a Kafka endpoint within Event Hubs. To be compatible with Kafka, Event Hub uses SASL PLAIN for authentication and SASL SSL for transport security.

For more information about the parameters to be selected during installation, see Microsoft Azure Event Hub Parameters

CEF File

The Common Event Format (CEF) is an open log management standard that improves the interoperability of security-related information from different security and network devices and applications. The CEF destination allows you to capture security events in a CEF file rather than forwarding them to a Manager.

For more information about the parameters to be selected during installation, see CEF File Parameters.

CEF Syslog

If SmartConnectors are configured to use **CEF Syslog** as a destination, they send events in CEF (converted to bytes using the UTF-8 character encoding), by using UDP, TCP, or TLS protocol.

The TCP and UDP protocols can be used to send events to Logger, where data is received using a TCP or UDP Receiver. One receiver can receive events from more than one connector. The protocols can also be used to send events to a Syslog Daemon connector or non-ArcSight syslog receivers.

The TLS protocol sends events through a secure channel (an option that does not apply to Logger), and allows for one-way or two-way authentication. This data can be received by any application that supports TLS syslog reception, which includes ArcSight's Syslog NG Daemon connector.

For more details about the Syslog NG Connector, see the SmartConnector for Syslog NG Daemon.

For more information about the parameters to be selected during installation, see CEF Syslog Parameters.

CEF Encrypted Syslog (UDP)

If SmartConnectors are configured to use **CEF Encrypted (UDP)** as a destination, they send events in Common Event Format (CEF) using the UDP protocol, providing symmetric-key encryption. This option allows for a “Shared Secret” key that requires configuration to encrypt the data. This data can be decrypted on the receiver side by the CEF Encrypted Syslog (UDP) connector.

To decrypt the data on the receiving side, ensure that you have installed and configured the ArcSight CEF Encrypted Syslog (UDP) connector.

For more information about installing the connector and decrypting the data, see the SmartConnector for ArcSight CEF Encrypted Syslog (UDP) documentation.

For more information about the parameters to be selected during installation, see CEF Encrypted Syslog (UDP)

CSV File

Use this destination to capture events that a connector sends to ESM Manager into a CSV file. Typical ArcSight configurations do not require the use of external files to communicate events to the ESM Manager.

Event data is written to a file in Excel-compatible comma-separated values (CSV) format, with comments prefixed by ‘#.’ A connector can be configured to preface the data with a comment line that describes the fields found on a subsequent line.

Event data is written to files in the specified folder and can be configured to rotate periodically.

Following are the contents of an example event file:

#event.eventName,event.attackerAddress,event.targetAddress

"Port scan detected","1.1.1.1","2.2.2.2"

"Worm ""Code red"" detected","1.1.1.1","2.2.2.2"

"SQL Slammer detected","1.1.1.1","2.2.2.2"

"Email virus detected","1.1.1.1","2.2.2.2"

Rotating Event Data

Events are appended to the current file until the rotation time interval expires, at which time a new current file is created and the previous current file is renamed. One hour is a typical

rotation time interval.

Event files are named using the time stamp of their creation, and all files, except for the current file, have the text '.done.csv' appended. For example, a typical CSV file set configured to rotate every hour might consist of files named as follows:

```
2007-01-28-10-55-33.csv
```

```
2007-01-28-09-55-33.csv.done
```

```
2007-01-28-08-55-33.csv.done
```

Using the properties file, the configuration of your CSV Connector can be customized to filter and aggregate events as desired.

A Connector can also be configured to send events to a CSV file and an ESM Manager at the same time.

For more information about the parameters to be selected during installation, see CSV File Parameters

Raw Syslog

Although normalized data is faster and easier to parse and access, many IT professionals prefer having the raw data available for review, forensics, and litigation. This destination sends raw syslog events through the UDP, TCP, or TLS protocol.

It works in conjunction with the Raw Syslog connector, which captures raw, unparsed security events for further processing. If you are transporting data to ArcSight Logger, you can use specific configuration parameters to provide minimal normalization of the syslog data (for source and timestamp). For more information, see the *SmartConnector for Raw Syslog Daemon Configuration* Guide.

Note: Connections to Qualys Cloud Platform require TLS 1.1 or higher.

For more information about the parameters to be selected during installation, see Raw Syslog Parameters.

Overview of SmartConnector installation

You can install connectors on the ESM Manager machine, the machine hosting ArcSight Management Center, a host machine, or a device. Based on their configuration, connectors also can receive events over the network using SNMP, HTTP, Syslog, proprietary protocols such as OPSEC, or direct database connections to the device's repository such as ODBC or proprietary database connections.

ArcSight components install consistently across UNIX, Windows, and Macintosh platforms. You can deploy connectors based on the requirements of your network security enterprise.

The deployment scenarios discussed in the following sections are only examples of how you might introduce ESM into your enterprise. ESM is not limited to just these scenarios and deployments.

Deployment scenarios

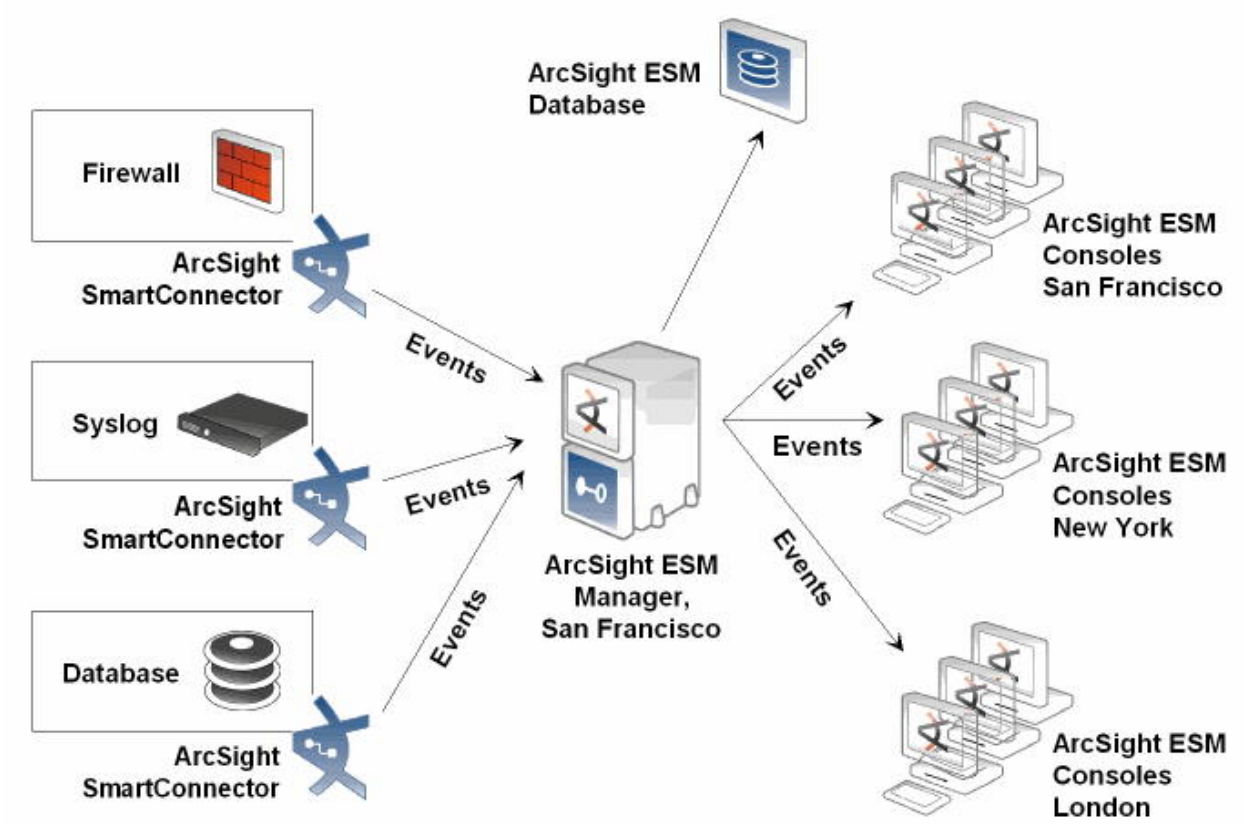
The best deployment scenario for your system depends upon the connector type, your network architecture, and your operating system.

- Scenarios for syslog deployment are documented in the Connector for UNIX OS Syslog Configuration Guide.
- Scenarios for deploying Windows Event Log connectors are documented in the follwoing configuration guides:
  - SmartConnector for Microsoft Windows Event Log Native
  - SmartConnector for Windows Event Log.

This section has the following scenarios:

Scenario 1: Connectors reside on three different devices

In this scenario, there are three connectors residing on three different devices: a firewall, an IPS, and a UNIX operating system. These connectors receive information from the devices or their logs and send captured events to the Manager based on the connector configuration.

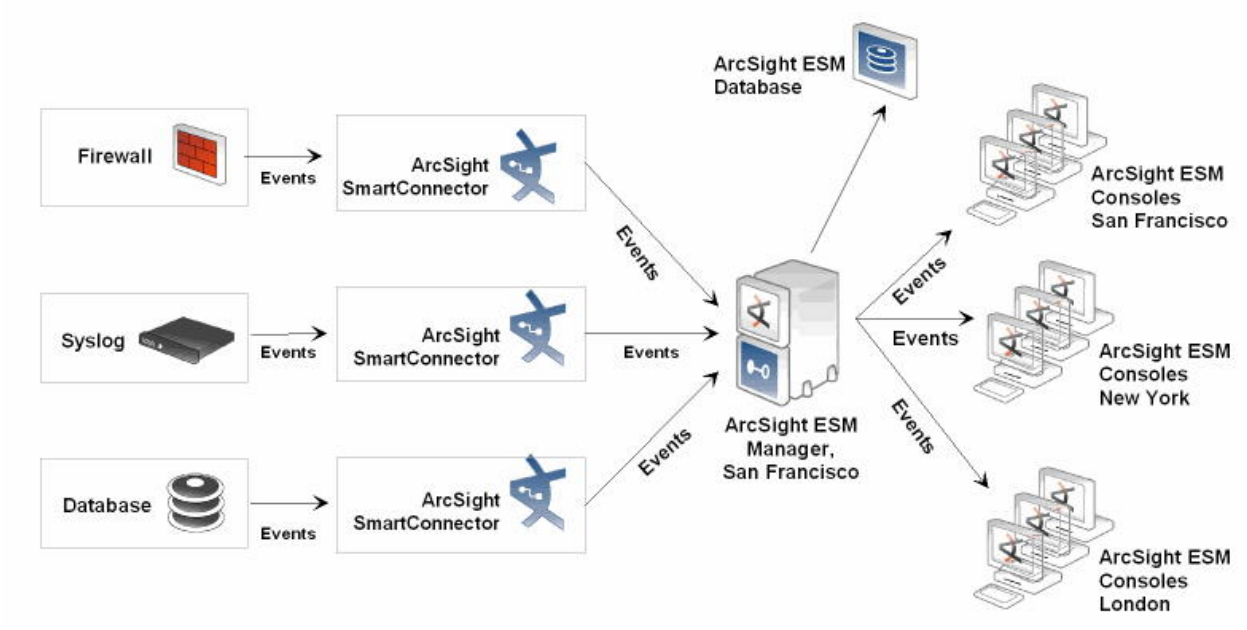


After the events are received by the Manager, it cross-correlates the events using rules, and sends meta-events to the database and to any Consoles that access the database.

The ESM Manager also can perform preset actions. Events and meta-events within the database can be played using the Replay channel to investigate, analyze, or create a report about event history.

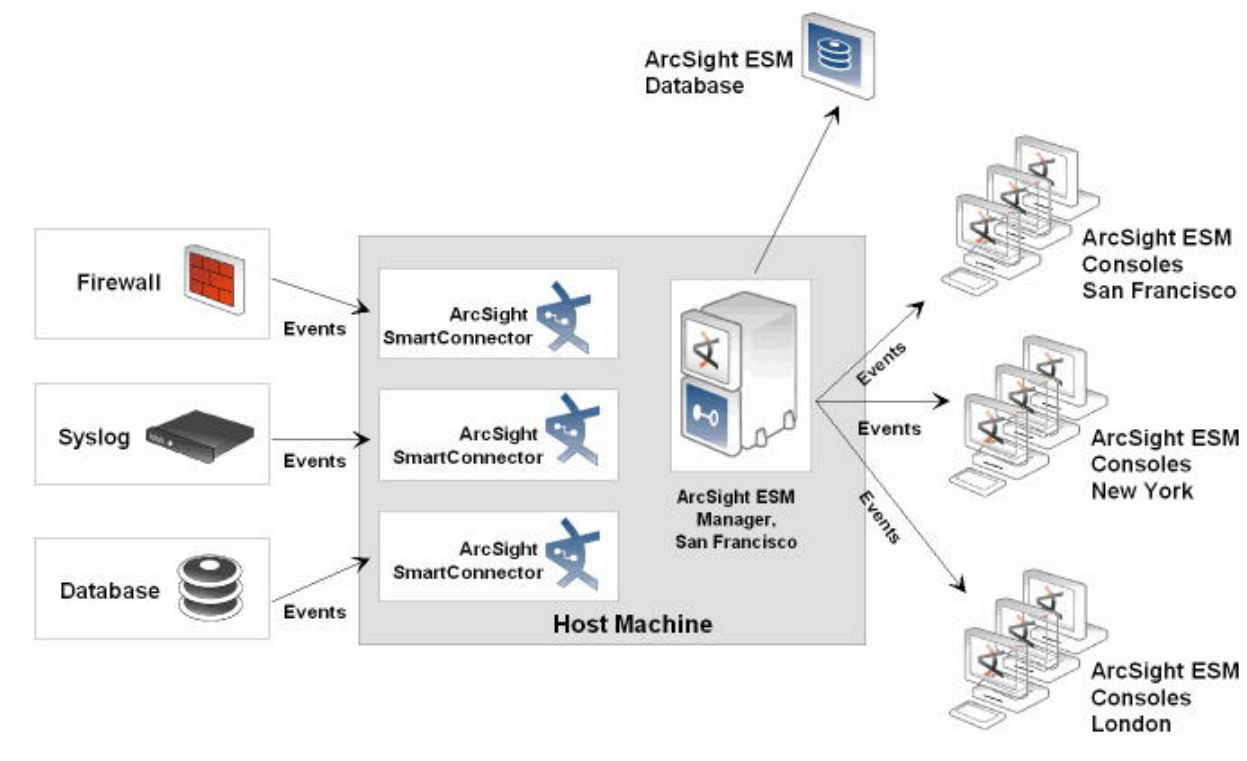
Scenario 2: Connectors reside on a host machine

In this scenario, the three connectors reside on a host machine rather than in the device itself. The connectors receive information from the devices and send captured events to the Manager based on the connector configuration. ArcSight ESM Manager and database function similar to Scenario1.



Scenario 3: Connectors reside on ESM Manager

In this scenario, the connectors reside on the ESM Manager itself, not on a host machine, but still retrieve events from devices in the network. The processing performed by the ArcSight connector, ESM Manager, and Consoles is identical to the other scenarios.



Scenario 4: Connectors are configured to send events to Logger

In this scenario, any of the previous scenarios are implemented, and the connectors are configured to send events to Logger. Events can be forwarded from Logger to ESM.

Identifying ArcMC deployment scenario

ArcSight Management Center can be deployed wherever Connectors are needed.

You can choose any one of the following deployment scenarios:

ArcSightLogger

Logger receives events from device and sends to Connectors, but lacks the depth of Connector management found in ESM.

- A Logger-only deployment benefits from ArcSight Management Center in many ways, and provides most, but not all, ESM's management function (for example, it does not contain the filter designer). ArcSight Management Center also offers features that ESM does not, such as bulk operations (enabling control of many Connectors at one time).
- ArcSight Management Center also can configure Connectors with failover destinations, providing central failover control when redundant Loggers are deployed. All or some Connectors can be configured to send events to a second Logger or to an event file in the case of communication failure with the primary destination.

  For more information about Logger, see ArcSight Logger SmartMessage Pool (encrypted) .

ArcSight ESM

Deploying ArcSight Management Center in an ESM environment centralizes connector upgrade, log management, and other configuration issues.

ESM and Logger

Management Center centralizes control when events are sent to ESM and Logger simultaneously. In one scenario, all events are sent to Logger while only high-value events are sent to ESM (for further analysis, for example). In another scenario, all events are sent to both, but Logger implements a longer retention policy.

Although each connector has specific destination parameters, Management Center allows "bulk" management of connectors, eliminating the need to manually access each remote connector host to add or change destinations.

For more information about Management Center, see the *ArcSight Management Center Administrator's Guide*.

Planning to install and deploy

This section describes the installation and deployment options, considerations, and caveats that you need to know for a successful deployment.

Installation checklist

	Task	See
❑	1. Learn about the product features, latest updates, known issues, and workarounds.	ArcSight SmartConnector Release Notes
❑	2. Deployment Scenarios	Deployment Scenario
❑	3. Make sure that the ArcSight products with which the connectors will communicate have already been installed correctly (such as ArcSight ESM or ArcSight Logger).	ArcSight Documentation
❑	4. Review and decide on the destination that you want to select	SmartConnector Destinations

Reviewing the considerations and best practices

Considerations	Best Practices
Installation Directory	• On Windows, do not install in a directory with an open or close parenthesis () character in the name. • Whether a host is dedicated to the ArcSight Database, Manager, Console, or other component, ESM software is installed under a single root directory on each host (DBMS and other third-party software is not necessarily installed under this directory.) • This is consistent with connector configuration guide information, and underscores the fact that connectors are not installed on the same machine as the remaining ESM components. Rather, they are typically installed on the same machine as the device whose activity will be monitored. • The path to this root directory is referred to as $ARCSIGHT_ HOME. In connector documentation, the 'current' directory is specified rather than presumed to be part of the $ARCSIGHT_HOME location, and the path separator is a backslash (\) (for example, $ARCSIGHT_HOME\current). • The directory structure below $ARCSIGHT_HOME is standardized across components and platforms. ArcSight software is generally available in the $ARCSIGHT_HOME\current\bin directory. Properties files, which control the ArcSight configuration, are found in $ARCSIGHT_HOME\config and log files are written to $ARCSIGHT_HOME\logs.
Installation considerations	• When installing the 32-bit SmartConnector executable on 64-bit machines,both 32-bit and 64-bit versions of glibc, libXext, libXrender, and libXtst must be installed. • If you are using RHEL 6.x or later, ensure that you have the following libraries or packages installed before installing a connector: ◦ X libraries • glibc • libXext • libXrender • libXtst • unzip ◦ fontconfig \ dejavu-sans-fonts
Naming convention	• Use a standard naming convention to specify directory locations, file names, and menu option names for the connectors you install. • If you install multiple connectors on a particular machine, install each connector in a separate directory.
User credentials	Before installing the SmartConnector, ensure that you have: • Local access to the machine where the SmartConnector is to be installed • Administrator passwords

Considerations	Best Practices
SmartConnector 64-Bit Support	The 64-bit installation executable contains a subset of available SmartConnector. See the 64-bit SmartConnector installer for your platform from the list of available connectors.
User Privileges when Installing on Unix	SmartConnectors can be run as a *non-root* user, such as *arcsight*. A SmartConnector that listens on port less than 1024 needs a *root* privilege to listen to a restricted port. For example, a Syslog Daemon connector needs a *root* privilege to bind to a restricted port such as port 514. For more information, see User Privileges when Installing on Unix.
FIPS Support	Understanding FIPS
ArcSight Management Center	If you decide to use ArcSight Management Center to manage SmartConnectors, see Identifying ArcMC Deployment Scenario.
Cloud environment	All SmartConnector remote connections depend on the Operating System random number pool (entropy pool) to generate private keys for secure communication. For a cloud environment, you might need to increase the entropy pool beyond the lower limit of 3290 to ensure uninterrupted communication. For more information see, "SmartConnector Remote Connections Failing Due to Low Entropy" on page 235.

User privileges when installing (UNIX only)

SmartConnectors can be run as a *non-root* user, such as *arcsight*. A SmartConnector that listens on port less than 1024 needs a *root* privilege to listen to a restricted port. For example, a syslog daemon connector needs a *root* privilege to bind to a restricted port such as port 514.

The following sections describe the recommended options for two concepts:

- Connectors that require to be configured to listen to low numbered ports
- Connectors that are run as a service

This section has the following information:

When running as a service

Option 1: Install as *arcsight* user, run as *arcsight* user

This is the recommended option. The following instructions refer to *arcsight* user as a generic name for any user with *non-root* privileges:

Note: When you install a connector as the *arcsight* user, the ArcSight connector files will be owned by **arcsight** user.

After installing ArcSight connector, run the connector setup wizard as *arcsight* user.

- If a Syslog Daemon connector is selected, then the configured port number must be 1024 or greater. If the configured port number is less than 1024, then see "User privileges when installing (UNIX only)" on the previous page
- When running as a service, the setup wizard displays a dialog that states:

  ```
  The Connector Setup Wizard is not able to modify the service configuration because the Wizard is not running as root. Please run this Wizard as root. Or to manually install, logged on as root, execute the following script:
  $ARCSIGHT_HOME/current/bin/arcsight agentsvc -i -u user
  To manually remove the service, execute the following script as a root user:
  $ARCSIGHT_HOME/current/bin/arcsight agentsvc -r
  ```

We do not recommend running the wizard as *root*. Instead, run the wizard as *arcsight* user and then manually install the service. Execute the following script while logged on as root to install the connector as a service:

```
$ARCSIGHT_HOME/current/bin/arcsight agentsvc -i -u arcsight
```

The `-u arcsight` option ensures that the service runs as *arcsight* user.

Option 2: Install as *arcsight* user, run as *arcsight* user with port forwarding

This option allows a Syslog Daemon to receive events that are sent to ports below 1024. After installing ArcSight connector, run the connector setup wizard as *arcsight* user and use the procedure mentioned in "Option 1: Install as arcsight user, run as arcsight user" on the previous page. In addition, use another program that forwards traffic from a low number port to the port configured for the connector.

For example, if the syslog events are being sent to port 514 and the connector is configured to receive on port 6000, the forwarder re-routes from port 514 to port 6000. There are several programs that can do port forwarding including **iptables**, **ncat**, and **socat**. The **iptables** program is packaged with some versions of Linux/Unix.

Option 3: Install as *root* user, run as *root* user

This option is less secure than the other options as *root* privileges are required for installation, configuration, and maintenance of the connectors.

A user logs on to the system as *root* and installs the ArcSight connector. This results in all the ArcSight connector files to be owned by user *root*. The connector setup wizard is also run while logged on as *root*. If the connectors are to be run as a service, the service configuration is done by the connector setup wizard and no additional steps are required.

Caution: Avoid installing as user *arcsight*, and run as user *root.*

This can lead to security vulnerability issues. The potential problem with this option is that the connector configuration files are owned by user *arcsight* and so may be more susceptible to modification by a malicious user. Since the connectors are run as *root*, those modifications may result in undesirable privilege escalation.

When running in standalone mode

Option 1: Install as user *arcsight*, run as user *arcsight*

This is the recommended option. The following instructions refers to *arcsight* user as a generic name for any user with *non-root* privileges.

Note: When you install a connector as the *arcsight* user, the ArcSight connector files will be owned by *arcsight* user.

After installing ArcSight connector, run the connector setup wizard as *arcsight* user.

If a Syslog Daemon connector is selected, then the configured port number must be 1024 or greater for this option.

Option 2: Install as *arcsight* user, run as *arcsight* user with port forwarding

This option allows a Syslog Daemon to receive events that are sent to ports below 1024. After installing ArcSight connector, run the connector setup wizard as *arcsight* user and use the procedure mentioned in "Option 1: Install as arcsight user, run as arcsight user" on page 128. In addition, use another program that forward traffic from a low number port to the port configured for the connector. For example, if the Syslog events are being sent to port 514 and the connector is configured to receive on port 6000, the forwarder re-routes from port 514 to port 6000. There are several programs that can do the port forwarding including **iptables**, **ncat**, and **socat**. The **iptables** program is packaged with some versions of Linux/Unix.

Caution: Avoid installing connectors using the two following scenarios:

- as user *arcsight*, and run as user *root*

  This can lead to security vulnerability issues. The potential problem with this option is that the connector configuration files are owned by user *arcsight* and so may be more susceptible to modification by a malicious user. Since the connectors are run as *root*, those modifications may result in undesirable privilege escalation.

- as user *root* and run as user *root*

  This option is less secure since *root* privileges are required for installation, configuration, and maintenance of the connectors. A user logs on to the system as *root* and installs the ArcSight connector. This results in all the ArcSight connector files to be owned by user *root*. The connector setup wizard is also run while logged on as *root*.

Estimating storage requirements

Different devices generate different event volumes, and also respond differently to various event aggregation policies. Understanding the range of devices and connectors you want to deploy helps in estimating your daily event volume. Apart from the log file size, you also need to know how many events are generated during an average day.

In a distributed environment with multiple ESM Managers, the event volume metric must consider both the connector feeds to the ESM Manager and the event forwarding from other ESM Managers.

Connectors can also aggregate events to reduce event traffic.

The average size of the data stored for each event depends on the turbo mode specified for a particular connector.

Understanding the turbo mode

Turbo mode is used to vary the event data sent by connectors. This helps improve the performance of event data transfer from the connector to the ESM Manager. The ESM Manager can be set to read and maintain event data, independent of the connector setting.

Some events require more data than others. For example, operating system logs often capture a lot of environmental data that might not be relevant to a particular security event. On the other hand, Firewalls report only basic information.

The following are the different turbo modes available:

- Fastest (Mode 1)
- Faster (Mode 2)
- Complete (Mode 3)

Fastest (Mode 1):

This is the recommended mode for simpler devices, such as firewalls.

The Fastest mode eliminates a core set of event attributes to achieve the best throughput. Because, the event data is smaller, it requires less storage space to provide the best performance.

The Fastest turbo mode contains the following default fields:

agentReceiptTime, baseEventCount, category, destinationAddress, destinationTranslatedAddress, destinationGeo,destinationPort, destinationTranslatedPort, agent, device, endTime, eventId, name, type, generator, priority, rawEvent, sourceAddress, sourceGeo, sourcePort, sourceTranslatedAddress, sourceTranslatedPort, transportProtocol, startTime, managerReceiptTime, sourceZone, sourceTranslatedZone, destinationZone, destinationTranslatedZone, customer, originator, agentSeverity, bytesIn, bytesOut

If you want to add extra fields, then update the `agent.properties` file as follows:

```
turbo.field-list.mode-1.com.arcsight.event.SecurityEvent.includes=<Fastest turbo mode default fields list>, <append new field here>
```

For example, you can add the "deviceEventClassId" field as follows:

```
turbo.field-list.mode-1.com.arcsight.event.SecurityEvent.includes=agentReceiptTime, baseEventCount, category, destinationAddress, destinationTranslatedAddress, destinationGeo,destinationPort, destinationTranslatedPort, agent, device, endTime, eventId, name, type, generator, priority, rawEvent, sourceAddress, sourceGeo, sourcePort, sourceTranslatedAddress, sourceTranslatedPort, transportProtocol, startTime, managerReceiptTime, sourceZone, sourceTranslatedZone, destinationZone, destinationTranslatedZone, customer, originator, agentSeverity, bytesIn, bytesOut, deviceEventClassId
```

Faster (Mode 2)

The first level of turbo acceleration is called Faster. This drops the additional data by retaining all other information.

This is the default mode for ESM Manager.

Complete (Mode 3)

All event data received by the connector, including additional data, is maintained.

Complete is the default transfer mode. This mode passes all the data sent from a device, including any additional data (custom or vendor-specific). This corresponds to turbo.enabled=false on the Manager.

Make sure that you add this property to the `<ARCSIGHT_HOME>/config/server.properties` file, as this is not the default mode value for ESM Manager. After making changes to this file, you must restart the ESM Manager.

The specific event attributes that apply to all these turbo modes in your enterprise are defined in the `<ARCSIGHT_HOME>/config/server.default.properties` file for the ArcSight Manager. Because, these properties might have been adjusted in the corresponding server.properties file for your needs, you can refer to the server.properties file for definitive lists. For more information, see the Managing and Changing Properties File Settings section in the ESM Administrator's Guide.

The ESM Manager can also have its own turbo mode set to read and maintain specific event data, independent of the connector setting. This gives additional flexibility in event data collection. However, this leads to the following two scenarios:

1. The connector is set at a higher turbo mode than the ESM Manager. Which means the connector reports more event data than required by the ESM Manager. In this case, the ESM Manager ignores the extra event data.
2. The connector is set at a lower turbo mode than the ESM Manager. Which means the connector reports less event data than required by the ESM Manager. In this case, the ESM Manager maintains the fields that do not have event data.

Both scenarios are considered as normal occurrences in a practical situation.

Installing SmartConnectors

This chapter describes the different installation methods in which the SmartConnector can be installed. You can install connectors in GUI Mode, Console Mode, or Silent Mode.

Important: Before installing any connector, ensure that the random number pool (also known as entropy pool) of Operating System must not be less than the ideal lower limit of 3290. For more information, see SmartConnector Remote Connections Failing Due to Low Entropy.

Understanding installation parameters

The following sections provide information about the installation parameters and information that can help you select:

Global parameters

You can set the following optional global parameters, either during installation or after the installation:

Unique Generator ID

Global Parameter	Setting
Unique Generator ID	Connectors require a Unique Generator ID to generate unique GEIDs. Generator Ids cannot be encrypted. The valid Generator Id value is 1 to 16383. **Note**: If a value is not specified for Unique Generator ID, then events will not be processed when **Amazon S3** is configured as one of the destinations or if Recon mode is selected as the value for the **Check Event Integrity Method** parameter while configuring any destination.

FIPS Mode

Global Parameter	Setting
FIPS mode	Select **Enabled** to enable FIPS compliant mode. To enable FIPS Suite B Mode, see Enable FIPS Suite B Mode for instructions.

Remote Management from ArcSight Management Center

Global Parameter	Setting
Remote Management	Select **Enabled** to enable remote management from ArcSight Management Center.
Remote Management Listener Port	The remote management device will listen to the port specified in this field. The default port number is 9001. When queried by the remote management device, the values you specified here will be used.

Preferred IPV Version

Global Parameter	Setting
Preferred IP Version	If both IPv4 and IPv6 addresses are available for the local host, you can select the preferred version.

Format Preserving Encryption

Global Parameter	Setting
Format Preserving Encryption	Data leaving the connector machine to a specified destination can be encrypted by selecting **Enabled** to encrypt the fields identified in **Event Fields to Encrypt** before forwarding events. If encryption is enabled, it cannot be disabled. Changing any of the encryption parameters again will require a fresh installation of the connector.
Format Preserving Host URL	Enter the URL where the OpenText SecureData server is installed.
Proxy Server (https)	Enter the proxy host for https connection if any proxy is enabled for this machine.
Proxy Port	Enter the proxy port for https connection if any proxy is enabled for this machine.
Format Preserving Identity	The OpenText SecureData client software allows client applications to protect and access data based on key names. This key name is referred to as the identity. Enter the user identity configured for OpenText SecureData.
Format Preserving Secret	Enter the secret configured for OpenText SecureData to use for authentication.
Event Fields to Encrypt	Displays recommended fields for encryption. You can add or delete any fields for encryption. Encrypting more fields can affect performance, with 20 fields being the maximum recommended. Also, because encryption changes the value, rules or categorization might be affected. You cannot edit the event fields after you have enabled encryption.

Destination parameters

Depending on the destination selected, you might enter any of the following parameters:

ArcSight Manager (Encrypted)

Parameter	Description
Manager Hostname	This is the local host name, IP address, or fully-qualified domain name of the machine where the ArcSight Manager is installed. This name is what all clients (such as ArcSight Console) specify to talk to the Manager. Using a host name and especially a fully-qualified domain name instead of an IP address is recommended for flexibility. The Manager host name is used to generate a self-signed certificate. The Common Name (CN) in the certificate is the Manager host name that you specify in this screen. Although the Manager uses a self-signed certificate by default, you can switch to using a CA signed certificate if needed. See the *ESM Administrator's Guide* for more information.
Manager Port	8443
User	Enter a valid ESM User name.
Password	Enter the password for the ESM user.
AUP Master Destination	Default: **false**. A connector can send events to ESM and non-ESM destinations simultaneously. In this configuration, it is helpful to use the **AUP Master Destination** feature. See ArcSight Content AUPs for more information. **Note:** Set this to **True** for ESM to use zone information from the Manager for non-Manager destinations, such as SmartMessage (Logger) or Transformation Hub.
Filter Out All Events	Default: **false**. If **AUP Master Destination** is set to **true**, you may or may not want to send this connector's events to that Manager. If the Manager should not get the events, set this to **true**. In that case the manager will only be used as a source of zone information. An example of when this would be a useful case is if the connector is sending events to the Transformation Hub, and ESM is reading those events from there.
Enable Demo CA	Default: **false** The ArcSight Manager host name is used to generate a self-signed certificate during ArcSightESM installation. The Common Name (CN) in the certificate is the Manager host name that you specified during ESM installation. Do not use demo SSL certificates in production. Make sure when switching that you remove the demo CA from **cacerts** on all SmartConnectors and ArcSight Consoles.

ArcSight Logger SmartMessage (encrypted)

Parameter	Description
Host Name/IP	The destination host name or IP address.
Port	The destination port 443 for Logger Appliance or 9000 for Software Logger.
Receiver Name	The destination receiver name.
Compression Mode	Select to enable data compression. Default is **Disabled**.
CEF Version	Select any of the following options: • **0.1** - The Device Address, Source Address, Destination Address, and Agent Address fields will always be IPv4 or will be omitted. If there are any IPv6 addresses, they will be in Device Custom IPv6 Address fields. The **Bytes In** and **Bytes Out** fields are limited to the size of an integer (up to 2^31-1). Select **0.1** if you are not sure the chosen destination can handle CEF 1.0, which supports both IPv4 and IPv6 modes. • **1.0** - Any of the address fields can be either IPv4 or IPv6 and the **Bytes In** and **Bytes Out** fields can be long values (up to 2^63-1).

ArcSightLogger SmartMessage Pool (encrypted)

Parameter	Description
Host Name/IP	The destination host name or IP address.
Port	The destination port 443 for Logger Appliance or 9000 for Software Logger.
Receiver Name	The destination receiver name.
Compression Mode	The data compression mode checkbox. Select to enable or leave as default for disable.
CEF Version	Select **0.1** or **1.0** from the drop-down menu. Select 0.1 if you are not sure the chosen destination can handle CEF 1.0, which supports both IPv4 and IPv6 modes. 0.1 - The Device Address, Source Address, Destination Address, and Agent Address fields will always be IPv4 or will be omitted. If there are any IPv6 addresses, they will be in Device Custom IPv6 Address fields. Bytes In and Bytes Out fields are limited to the size of an integer (up to 2^31-1). 1.0 - Any of the address fields can be either IPv4 or IPv6 and the Bytes In and Bytes Out fields can be long values (up to 2^63-1).

Button	Description
Add	Adds a row to the table to add a logger to a pool. Fill in the information manually. Use the checkbox for **Compression Mode** to enable or disable it. The default is unchecked for disabled. The default port for logger is 443.
Remove	Removes the row corresponding to the logger from the loggersecure pool.
Import	Opens a dialog window to import the .csv file type containing the pre-recorded information for loggersecure pool.
Export	Opens a dialog window where you can export and save the data entered in the panel. Use a .csv file extension for export. The file lists Disabled for default Compression Mode and TRUE for enabled.

ArcSight SaaS

Parameter	Description
Registration URL	Specify the Registration URL to register for ArcSight SaaS service. You must modify the registration URL only by using the **Re-register destination** option, if required. For more information, see "Re-registering a destination" on page 168.

Modifying ArcSight SaaS Default Parameter

You can use the following default parameter to refine or elaborate the way the connector works with the **ArcSight SaaS** destination.

To modify the ArcSight SaaS default parameter:

1. Go to `ArcSight_home>\config\agent\agent.default.properties` file.
2. Copy the parameter line that you want to modify to your `agent.properties` file.
3. Modify the value of the following parameter as required:

Parameter	Description
transport.arcsightsaas.max.wait.time.in.minutes=5	The maximum period of time a connector can take to retry the failed API requests.

4. Save the file and restart the connector.

Transformation Hub

For information about Configuring a SmartConnector as a Transformation Hub Producer, refer to Administrator's Guide to ArcSight Platform available on the ArcSight Documentation site.

Parameters	What to specify or select
Kafka Broker Host(s):Port(s)	This is a mandatory field. You must specify at least one server. If there are multiple servers, then specify a comma-separated list of hostnames and ports to establish a communication with the Transformation Hub cluster. While it is not necessary to list all servers in the cluster, listed, if none of the servers listed can be contacted, the Connector cannot send events to Transformation Hub. For example: `kafka1.example.com:9093,kafka2.example.com:9093`.
Kafka Broker on SSL/TLS	Determines whether events are sent with TLS encryption. Select one of the following options: • **false** - (default) • **true** - Select **true** to access the Kafka broker on SSL/TLS. If you select **true** , you must provide the **SSL/TLS Truststore Password** and the location of the **SSL/TLS Truststore File Path**. When **Kafka Broker on SSL/TLS** is set to **true**, a secure connection will be established with the Kafka broker(s) specified in the **Kafka Broker Host(s):Port(s)** field. **Note**: If you want to set the **Kafka Broker on SSL/TLS** parameter to **true**, refer to the ArcSight Platform admin guide for instructions on performing the certificate trust exchange between the SmartConnector and Transformation Hub for the secure connection to work properly.
TH User Name **TH Password**	Specify the user name and password of the TH server to connect to the server over SSH or SCP. Connector connects to the TH server to fetch the server certificate and import into the truststore of the Connector, copies the Certificate Signing Request (CSR) to the server and gets the CSR signed.
Receive Acknowledgment	This is a mandatory field. Select a value to determine if and how the Connector waits for acknowledgment from Transformation Hub that it has received the event. Select one of the following options: • **none**: Default. The Connector does not wait for acknowledgment. This can result in lost events if the receiving Kafka server fails. However, selecting this option provides a significantly higher throughput. • **leader**: The Connector waits for acknowledgment from the primary Transformation Hub server for the event's partition. This option protects against data loss in most circumstances while providing reasonable performance. However, selecting this option can affect the throughput. • **all**: The Connector waits for an acknowledgment from all Transformation Hub servers that contain a backup for the event's partition. This protects against lost events in nearly all circumstances, but significantly reduces throughput.

<table>
<tr><th>Parameters</th><th>What to specify or select</th></tr>
<tr><td>Content Format
Kafka Topic</td><td>Select any of the following topics for the corresponding content format:<table><tr><th>Content Format</th><th>Kafka Topic</th></tr><tr><td>Avro</td><td><code>th-arcsight-avro</code>
Supports ArcSight 2020.3 or later. Supports Avro events to be sent to Transformation Hub.
Note: ArcSight 2020.3 refers to the third release of ArcSight in the year 2020.</td></tr><tr><td>CEF (for IPv4)</td><td><code>th-cef</code>
Supports IPv4. Use with Logger 6.3.0 or later versions.
Selecting CEF (for IPv4) allows to configure content format for Logger/Investigate/Hadoop/3rd parties.</td></tr><tr><td>CEF (for IPv4 and IPv6)</td><td><code>th-cef</code>
Supports IPv4 and IPv6. Use with Logger 6.4.0 or higher versions. In addition to IPv6 support, this option adds support for long values for Bytes In/Out fields.
Selecting CEF (for IPv4 and IPv6) allows to configure content format for Logger 6.4 or higher/IPv6/Investigate.</td></tr><tr><td>ESM Binary</td><td><code>th-binary_esm</code>
Supports all versions of ESM.
For more information, see the Support Matrix for ArcSight ESM guide, available on the ArcSight Enterprise Security Manager (ESM) Documentation page.
Selecting ESM Binary allows to configure content format for ESM.</td></tr></table>Note: The default Content Format is CEF (for IPv4 and IPv6) and Kafka Topic is <code>th-cef</code>. However, you can change the content format as required.</td></tr>
<tr><td>Compression Type</td><td>Compression reduces disk space and network bandwidth requirements.
Select the compression algorithm used (gzip, zstd, none) when Transformation Hub copies events, such as when routing events between Topics.
gzip - is the default value.
Note: The zstd algorithm performs better than gzip, but requires Kafka client library version 2.1.0 or later.
Zstd - only is supported in Transformation Hub 3.3 and SmartConnector 8.0.0. If your Transformation Hub version is 3.2, use gzip as a compression type. This compression type works only for Logger 7.0, ESM 7.2, IDI 1.1, or their later versions.</td></tr>
<tr><td>ESM Version for ESM Topic</td><td>Select the ESM version number of the desired ESM topic. If you do not select any value, the latest version of ESM is considered.
This field is mandatory when the Content Format is selected as ESM Binary.</td></tr>
</table>

Parameters	What to specify or select
Schema Registry Host:Port	Specify the host:port of the Schema Registry node to fetch schema using HTTPS. Use the FQDN or the IP address for the Virtual IP of the master node of the Transformation Hub to achieve high availability. In this case, if the primary node fails, the Virtual IP will automatically migrate to a failover master node and the connector will still be able to access the schema registry without having to reconfigure the connector. If Transformation Hub is configured with only a single master node, use the FQDN or IP address of that master node. Use **32081** as the port unless it is customized in your environment. **Note**: For an AWS environment, use the cluster `DNS hostname:32081`. This field is mandatory when the **Content Format** is selected as **Avro**.
SSL/TLS Truststore File Path	Specify the location of the SSL/TLS truststore file. This is required to access HTTPS Schema Registry for Avro or the TLS-based secure communication for the Kafka brokers. It is optional for Text-based communication with Kafka brokers. This field is mandatory when the **Content Format** is selected as **Avro** or when **Kafka Broker on SSL/TLS** is set to **true**.
SSL/TLS Truststore Password	Specify the password for the SSL/TLS truststore file. This field is mandatory when the **SSL/TLS Truststore File Path** is specified.
Use SSL/TLS Client Authentication	Determines whether a client certificate is used for TLS to identify the Connector. Select one of the following options: • **false** - (default) • **true** - Select **true** if client authentication is enabled for Kafka broker, Schema Registry, or both. If you select **true**, ensure that the **Kafka Broker on SSL/TLS** is enabled. You must also provide values for the **SSL/TLS Keystore File Path**, **SSL/TLS Keystore Password**, and **SSL/TLS Key Password** parameters. **Note**: If you want to set the **Use SSL/TLS Client Authentication** parameter to **true**, refer to the Administrator's Guide to ArcSight Platform for instructions on performing the certificate trust exchange between the SmartConnector and Transformation Hub for the secure connection to work properly.
SSL/TLS Keystore File Path	Specify the location of the SSL/TLS keystore file path for client authentication.
Organizational Unit (OU)	Specify the name of your organizational unit.
Organization (O)	Specify the name of your organization.
Location (L)	Specify the name of your city or locality.
State (ST)	Specify the name of your state or province.
Country (C)	Specify the two-letter country code for this unit.

Amazon MSK

Parameter	Description
MSK Broker Host (s):Port(s)	Specify the Amazon Managed Streaming for Apache Kafka (Amazon MSK) details as <Fully qualified host name:port number>.
MSK User Name	Specify the user name to connect to Amazon MSK <Your MSK user name>.
MSK Password	Specify the password for the user name.
Require Acknowledgment	Select an option to determine if and how the connector waits for an acknowledgment of receipt of records from Amazon MSK. If you choose to receive the acknowledgment, Amazon MSK stores the data. However, this might affect the performance. Select one of the following options: • **leader**: Default. The Connector waits for acknowledgment from the Amazon MSK server for the event's partition. This option protects against data loss in most circumstances while providing reasonable performance. However, selecting this option can affect the throughput of a connector. • **none**: The Connector does not wait for acknowledgment. This might result in loss of events if the receiving Amazon MSK server fails. However, selecting this option provides a significantly higher throughput of a connector. • **all**: The Connector waits for an acknowledgment from all Amazon MSK servers that contain a backup for the events partition. This protects against loss of events in nearly all circumstances, but significantly reduces throughput.
MSK Topic	Specify the topic name.

Amazon S3

Parameters	Description
Avro File Storage Path	The path to the location where the Avro files will be stored.
File Rotation Interval (Sec)	The desired file rotation interval, in seconds. The default value is 3,600 seconds (one hour). The maximum value is 36,000 seconds (10 hour).
Number of Events in a File	The number of events that can be stored in each Avro file. The default value is 5,000. The maximum number is 50,000.
Proxy Host	If proxy is enabled for your machine, the IP address or host name of the proxy server for HTTPS connection.
Proxy Port	If proxy is enabled for your machine, the port number of the proxy server for HTTPS connection.

Parameters	Description
Proxy User Name	If proxy is enabled for your machine, the user name for the proxy server. This value is optional for additional proxy authentication. If you enter the proxy user name, you must provide the proxy password.
Proxy Password	If proxy is enabled for your machine, the password for the proxy server user.
Default AWS Credentials Provider	If set to **true**, the connector will use the Default Credential Provider Chain. The default value is **false**.
Amazon Access Key	The access key that is used to access Amazon S3. This parameter is not applicable if the **Default AWS Credentials Provider** parameter is set to **true**.
Amazon Secret Key	The secret key that is used to access Amazon S3. This parameter is not applicable if the **Default AWS Credentials Provider** parameter is set to **true**.
Amazon S3 Bucket Name	The name of Amazon S3 bucket that is created on the Amazon account to which the Avro output files will be sent.
Amazon S3 Bucket Folder Name	The name of the folder in the Amazon S3 bucket. This is an optional field. **Note**: If the folder is not present in the Amazon S3 bucket, then it will be automatically created with the name specified in this field.
Amazon S3 Region Code	The Amazon S3 region code in which the Amazon S3 bucket was created on Amazon account with the name specified in the **Amazon S3 Bucket Name** field.

Note: To use the Default Credential Provider Chain for **Amazon Access Key** and **Amazon Secret Key**, see AWS Credentials.

Amazon S3 Default Parameters

You can use the following default parameters to refine or elaborate the way the Connector works with the **Amazon S3** destination.

To modify the Amazon S3 default parameters:

1. Go to `ArcSight_home>\config\agent\agent.default.properties` file.
2. Copy the property line that you want to modify to your `agent.properties` file.

3. Modify the values of the following parameters as required:

Parameter	Description
transport.avroawss3.file.s3done.retention.days=5	Modify this parameter to increase the maximum number of days for which the Avro output file will be retained in the Amazon S3 destination.
transport.avroawss3.file.event.max.limit=10000	Modify this parameter to increase the number of events that will be saved to the Avro output file.
transport.avroawss3.file.upload.interval.minutes=5	Modify this parameter to increase the time interval in minutes between each upload of the Avro output file to the Amazon S3 destination.

4. Save the file and restart the connector.

Microsoft Azure Event Hub

For information about Configuring the Azure event Hub SmartConnector , refer to the Configuration guide for Microsoft Azure Event Hub available on the ArcSight Documentation site.

Parameter	Description
Acknowledgment mode	Select a value to determine if and how the Connector waits for acknowledgment from Microsoft Azure Event Hub that it has received the event. Select one of the following options: • **none**: Default. The Connector does not wait for acknowledgment. • **leader**: Acknowledgement is sent by the broker when the message is successfully written on the leader. • **all**: Acknowledgement is sent by the broker when the message is successfully written on all replicas.
Application (Client) ID	Enter the Client ID generated for your registered application. For this value, refer to the Overview section of the registered application.
Bootstrap servers	Include: {Your event hub namespace}.servicebus.windows.net:9093

Parameter	Description
CEF Version	Select one of the following options: • **0.1** - The Device Address, Source Address, Destination Address, and Agent Address fields will always be IPv4 or will be omitted. If there are any IPv6 addresses, they will be in Device Custom IPv6 Address fields. The **Bytes In** and **Bytes Out** fields are limited to the size of an integer (up to 2^31-1). Select **0.1** if you are not sure the chosen destination can handle CEF 1.0, which supports both IPv4 andIPv6 modes. The destination could be Logger, another SmartConnector, or a non-ArcSight product. • **1.0** - Any of the address fields can be either IPv4 or IPv6 and the **Bytes In** and **Bytes Out** fields can be long values (up to 2^63-1).
Client Certificate	Specify the client certificate path. This field is mandatory if the Credential Type is Client Certificate. For detailed information, see Troubleshooting section of the Configuration Guide of Microsoft Azure Event Hub Connector.
Client Certificate Password	Enter the password of client certificate.
Client Id	Specify a unique identifier to be assigned to the client.
Client Secret	Enter the client secret value generated while registering the application. This value is obfuscated. This field is mandatory if the Credential Type is Client secret. For detailed information, see Troubleshooting section of the Configuration Guide of Microsoft Azure Event Hub Connector.
Credential Type	If Client secret is selected, then client secret will be used to authenticate the app. If Client certificate is selected, then client certificate will be used to authenticate the app
Directory (tenant) ID	Enter the Directory (tenant) ID of your registered application. For this value, refer to the Overview section of the registered application.
Request timeout	The timeout of the request, make sure that your request.timeout.ms reaches at least the recommended value of 60000.
Topic	The name of the event hub on your namespace.

CEF File

Parameter	What to enter or select
CEF Folder	Path where the CEF files are stored
File Rotation Interval	The desired file rotation interval, in seconds. The default is **3,600** (one hour).

Parameter	What to enter or select
File Size	File size in megabytes (default: 10 MB)
CEF Version	Select **0.1** or **1.0** from the drop-down menu. Select 0.1 if you are not sure the chosen destination can handle CEF 1.0, which supports both IPv4 and IPv6 modes. The destination could be Logger, another SmartConnector, or a non-ArcSight product. 0.1 - The Device Address, Source Address, Destination Address, and Agent Address fields will always be IPv4 or will be omitted. If there are any IPv6 addresses, they will be in Device Custom IPv6 Address fields. Bytes In and Bytes Out fields are limited to the size of an integer (up to 2^31-1). 1.0 - Any of the address fields can be either IPv4 or IPv6 and the Bytes In and Bytes Out fields can be long values (up to 2^63-1).

CEF Syslog

Parameter	What to enter or select
IP/Host	Enter the IP/ Host information.
Port	Enter the Port information.
Protocol	Select the appropriate protocol from the drop-down menu.
Forwarder	The default value is **False**. If the destination is a Syslog Daemon connector and you want to preserve information about the original connector, then the **CEF Forwarder** mode should be set to **True** both in this destination and in the receiving connector. In other words, if you have a chain of connectors connected by Syslog, Syslog NG, or CEF encrypted Syslog (UDP), and you want to preserve information about the original connector, the destinations should all have the **CEF Forwarder** mode set to **True** (which is implicitly true for CEF Encrypted Syslog (UDP)), and the connectors receiving from them should also have the CEF Forwarder mode set to **True**.
CEF Version	Select any of the following options: • **0.1** - The Device Address, Source Address, Destination Address, and Agent Address fields will always be IPv4 or will be omitted. If there are any IPv6 addresses, they will be in Device Custom IPv6 Address fields. The **Bytes In** and **Bytes Out** fields are limited to the size of an integer (up to 2^31-1). Select **0.1** if you are not sure the chosen destination can handle CEF 1.0, which supports both IPv4 and IPv6 modes. The destination could be Logger, another SmartConnector, or a non-ArcSight product. • **1.0** - Any of the address fields can be either IPv4 or IPv6 and the **Bytes In** and **Bytes Out** fields can be long values (up to 2^63-1).

CEF Encrypted Syslog (UDP)

Caution: Logger does not accept CEF Encrypted Syslog.

Parameter	What to enter or select
IP/Host	Enter the IP/Host.
Port	Enter the Port information.
CEF Version	Select one of the following options: • **0.1** - The Device Address, Source Address, Destination Address, and Agent Address fields will always be IPv4 or will be omitted. If there are any IPv6 addresses, they will be in Device Custom IPv6 Address fields. The **Bytes In** and **Bytes Out** fields are limited to the size of an integer (up to 2^31-1). Select **0.1** if you are not sure the chosen destination can handle CEF 1.0, which supports both IPv4 and IPv6 modes. The destination can only be the corresponding SmartConnector. • **1.0** - Any of the address fields can be either IPv4 or IPv6 and the **Bytes In** and **Bytes Out** fields can be long values (up to 2^63-1).
Shared Key (16 characters)	Enter a 16 character shared key for encryption (Shared Secret). The same Shared Key must be used when configuring the CEF Encrypted Syslog (UDP) connector on the receiving side.

CSV File

Parameter	What to enter or select
CSV Path	The path to the output folder. If it does not exist, the folder is created.
Fields	A comma-delimited list of field names to be sent to the CSV file. The default is: **event.deviceReceiptTime,event.name,event.deviceAddress, event.deviceHostName,event.sourceAddress, event.sourceHostName,event.sourcePort, event.destinationAddress,event.destinationHostName, event.destinationPort** To modify the list, each entry needs to begin with either: • “event.” and the name of a normal pre-defined event field, or • “additionaldata.” and the name of some additional data field that applies to this particular connector. These names are not common across all connectors. There are no spaces allowed around the commas in the field names. For example: “event.deviceReceiptTime,event.name” is correct. But, “event.deviceReceiptTime, event.name” is not correct.
File rotation interval	Enter the desired file rotation interval, in seconds. The default is **3,600** (one hour).
Write format header	Select **true** to send a header row with labels for each column, as described above.

Raw Syslog

Parameter	Description
IP Host	Enter the IP address or host name to which the connector is to send events.
Port	Specify the port to which the connector is to send events.
Protocol	Select either UDP, Raw TCP, or TLS as the protocol to be used by the co nnector to send events. The default value is UDP.
Enable Metadata for Logger	Select either true or false. If you select **true**, metadata about the source and time stamp is included in the outgoing message for ArcSight Logger, though you should only select this if you previously choose a level other than **None** for the **Metadata Capture Level** parameter.

Installing and configuring SmartConnectors by using the wizard

You can install and configure the SmartConnectors by using the installation wizard.

Installing the Core Software

1. Download the SmartConnector executable for your operating system from the OpenText SSO site.
2. Double-click the executable to start the installation wizard.
3. Specify the installation folder, and then click **Next**.
4. Specify a location to display product icon, and then click **Next**.
5. Review the pre-installation summary, and then click **Next** to install the core software.
6. You can either exit from the wizard or proceed with Configuring the SmartConnectors.

Configuring the SmartConnector

1. (Optional) If you have exited from the installation wizard after the installation of core software, then run the `<install_directory>\current\bin\runagentsetup.bat` file.
2. Configure any of the relevant global parameters:

- Unique Generator ID.
- Enable FIPS Mode.
- Remote Management by using the ArcSight Management Center.
- Preferred IP version if both IPV 4 and IPV 6 are both available for the local host.
- Format Preserving parameters, if you are using OpenText SecureData solutions to provide encryption.

  For more information, see *OpenText SecureData Architecture Guide*.

3. Click **Next** to confirm the connector setup parameters.
4. In the **Type** list, select the type of connector to install.
5. Specify the parameters for the selected connector. For configurations specific to the connector, see the configuration guide specific to that connector.
6. Select a destination and configure parameters.
7. Configure the relevant destination settings.
8. Specify a connector details such as name and other information identifying the connector's use in your environment, and then click **Next**. The connector starts the registration process.
9. In the Add connector Summary window, click **Next**.

Completing Installation and Configuration

1. Select one of the following options, when prompted to select a mode to run the SmartConnector:
   - To run the connector as a stand-alone process, select **Leave as a standalone application**, click **Next.**
   - To run the connector as a service:
     a. Select I**nstall as a service**, and then click **Next**.
     b. Specify values for **Service Internal Name** and **Service Display Name**.
     c. Select **Yes** or **No** for **Start the service automatically**.
2. Complete the installation process by following the instructions on the wizard.

Installing SmartConnectors From the Command Line

To install SmartConnectors without using the graphical user interface wizard, enter `-i console` on the command line when you invoke the self-extracting archive.

1. Download the ArcSight SmartConnector build.
2. Go to the folder in which the connector is located and run the binary file command:

   ```
   ./ArcSight-x.XXXX.bin –i console
   ```

3. Press **Enter** to continue.
4. Specify the installation location. The default installation location is the root folder: `/root/ArcSightSmartConnectors`.
5. Follow the instructions in the command window.
6. When the installation completes, go to `$_ARCSIGHT_HOME/current/bin` and run the following command:

   ```
   ./runagentsetup
   ```

   Note: $_ARCSIGHT_HOME is the path where the connector is installed.

   You can choose if you want to hide sensitive data.
7. Enter the parameters. A summary is displayed.

   You can type **Y** to confirm changes or **N** to go back to modify the parameters.

   If you did not enter a Generator ID, a message is displayed. Click **Yes** to proceed with the installation.
8. Select the connector you want to install. If it is not shown on the first screen, click **N** to see more connectors.
9. Select a destination and configure parameters.
10. Name the connector and its location.
11. Select if you want to install the connector as a service or as a standalone application.
12. Click **Continue** to change any values or **Exit** to complete the installation.
13. To start the SmartConnector, go to `$_ARCSIGHT_HOME/current/bin` and run the following command:

    ```
    ./arcsight agents
    ```

To change the remote management password from the Command Line

The following steps must be implemented to change the remote management password from the Command Line:

1. Delete the `remote.management.password.hashed` from `agent.properties` if present.
2. Add the following property to `agent.properties`:

   `remote.management.password=new password`
3. Run the `runagentsetup`.

   Connector will update the `connector_config.xml` with new password. The following properties will get added to the `agent.properties` file:

   `remote.management.password.hashed`

   `remote.management.user`

Note: Hashed property must not be added directly to the `agent.properties`.

Installing SmartConnectors on Solaris using Java

Before you install SmartConnectors on Solaris by using Java, ensure the following:

1. Download the appropriate JDK version before installing the connector on Solaris using Java. To download JDK, see https://www.azul.com/downloads/#zulu. JDK is not bundled with Solaris binary.

   Note:

   - The Java version must be compatible with the SmartConnector version. For example, SmartConnector 8.4.0 does not work with Java 392. Use the same Java version as the release. For more information on the Java and SmartConnector version compatibility, see the Compatibility Matrix of Java and SmartConnector Version section of the Technical Requirements Guide for SmartConnectors.
   - If you are installing SmartConnectors 8.4.1P1 or later, the Java version must be 362 or higher. For these versions, the removal of the `legacy8ujsse` jar and security file will impact the functionality of the connector. To fix this issue:
     a. Download the JDK version 8u362.
     b. Locate the `legacy8ujsse.jar` file in the `jre/lib/ext` folder and the `legacy8ujsse.security` file in the `jre/lib/security` folder within the JDK directory.
     c. Copy and paste the files in the corresponding directories present in the JDK directory that is being used to install the SmartConnector on Solaris by using Java.
2. Multiple versions of the Java platform are present simultaneously on a Solaris system (using the default Solaris package installations), but only one can be the **default** Java platform. The default Java platform is defined by the directory that the `/usr/java` symbolic link points to. Run the following commands to modify the symbolic link, so that it will point to the appropriate JDK:

```
ls -ld /usr/bin/java
```

```
rm /usr/bin/java
```

```
ln -s {path_of_your_installed_jdk}/bin/java /usr/bin/java
```

3. Run the following command to verify that the default Java on the system is the same as it was set in previous steps:

```
java -version
```

The connector is now ready for installation.

Upgrading SmartConnectors on Solaris using Java

The procedure for upgrading the SmartConnector on Solaris by using Java is the same as the procedure for installing the connector on Solaris by using Java. Ensure that the Java version matches the upgrade version of the SmartConnector.

Installing the SmartConnectors in Silent Mode

You can record the installation parameters in a properties file, which can be used later to install the connectors in an unattended mode. This feature is useful while deploying multiple identical connectors.

To use this feature, you must first install and configure one connector using the installation wizard or the command line and record the selected parameters in a properties file.

Tip: ArcSight recommends creating and testing the Properties file on a system other than your in-service, production environment.

This section contains the following topics:

Recording the Configuration parameters

To record the configuration parameters into a properties file:

Note: Ensure that the connector is in GUI mode before you perform step 1 and step 2.

1. Run the SmartConnector Configuration Wizard to extract and install the core files.
2. When you are prompted to select the **Add a Connector** or **Set Global Parameters** options, click **Cancel**.
3. Open a command prompt, browse to `ARCSIGHT_HOME\current\bin directory`, then enter the following command to launch the SmartConnector configuration wizard in record mode:
   - **On Unix and Linux:** `./runagentsetup.sh –i recorderui`
   - **On Windows:** `runagentsetup.bat -i recorderui`
4. In the **Silent Properties File Name** field:
   a. Enter the path or click the browse button and choose a location to create and store a silent properties file.
   b. Append **\<enter name of silent properties file.properties>** to the selected path (Example: C:\ArcSight\folder**\silentpropertyfile.properties**).
   The file will be created in the selected location with the specified name.
5. In the **Installation Target Folder** field, enter the installation target folder path or click the browse button and select the folder location where you want to install the connector.
6. Continue through all SmartConnectorConfiguration Wizard windows. The wizard creates a Properties file using the name and location you specified.
   If you do not enter a value for Unique Generator ID, when you move on to create the silent-properties file, the value of `containeroptionsconfig.agent.generator.id` will be empty.
7. Select **Exit** and click **Next** at the end of the setup process to ensure that the properties file is created.

Note: The properties file that you create will show passwords in readable text.

Setting Generator Id while installing in Silent Mode

While installing in silent mode you can set Generator Id in one of the following ways:

- When creating the Silent Properties file on the Configuration Wizard, you must enter the correct value on the **Unique Generator Id** text box.
- If you did not enter a Generator Id while recording the silent properties file, then open the silent properties file, find the parameter **containeroptionsconfig.agent.generator.id** in the file and enter a valid value between 1 to 16383.
- While setting up the SmartConnector in silent mode, to add Generator Id, complete the following steps:

a. From the command prompt, move to the bin directory of the SmartConnector core.

b. Run the following command:

```
% ARCSIGHT_HOME% \current\bin> arcsight agentsetup -c -i silent -f
<silent template> -gi <generator id>
```

> **Note**: The gi parameter differences the **Unique Generator Id** from the one in the silent template. The values will only be differentiated when the installation on the connector core is extracted from the connector installer file.

Using the Properties file for unattended installation

Perform the following steps on the system on which you want to install the SmartConnector in silent mode:

1. Copy the **Properties** file to the system on which you want to install the connector, preferably to the same directory where you downloaded the installation file.

   > **Note**: Ensure that the configuration on the system on which you want to install the SmartConnector in silent mode matches that of the machine on which you created the properties file. Otherwise, the installation fails.

2. Open the **Properties** file, locate the `USER_INSTALL_DIR` property and ensure that the path value is the **absolute** path to the location where you want to install the current system.

   - **For Linux**: `USER_INSTALL_DIR=/opt/ArcSight_syslog`
   - **For Windows**: `USER_INSTALL_DIR=C\:\\Program Files\\ArcSightSmartConnectors`

   > **Note:** The colon (:) and backslash (\) characters must be preceded by a backslash (\).

3. Find the `ARCSIGHT_AGENTSETUP_PROPERTIES` property in the file and make sure that the path value is the **absolute** path to the location where you copied the **Properties** file on this system.

   For example, if you copied the **Properties** file to `C:\properties_files\silent.properties`, then the path value must be as follows:

   - **For Linux**: `ARCSIGHT_AGENTSETUP_PROPERTIES=/opt/Silent_properties/syslog.properties`
   - **For Windows**: `ARCSIGHT_AGENTSETUP_PROPERTIES=C\:\\properties_files\\silent.properties`

4. Modify the properties as required. For example, modify the `connectordetails.name` property in the file and change its value to the name of the SmartConnector you are going to install in silent mode. The following is an example of a properties file:

```
#================================================
# Panel 'connectordetails'
#================================================
# Enter the connector details.
#
# Name
connectordetails.name=The Name
# Location
connectordetails.location=The Location
# DeviceLocation
connectordetails.devicelocation=The Device Location
# Comment
connectordetails.comment=The Comment
#================================================
```

Modify any property in the Properties file if needed

Definitions of properties:

- **connectordetails.name**: The name of the connector in ESM.
- **connectordetails.location:** The name of the folder that contains the connector in ESM.
- **connectordetails.devicelocation:** The location of the machine on which ESM is installed.
- **connectordetails.comment:** Comments that were added about the connector.

5. Save the **Properties** file.
6. Download the SmartConnector installation file appropriate for your platform.
7. Run the following command to install the new SmartConnector in silent mode:

   ```
   ArcSight_Agent_install_file -i silent –f <path_to properties_
   file>\properties_filename
   ```

   The command launches the InstallShield program and installs the SmartConnector Appliance silently.

   Example: To install a SmartConnector with the property file name as **silent_properties**, run the following command:

   - **For Linux platform**: `./ArcSight-7.11.0.8139.0-Connector-Linux64.bin -i silent -f /home/arcsight/ silent_properties`
   - **For Windows platform**: `ArcSight-3.5.x.nnnn.y-Agent-Win.exe –i silent –f C\:\\Program Files\\silent_properties`

Note: After running the silent install, the original command in the `runagentsetup.bat` file is modified after specifying the Silent Install answer file.

To correct the problem, manually edit and remove the entries between the double quotation marks (" ") and return to the default setting. There should be no entries between the second double quotation marks (" "). Here is an example of the script before modifying:

```
call arcsight.bat agentsetup -c -i "SILENT" -f "C:\ArcSight\silent_properties_AD" %*
```

An example of the modified script as follows:

```
call arcsight.bat agentsetup -c -i "SWING" -f "" %*
```

To avoid this issue:

Extract first and use the `silent_properties` file to configure. Run the command similar to following:

```
<connector_installpath>\current\bin\arcsight.bat agentsetup -c -i silent -f 2_addwinc
```

Then, the `runagentsetup.bat` file would not contain the `silent_properties` and the path will be correct.

Caution: It is important to know:

- After installing SmartConnector, configure your system's default file permissions so that files created by ArcSight (events, log files, and so on) are reasonably secure.
- On UNIX systems, file permissions typically are set by adding the `umask` command to your shell profile. An umask setting of 077, for example, would deny read or write file access to any but the current user. An umask setting of 000 creates an unnecessary security hole.

Instant Connector deployment from ArcMC

The Instant SmartConnectors Deployment feature in ArcMC simplifies the installation process for enterprise customers who deploy to a high number of servers and install multiple connectors per server. All the installation information is captured, deployed, and installed through remote silent installation mode to many target nodes through ArcMC. It does not require a connector to be previously installed. For more information, see the *ArcSight Management Center Administrator's Guide*.

Note: If you install the connector on a virtual machine, enable NTP for both host and guest systems to ensure proper timekeeping. For more information, see Timekeeping best practices for Linux guests (1006427) on the VMWare Customer Connect website.

Before you begin with the SmartConnector deployment, ensure that you perform the prerequisites for the latest version of OS Rocky Linux, that is, 8.6 and also for OS from RHEL 8.X.

For more information, see the *Prerequisites for Instant Connector Deployment* section of Configuration Guide for ArcSight Management Center Help .

Post-Installation configuration

The installation folder stores raw event data, which contains sensitive information. For the optimal security of your sensitive data, it is recommended that you restrict the permissions of the installation folder to authorized users only such as the install user and the system administrators.

For more information related to the folder permissions and how to set them, refer to the Microsoft Documentation.

Running SmartConnectors

SmartConnector can be run in stand-alone mode or as a service, depending on the mode selected during installation.

Caution: Some SmartConnectors require that you restart your system before the configuration changes take effect.

To run a scanner SmartConnector in interactive mode, run in standalone mode and *not* as a Windows service or UNIX daemon.

To verify that a connector is running, you can check the ArcSight Console Navigator in the **Resources** tab, under **Connectors**. If the connector is running, you will see ***<connector_name>* (running)** listed.

Running in standalone mode

If the connector is installed in stand-alone mode, it must be started manually and is not automatically active when a host is restarted.

- To run all SmartConnector installed in stand-alone mode on a particular host, open a command window, go to the `$ARCSIGHT_HOME\current\bin` directory and run the following command:

  ```
  arcsight connectors
  ```

- To view the SmartConnector log, read the file:

  ```
  $ARCSIGHT_HOME\current\logs\agent.log
  ```

- To stop all SmartConnectors, enter `Ctrl+C` in the command window.

Running as a Windows Service

- **To start or stop SmartConnectors installed as services on Windows platforms:**
  a. Right-click **My Computer**, then select **Manage** from the **Context** menu.
  b. Expand the **Services and Applications** folder and select **Services**.
  c. Right-click the SmartConnector service name and select **Start** to run the SmartConnector or **Stop** to stop the service.
- To verify that a SmartConnector service has started, view the following file:
  `$ARCSIGHT_HOME\logs\agent.out.wrapper.log`

- To reconfigure a SmartConnector as a service, run the SmartConnector Configuration Wizard again. Open a command window on `$ARCSIGHT_HOME\current\bin` and run:
  `runagentsetup`

Running Connectors as a UNIX Daemon

Connectors installed as a daemon can be started and stopped manually by using platform-specific procedures.

On UNIX systems, when you configure a SmartConnector to run automatically, ArcSight creates a control script in the `/etc/init.d` directory.

To start or stop a particular SmartConnector, find the control script and run it with either a **start** or **stop** command parameter.

For example:

```
/etc/init.d/arc_serviceName {start|stop}
```

To verify that a SmartConnector service has started, view the file:

```
$ARCSIGHT_HOME/logs/agent.out.wrapper.log
```

To reconfigure SmartConnectors as a daemon, run the SmartConnector Configuration Wizard again. Open a command window on `$ARCSIGHT_HOME/current/bin` and enter:

```
runagentsetup
```

For more information about modifying the connector settings, see Modifying Connector Settings.

Managing SmartConnectors with ArcSight Management Center

ArcSight Management Center (ArcMC) serves as a centralized management interface to help you effectively administer and monitor Transformation Hub and the SmartConnectors. It also centralizes connector management and offers unified control of connectors on local and remote Management Centers as well as software-based connectors installed on remote hosts.

For more information, refer to the Managing Connectors section in the *ArcSight Management Center Help*.

When a connector is managed by ArcMC, **runagentsetup** can no longer be used to manage that connector.

Benefits of Using ArcMC to Manage SmartConnector

ArcSight Management Center features a web-based user interface to enable the management of local or remote connectors. ArcSight Management Center connectors are grouped in *containers*. Each container is a Java Virtual Machine (JVM) that can contain multiple connectors.

Management Center includes on-board connectors that connect event sources to destinations such as Logger and ESM. ArcSight Management Center is useful when connectors target multiple heterogeneous destinations (for example, when Logger is deployed along with ESM), in a Logger-only environment, or when many connectors are involved, such as in a MSSP deployment.

The Management Center delivers the following features and benefits:

- Manage these local connectors as well as remote connectors.

Note: Busy on-board connectors might impact the performance of the ArcSight Management Center web-based interface.

- Manage connectors on remote ArcSight Management Centers, as well as other ArcSight hardware solutions such as Logger.
- Remotely manage previously-installed, software-based connectors
- Supports bulk operations across all connectors and is particularly useful in ESM deployments with many connectors, such as a Managed Security Services Provider (MSSP).
- Provides an ESM-like connector management facility in Logger-only environments.

- Centralized troubleshooting of specific connectors.
- Provides a single interface through which to configure, monitor, tune, and update connectors. The Management Center does not receive events from the connector it manages, and this allows for management of many connectors at one time. The Management Center does not affect working connectors unless it is used to change their configuration. In some cases, the connector is commanded to restart.

ArcSight produced two solutions for the central management of multiple connectors: Connector Appliance and ArcSight Management Center. Connector Appliance is an ArcSight legacy product that enabled central management and monitoring of multiple connectors. Its successor, ArcSight Management Center (ArcSight Management Center) includes all the Connector Appliance management functionality, but its capabilities also include management and monitoring of an additional range of ArcSight products, such as Loggers and other ArcSight Management Centers. For more information about ArcSight Management Center, see the *ArcSight Management Center Administrator's Guide*.

Connectors that forward events to ESM can be managed using the Console, so ArcSight Management Center is not required if all connectors have ESM as their only destination.

Ports and protocols used by SmartConnectors for remote management

The following table describes the most commonly used ports and protocols by SmartConnectors for remote management:

Source Device	Destination Device	Destination Ports using TCP	Additional Protocols	Notes
SmartConnector Remote Management	SmartConnector	TCP 9001 (default, but configurable)	TLS	CWSAPI is an internal API that exposes an API via SOAP. It is used by ArcMC for remotely managing connectors. By default, ArcMC appliance uses ports 9001-9008. This port number must be different for multiple instances of connector in one machine.
Load Balancer - Primary/Secondary Node	SmartConnector	9001	N/A	remote.management.listener.port from agent.properties.

Load Balancer Remote Management	Load Balancer - Secondary Node	TCP 9090 (default, but configurable)	N/A	'vipPingPort' is internally used to check if VIP address is still bound to one of the member hosts for continuous event collection.
SmartConnector	Transformation Hub	9092, 9093	TLS	Ports 9092 must be reachable by all Transformation Hub nodes, consumers, and producers. If you are using TLS, port 9093 must also be reachable. Producers are SmartConnectors.
SmartConnector	Logger	9000	N/A	The SmartMessage receiver listens on 9000/tcp on Software Logger installed as non-root. The Software Logger ports might vary.
SmartConnector	Azure Event Hub	9093	Kafka Wire	SmartConnector communication with Azure Event Hub.
SmartConnector	ArcSight SaaS	9196	TLS	SmartConnector communication with MSK Broker.

Remotely managing software-based Connectors

Previously-installed, software-based connectors can be remotely managed by some ArcSight Management Center models, but the remote management feature is disabled on software connectors by default.

Note: You do not need to do the following processes for ESM or Express. These processes are only done for SmartConnectors running as a service, not for standalone SmartConnectors because they cannot be restarted automatically.

To manage software-based connectors with ArcSight Management Center, you need to enable remote management on them. Add the following property to the `user/agent/agent.properties` file in the installation directory of each connector that you want to manage with ArcSight Management Center:

```
remote.management.enabled=true
```

Restart the connectors for property changes to take effect.

You can also customize the port on which the connector will be listening. By default, this port is set to 9001, but it can be changed by adding the following property to `user/agent/agent.properties`:

```
remote.management.listener.port=9002
```

In the example above, the connector listens on port 9002.

Caution: Only fifth-generation connectors support remote management, so you will need connector build 4855 (4.0.5.4878.0) or later to use this feature. Remote Management is not supported on connectors running AIX. This limitation is due to elements within the AIX platform.

Tip: Multiple software-based connectors installed on the same host require a separate port assignment. The default port for connectors is **9001**, so the second connector installed on the same host must use an alternate port. OpenText recommends using port **9002, 9003, 9004**, and so on.

For a complete list of all connectors supported by ArcSight Management Center, see the ArcSight Management Center Release Notes. You can also visit the Community site at https://community.softwaregrp.com/t5/ArcSight/ct-p/arcsight. ArcSight adds new connectors regularly.

Login Credentials for Software-Based Connector Remote Management

Login credentials are required for software-based connector remote management. Each connector ships with default credentials, which are provided below. The username cannot be changed. To change the default password, administrators can refer to "Changing Container Credentials" in the *ArcSight Management Center Administrator's Guide*.

Note: Load Balancer only works with connectors that use default remote management user name and password values.

Verify with your administrator what are the correct credentials for your environment.

The default connector remote management credentials are:

- Username: `connector_user`
- Password: `change_me`

Limiting Connector Access for Specific IP Address

The new property, `remote.management.listener.client.ip.allow`, allows precise control over the access of the connector for specific IPv4 addresses. This property accepts only IPv4 addresses and does not support hostnames.

Specify the `remote.management.listener.client.ip.allow` property in the `agent.properties` file with the required IPv4 address or addresses. The addresses must be separated using the pipe symbol **|**. This property is useful when there is a need to restrict connector access to designated instances of ArcMc.

For example, `remote.management.listener.client.ip.allow = 10.10.10.10|20.20.20.20`

If your ArcMc instances are running on the following IP addresses `10.10.10.10`, `20.20.20.20`, and `30.30.30.30`, and you have specified `remote.management.listener.client.ip.allow = 10.10.10.10|20.20.20.20` in the `agent.properties` file, the connector can be added only to the ArcMc instances running on `10.10.10.10` and `20.20.20.20`. You cannot add the connector to the ArcMc instance running on `30.30.30.30` because of the restrictions imposed by the property.

Grouping of Connectors

The Connector logical grouping feature enables you to logically group the Connectors so that you can track the licensed EPS counts per group. You can enable this feature only from ArcMC.

Note: By default, the `agent[0].connector.group.name` property is empty in the **agent.properties** file.

To group Connectors:

1. For the standalone installation, open **agent.properties** and enter a valid value for the following property:
   `agents[0].connector.group.name=<group name>`
2. Update the `agents[0].connector.group.name` agent property in ArcMC.

   For more information about updating container properties (located in the **agent.properties** file), refer to the *ArcSight Management Center Administrator's Guide* available at the ArcSight Management Center (ArcMC) page.

Using a customer-supplied certificate for remote management

In the default configuration the connector uses the self-signed certificate for remote management. You can provide your own certificate and keystoreto replace those produced by the connector. You must copy the signed certificate and private key to the machine where Connector will run and create a keystore.

The following procedure is an example. You might have alternative procedures (signed by public CA) for creating the private key and certificate in your environment.

1. Open a **Command prompt/shell** window on the machine where the Connector is installed, and then navigate to the **user/agent** directory of the connector installation. Display the current `remote_management.p12 keystore` to obtain the **Alias name**. You must use alias

name in subsequent steps.

2. To display the current `remote_management.p12 keystore` to obtain the **Alias name**, enter the following command:

```
$ARCSIGHT_HOME/jre/bin/keytool -list -v -keystore remote_management.p12 -storetype
PKCS12 -storepass changeit
```

Sample output of this command is as follows:

```
Keystore type: PKCS12 Keystore provider: SunJSSE
```

```
Your keystore contains 1 entry
```

```
Alias name: cn=n15-214-157-
h159.my.company.com,ou=jjieufkbabcaarn85auxw,o=arcsight,l=n a,st=na,c=us
```

```
Creation date: Jan 26, 2017
```

```
Entry type: PrivateKeyEntry
```

```
Certificate chain length: 1
```

```
Certificate[1]:
```

Important: Ensure that you make a note of the Alias name displayed in the output.

3. Rename the `remote_management.p12 keystore to remote_management.p12- self-signed`.

The `remote_management.p12 keystore` will be replaced. This creates a backup of the original.

4. Generate a private key to be used by Connector, for example: `openssl genrsa -out Server_key.pem 2048`
5. Generate a certificate signing request for the connector certificate, for example: `openssl req -new -key server_key.pem -out server.csr`
6. Present the certificate signing request to a certificate authority and obtain a signed connector certificate.
7. Create a pkcs12 keystore on the machine where the connector will run. Use the alias name obtained in step 2 for the **-name** parameter. The keystore name is **remote_ management.p12**.

```
openssl pkcs12 -export -clcerts -in Server.crt -inkey Server_key.pem -out remote_
management.p12 -name "cn=n15- 214-157-
h159.my.company.com,ou=jjieufkbabcaarn85auxw,o=arcsight,l=n a,st=na,c=us" -password
pass:changeit
```

8. Verify `remote_management.p12 keystore`.

The keystore must be displayed without error and the alias name must be the same as obtained in step 2.

```
$ARCSIGHT_HOME/jre/bin/keytool -list -v -keystore remote_management.p12 -storetype
PKCS12 -storepass changeit
```

9. Verify that the certificate for the certificate authority that signed the certificate is present in the Java keystore used by the connector. The following command will display the keystore contents:

```
$ARCSIGHT_HOME/jre/bin/keytool -list -storepass changeit - keystore $ARCSIGHT_
HOME/jre/lib/security/cacerts
```

10. Import the certificate for the certificate authority, if it is already not there in the keystore.

```
$ARCSIGHT_HOME/jre/bin/keytool -importcert -file <ca_certificate file_name> -
storepass changeit -keystore $ARCSIGHT_HOME/jre/lib/security/cacerts
```

11. Delete the self-signed remote management certificate from both the Java keystore and the FIPS keystore. Use the alias obtained in step 2.
    - To delete the self-signed remote management certificate from the Java keystore, enter the following command:

      ```
      $ARCSIGHT_HOME/jre/bin/keytool -delete -alias "cn=n15-214- 157-
      h159.my.company.com,ou=jjieufkbabcaarn85auxw,o=arcsight,l=n a,st=na,c=us" -
      keystore $ARCSIGHT_HOME/jre/lib/security/cacerts -storepass changeit
      ```

    - To delete the self-signed remote management certificate from the FIPS keystore, enter the following command::

      ```
      jre/bin/keytool -delete -alias "cn=n15-214-157-
      h159.my.company.com,ou=jjieufkbabcaarn85auxw,o=myCompany,l= na,st=na,c=us" -
      keystore $ARCSIGHT_HOME/user/agent/fips/bcfips_ks -storepass changeit -
      storetype BCFKS -providername BCFIPS -providerclass
      org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider - providerpath
      $ARCSIGHT_HOME/lib/agent/fips/bc-fips- 1.0.0.jar -J-
      Djava.security.egd=file:/dev/urandom
      ```

12. Restart the connector so that it will begin using the new keystore and certificate. The ArcMC need to update the connector's certificate in the ArcMC console.
13. To verify the certificate used by the connector management service:

```
openssl s_client -connect <connector>:9001
```

Managing SmartConnector destinations

You can create additional destinations, remove destinations and failover destinations. You can also re-register destinations.

Configuring additional destinations

You can configure additional destinations so that a copy of events can be sent to each configured additional destinations. Additional destinations can be useful, for example, when you have a development ArcSight environment working in parallel with your production environment and you want to test rules and reports. In such cases, you can configure the connector to send alerts to both your production Manager and your development Manager to be able to view real-time event flows on both systems. Because the destinations are independent, you do not compromise the events sent to the production Manager.

To add a destination:

1. Run the installation wizard, select **Modify Connector** , then click **Next**.
2. Select **Add**, **modify**, **or remove destinations**.
3. Select **Add destination** to add another destination.
4. Click **Next**; the window for adding, modifying, or removing destinations will be displayed.
5. Specify the relevant details to add a destination.

Adding a failover destination

Note: The **Adding a failover destination** option is not applicable for the **ArcSight SaaS** destination

Each connector destination can have a failover destination that receives security events from the connector for which it is configured. The failover activates when the primary destination is not available because of issues such as network problems or is not keeping up with incoming events. It acts as a real-time alternative for severe problems with the primary destination.

If a failover destination is configured, then the events which could not be sent to primary destination are backed up to be sent to the failover destination. The connector also, when possible, caches the events and resends them to the primary destination when the flow is restored.

A failover destination is not active when the primary destination is available, so the reports and replay features within the secondary Manager could contain incomplete information.

To add a failover destination:

1. Run the installation wizard, select **Modify Connector** , then click **Next**.
2. Select **Add**, **modify**, **or remove destinations**, then click **Next**.
3. Select the destination that you want to modify, then click **Next**
4. Select **Add a failover destination**.
5. Proceed with the rest of the screens in the wizard to add the failover destination.
6. Restart the Connector to apply your changes.

Re-registering a destination

When the Manager recognizes a connector, it generates an ID token to identify its security events. If the Manager stops accepting events from a connector for an unknown reason, or if you have upgraded a connector but its resource was removed from the database, then you must re-register the connector.

To re-register destination:

1. After running the wizard, **Modify Connector** is selected by default. Do not change this selection. Click **Next**.
2. Select **Add, modify, or remove destinations** and click **Next**.
3. Select the destination to re-register and click **Next**.
4. Select **Reregister destination** and click **Next**.
5. Specify the required credentials, if prompted.
6. After the reregistration completes, restart the connector to apply the new ID token.

Removing a destination

1. Run the installation wizard, select **Modify Connector**, then click **Next**.
2. Select **Add**, **modify**, **or remove destinations**.
3. Select **Add destination** to add another destination and click **Next**.
   The window for adding, modifying, or removing destinations will be displayed.
4. Specify the relevant details to add a destination and click **Next**.

5. From the list of destination selections, select the destination to remove, and then click **Next**.
6. Select **Remove destination** to start the destination removal process and click **Next**.
7. Complete the destination removal process and click **Next**.
8. Choose **Exit** to complete the connector modification, or choose **Continue** to continue to make connector modifications.

Configuring destination settings

Configuring Batching

SmartConnectors can create batch events to increase performance and optimize network bandwidth. When activated, SmartConnectors create blocks of events and send them when they either reach a certain size or the time window expires. You can also prioritize batches by severity, forcing the SmartConnector to send the highest-severity event batches first and the lowest-severity event batches later.

To configure Batching:

1. Run the `runagentsetup.bat` file.
2. Select **Modify Connector**, then click **Next**.
3. Select **Add, Modify, or Remove Destinations**, then click **Next**.
4. Select the destination for which you want to configure batching, then click **Next**.
5. Select **Modify destination settings**, then click **Next**.
6. Select **Batching**, then click **Next**.
7. Specify the following information, then click **Next**.

   Configurable Settings

Name Field	Value Field
Enable Batching (per event)	The SmartConnector creates batches with the specified number of events. (100,200, 300, 400, 500, or 600 events). Default is **100**. ⚠ You could potentially lose data with batch sizes 500 and 600. Contact Customer Support before using 500 or 600 batch size.
Enable Batching (in seconds)	The SmartConnector sends the events if the specified timer expires (1, 5, 10, 15, 30, 60). Default is **5**.
Batch By	Select **Time Based** if you want the SmartConnector to send batches as they arrive. This is the default value. Select **Severity Based** if you want the SmartConnector to send batches based on severity (batches with highest severity events are sent first).

8. Select **Done with editing destination settings**, then click **Next**.
9. Click **Exit**.

Configuring Time Correction

You can configure time correction to fix problems with devices that do not report the time correctly.

To configure Time Correction:

1. Run the `runagentsetup.bat` file.
2. Select **Modify Connector**, then click **Next**.
3. Select **Add, Modify, or Remove Destinations**, then click **Next**.
4. Select the destination for which you want to configure batching, then click **Next**.
5. Select **Modify destination settings**, then click **Next**.
6. Select **Time Correction**, then click **Next**.
7. Specify the following information, then click **Next**.

   Configurable Settings

Name Field	Value Field
Use Connector Time as Device Time	(**No** | Yes) Overrides the time reported by the device with the time the connector received the event. This option assumes that the connector is more likely to report the correct time. Default is **No**.
Enable Device Time Correction (in seconds)	The SmartConnector adjusts the time reported by the connector, using this setting. This is useful when a remote device's clock isn't synchronized with the ArcSight Manager. This should be a temporary setting. The recommended way to synchronize clocks between Manager and devices is the NTP protocol. This parameter also affects the `startTime` and `endTime` fields. Default is **0**.
Enable Connector Time Correction (in seconds)	The SmartConnector can also adjust the time reported by the Connector Time SmartConnector itself, using this setting. This is for informational purposes only and lets you to modify the local time on the SmartConnector. This should be a temporary setting. The recommended way to synchronize clocks between Manager and SmartConnectors is the NTP protocol. Default is **0**.
Set Device Time Zone To	(**Disabled** | <TimeZone>) It is assumed that the original device or the SmartConnector reports the time along with the time zone. If not, you can select the required time zone from the list. The selected time zone is applied to the reported time. Default is **Disabled**.

8. Select **Done with editing destination settings**, then click **Next**.
9. Click **Exit**.

Configuring Device Time Auto-Correction

You can configure the time spans to apply auto-correction for the device-time.

To configure Device Time Auto-Correction:

1. Run the `runagentsetup.bat` file.
2. Select **Modify Connector**, then click **Next**.
3. Select **Add, Modify, or Remove Destinations**, then click **Next**.
4. Select the destination for which you want to configure batching, then click **Next**.
5. Select **Modify destination settings**, then click **Next**.
6. Select **Device Time Auto-Correction**, then click **Next**.
7. Specify the following information, then click **Next**.

   Configurable settings

Name Field	Value Field
Future Threshold	The connector auto-corrects the device time, if it is greater than the connector time by the value specified for **Future Threshold seconds** field. If either or both the future and past thresholds are negative, auto-correction is disabled. Default is **-1**.
Past Threshold	The connector auto-corrects the device time, if it is greater than the connector time by the value specified for **Past Threshold seconds** field. Default is **-1**.
Device List	A comma-separated list of the devices to which the thresholds apply. The default, (ALL) means all devices.

8. Select **Done with editing destination settings**, then click **Next**.
9. Click **Exit**.

Configuring Time Checking

To configure Time Checking:

1. Run the `runagentsetup.bat` file.
2. Select **Modify Connector**, then click **Next**.
3. Select **Add, Modify, or Remove Destinations**, then click **Next**.
4. Select the destination for which you want to configure batching, then click **Next**.
5. Select **Modify destination settings**, then click **Next**.
6. Select **Time Checking**, then click **Next**.
7. Specify the following information, then click **Next**.

   Configurable Settings

Name Field	Value Field
Future Threshold	The number of seconds by which to extend the connector's forward threshold for time checking. Default is **5 minutes** (300 seconds).
Past Threshold	The number of seconds by which to extend the connector's rear threshold for time checking. Default is **1 hour** (3600 seconds).
Frequency	The SmartConnector checks its future and past thresholds at intervals specified by this number of seconds. Default is **1 minute** (60 seconds).

8. Select **Done with editing destination settings**, then click **Next**.
9. Click **Exit**.

Configuring Caching

SmartConnectors use a compressed disk cache to hold large volumes of events when the ArcSight Manager is down or when the SmartConnector receives bursts of events.

Changing these settings does not affect the events cached, it only affects new events sent to the cache.

To configure Caching:

1. Run the `runagentsetup.bat` file.
2. Select **Modify Connector**, then click **Next**.
3. Select **Add, Modify, or Remove Destinations**, then click **Next**.
4. Select the destination for which you want to configure batching, then click **Next**.
5. Select **Modify destination settings**, then click **Next**.
6. Select **Caching**, then click **Next**.
7. Specify the following information, then click **Next**.

   Configurable Settings

Name Field	Value Field
Cache Size	This parameter specifies the disk space for caching events. The default is **1 GB** depending on the connector, the cache can hold about 15 million events, but it also can go down to 200 MB. When this disk space is full, the SmartConnector drops the oldest events to free up disk cache space. Select the option available in the drop-down list.
Notification Threshold	The number of events in the cache that triggers a notification. Default is **10,000** events.
Notification Frequency	Indicates how often a notification must be sent when the notification threshold is reached. Select the option available in the drop-down list. Default is **10 min**.
Maximum File Count	The value set in the user properties represents the maximum number of cache files that guarantees no events dropping, and not the actual amount of cache files created for ingestion. Cache enters in Event drop mode after the number of cache files reache the limit set. Alternatively, when the number of cache files reaches double the amount set, caching enters the File drop mode.

8. Select **Done with editing destination settings**, then click **Next**.
9. Click **Exit**.

Configuring Network

To configure Network:

1. Run the `runagentsetup.bat` file.
2. Select **Modify Connector**, then click **Next**.
3. Select **Add, Modify, or Remove Destinations**, then click **Next**.

4. Select the destination for which you want to configure batching, then click **Next**.
5. Select **Modify destination settings**, then click **Next**.
6. Select **Network**, then click **Next**.
7. Specify the following information, then click **Next**.

Configurable Settings

Name Field	Value Field
Heartbeat Frequency	This setting controls how often the connector sends a heartbeat message to the ArcSight Manager. The default is **5 seconds**, but it can vary from **5 seconds** to **10 minutes**. Note that the heartbeat is also used to communicate with the SmartConnector; therefore, if its frequency is set to **10 minutes**, then it could take as much as 10 minutes to send any configuration information or commands back to the SmartConnector. Select from the options available in the drop-down list. Default is **10** seconds.
Enable Name Resolution	(No | Source/Dest only | **Yes**) The SmartConnector tries to resolve IP addresses to host names, and host names to IP addresses, if the event rate allows it and if required. This setting controls this functionality. The Source, Target and Device IP addresses and Host names may also be affected by this setting. The **Source/Dest Only** choice means that the **Device Address** and **Device Host Name** fields are ignored for name resolution. Default is **Yes**. **Note:** When setting this value to **NO**, update Name Resolution TTL (secs) to **-1**
IPv6 Name Resolution Control	• **IPv4 Only**: for Legacy Events . This is the default option. • **IPv6 (Prefer IPv4 for reverse resolution)** : for Legacy Events . • **IPv6 (Prefer IPv6 for reverse resolution)**: for Legacy Events.
Name Resolution TTL (Secs)	This is the amount of time (Time to Live) the name resolution is to be in effect. The name resolution entries are cached for this time . Default is **3600**.

Configurable Settings, continued

Name Field	Value Field
Wait for Name Resolution	(Yes | **No**) If set to **Yes**, the SmartConnector waits for name resolution to be completed. When **Yes** is selected, event processing might be slowed down significantly and even cause lost events. Default is **No**.
Name Resolution Host Name Options	• Set host name only (lowercase) • Set host and domain names • Set host and domain names (lowercase) For reverse resolution (IP Address to Host name), only the host name field is set. If host name only is not used, the host name is split up and put into both the DNS domain and the host name fields. This affects the source, destination, device and agent address. If one of the (lowercase) choices is made, then the name is changed to lowercase before it is put into the host name (and possibly DNS domain) field(s). Default is **Set host name only**.
Name Resolution Domain from E-mail	(**Yes** | No) If set to **Yes**, the host name and DNS domain fields are empty, and the corresponding user name field appears as an e-mail address, then the domain from the e-mail address is put in the DNS domain field. This only affects the source and destination fields. Default is **Yes**.
Clear Host Names Same as IP Addresses	(**Yes** | No) If set to **Yes** and the host name field is set to an IP Address that matches the corresponding IP Address field, then the host name field is cleared. This affects the source, destination, and device fields. Default is **Yes**.
Set Host Names to IP Addresses when Unknown	(Yes | **No**) If set to **Yes**, host names that remain unresolved are set to IP addresses. Default is **No**.
Don't Resolve Host Names Matching	By default, host names are resolved to their IP addresses. You have the option to specify a regular expression for all or part of a host name for which you do not want the system to attempt host name resolution to an IP address. When this option is configured, the system cannot resolve host names matching this expression.

Configurable Settings, continued

Name Field	Value Field
Don't Reverse-Resolve IP Ranges	By default, IP addresses are resolved to their domain names. You have the option to specify IP address ranges for which you do not want the system to attempt reverse-resolution to domain names. Click in the field to enter the IP address range. To enter a single IP address, enter the address under the From column and leave the `To` column blank, then click **Apply**. For an address range, enter the starting IP address under `From` and the ending address under `To`, then click **Apply**. This field allows you to enter a list of ranges. When this option is configured, the system cannot reverse-resolve IP addresses that fall within any of the specified ranges.
Remove Unresolvable Names/IPs from Cache	(Yes | Yes (w/ negative cache) | **No**) • If set to **No**, unresolvable host names or IP addresses continue to be in the cache. • If set to **Yes**, unresolvable host names or IP addresses are removed from the cache. • If set to **Yes (w/negative cache)**, the connector remembers what names/IPs have been unresolvable so that time is not wasted trying to resolve them frequently. Default is **No**.
Limit Bandwidth To	Select from a list of bandwidth options you can use to constrain the connector's output over the network. Default is **Disabled**.
Transport Mode	(**Normal** | Cache | Cache but send Very High severity events). You can configure the SmartConnector to cache to disk all the processed events it receives. This is equivalent to pausing the SmartConnector. However, you can use this setting to delay event-sending during particular periods. For example, you could use this setting to cache events during the day and send them at night. You can also set the connector to cache all events, except for those marked with a high severity, during business hours, and send the rest at night. Default is **Normal**.

Configurable Settings, continued

Name Field	Value Field
Cache Mode	(**Normal** | Drop if Dest Down) This option is meant to be used on a primary destination to control the caching behavior of the primary destination when it is down, and the connector starts sending events to the failover destination. In the Normal mode, events are cached and sent to the primary destination when it comes back up. In the **Drop if Dest Down** mode, the events are not cached and dropped and therefore not sent to the primary destination when it becomes available again. Default is **Normal**.
Address Based Zone Population Defaults Enabled	(**Yes** | No) If Yes, the default zones built into the connector will be used to assign zones. These zones are only used if a network model has not been sent by ESM or ArcMC, or if that network model does not cover some addresses. If the Address-Based Zone Population setting (below) is specified, you may want to change this to **No**. Default is **Yes**.
Address Based Zone Population	If specified in setup or ArcMC, this is a comma-separated list that must contain a multiple of three items. • The first of each three is the starting IP address of a zone. • The second is the ending IP address of the zone. • The third is the URI of the zone to assign to addresses in that range. These zones are only used if a network model has not been sent by ESM or ArcMC, or if that network model does not cover some addresses. If **Address-Based Zone Population Defaults Enabled** is set to Yes, the zones specified here take precedence over those. For example, for two zones, this can be: 15.0.0.0,15.255.255.255,/All Zones/ArcSight System/Public Address Space Zones/Hewlett-Packard Company,17.0.0.0,17.255.255.255,/All Zones/ArcSight System/Public Address Space Zones/Apple Computer Inc.
Zone Population Mode	(**Normal** | Rezone (override) | No Zoning (clear)) If set to **Normal** means zones are computed and assigned, if not already set. **Rezone (override)** re-computes and re-assigns already populated zones. **No Zoning (clear)** clears the zones, if already populated. Default is **Normal**.

Configurable Settings, continued

Name Field	Value Field
Customer URI	Applies the given customer URI to events emanating from the connector. Provided the customer resource exists, all customer fields are populated on the ArcSight Manager. If this connector is reporting data that might apply to more than one customer, you can use Velocity templates in this field to conditionally identify those customers.
Source Zone URI	When populated, this field shows the URI of the zone associated with the SmartConnector's source address. How this field gets populated is discussed in the Zones section of the SmartConnectors topic. This field is available for ESM v3.0 compatibility. It is not relevant in post ESM 3.0 releases because of integral zone mapping.
Source Translated Zone URI	When populated, this field shows the URI of the zone associated with the SmartConnector's translated source address. The translation is presumed to be NAT (network address translation). How this field gets populated is discussed in the Zones section of the SmartConnectors topic. This field is present for ESM v3.0 compatibility. It is not relevant in post ESM 3.0 releases because of integral zone mapping.
Destination Zone URI	When populated, this field shows the URI of the zone associated with the SmartConnector's destination address. How this field gets populated is discussed in the Zones section of the SmartConnectors topic. This field is present for ESM v3.0 compatibility. It is not relevant in post ESM 3.0 releases because of integral zone mapping.
Agent Zone URI	When populated, this field shows the URI of the zone associated with the SmartConnector's translated destination address. The translation is presumed to be NAT (network address translation). How this field gets populated is discussed in the Zones section of the SmartConnectors topic. This field is present for ESM v3.0 compatibility. It is not relevant in post ESM 3.0 releases because of integral zone mapping.

Configurable Settings, continued

Name Field	Value Field
Agent Translated Zone URI	When populated, this field shows the URI of the zone associated with the SmartConnector's translated address. The translation is presumed to be NAT (network address translation). How this field gets populated is discussed in the Zones section of the SmartConnectors topic. This field is present for ESM v3.0 compatibility. It is not relevant in post ESM 3.0 releases because of integral zone mapping.
Device Zone URI	When populated, this field shows the URI of the zone associated with the device's address. How this field gets populated is discussed in the Zones section of the SmartConnectors topic. This field is present for ESM v3.0 compatibility. It is not relevant in post ESM 3.0 releases because of integral zone mapping.
Device Translated Zone URI	When populated, this field shows the URI of the zone associated with the device's translated address. The translation is presumed to be NAT (network address translation). How this field gets populated is discussed in the Zones section of the SmartConnectors topic. This field is present for ESM v3.0 compatibility. It is not relevant in post ESM 3.0 releases because of integral zone mapping.

8. Select **Done with editing destination settings**, then click **Next**.
9. Click **Exit**.

Configuring Connector Networks and Zones

Network Model: Networks and Zones

A network model is a set of networks with its specific zones. The zones, unlike the networks, do not overlap. Because destinations can have different network models, a connector can set specific zones, while sending events to multiple destinations. Consequently, the network model data for each destination must be kept separately as well as the components in the destination-specific event flow.

The network model is supported from:

- ESM 3.5 or later. It is used for ESM destinations and non-ESM destinations if there is an ESM and the AUP Master feature is in use.
- An ArcMC-pushed Network Model.

To configure ESM network models, see OpenText Security ArcSight ESM Administrator´s Guide.

The following `networks.csv` and `zones.csv` files are applicable for ArcMC-pushed Network Model:

Networks CSV:

The `networks.csv file` defines the networks used in the `zones.csv file`.

> **Tip:** While creating zones, enter the exact URI for each network. Any zones connected to unknown networks (or most likely to networks incorrectly specified) cannot be used.

To Add the Network:

The code accepts details in the following without any extra spaces but is not case-sensitive:

```
#Type,Name,Parent Group URI,Customer URI
```

The CSV file must contain the following details:

First column: Name the type as `"Global"` or `"Network"`. The type, `Global` must appear at least once. The type `"Network"`is used to define other networks.

Second column: Add a name for each network.

Third column: Name the URI of the parent group in the network URI hierarchy.

Fourth column: (Optional) This column must be blank for lines with Global in the first column, and can be blank for lines with Network in the first column. If the field is blank, add a comma after specifying the parent group URI, so that all the rows display four columns. If the field is not blank, the value is displayed in the customer's URI field. Any events that do not have this specific value in the customer's URI field are not considered by the network (or the zones within it).

The order of the networks depends on how the non- global (user) networks are displayed.

Zones CSV:

The `zones.csv file` defines the zones within the networks already populated in the `networks.csv file`.

To Add Zones:

Add the header with no extra space in the following format:

`#Name,Start Address,End Address,Parent Group URI,Network URI`

The code only tolerates a difference in upper and lower cases.

The file is divided into 5 columns and each row defines a zone.

- **First column:** Define the name of the zone.
- **Second column:** Name the starting IP address in the range.
- **Third column:** Name the ending IP address in the range. It can either be IPv4 or IPv6. However, the type must match with the starting IP address in the previous column.
- **Fourth column:** Label the URI of the parent group in the zone URI hierarchy. This is the first part of the zone URI.
- **Fifth column:** Add the network URI to define its corresponding zone. It must be an exact match, the concatenated URI from a network defined in the `networks.csv file` (the third column and second column, concatenated, in one row of the file).

Note: Events with addresses that fall outside the zones defined by the network model cannot be changed, so defining a network model does not necessarily affect all events.

To discard incoming zones and apply new ones, if applicable, go to **Destination Settings** > **Network Group** > **Population Mode** and set **SmartConnector** to **Rezone (override)** instead of **Normal**. You can also set this from ArcMC.

Debugging tips:

- From your destination, search for **AddrBasedSysZonePopRows** and **AddrBasedUsrZonePopRows** in **get status results** or check the logs in **logStatus**. To look for errors and consider the time range in which the network model was pushed, or when the connector was started.
- For additional information on ArcMC network models, see the Managing Configurations section of ArcSight Management Center Help.

Configuring Field-Based Aggregation

Field-based aggregation implements a flexible aggregation mechanism; two events are aggregated if only the *selected* fields are the same for both events.

Note: Field-based aggregation creates a new alert that contains only the fields that were specified, so the rest of the fields are ignored, unless **Preserve Common Fields** is set to **Yes**.

SmartConnector aggregation significantly reduces the amount of data received, and should be applied only when you use less than the total amount of information the event offers. For example, you could enable field-based aggregation to aggregate "accepts" and "rejects" in a firewall, but you should use it only if you are interested in the count of these events, instead of all the information provided by the firewall.

To configure Field-Based Aggregation:

1. Run the following file:
   - For Windows machine: `runagentsetup.bat`
   - For Linux machine: `runagentsetup.sh`
2. Select **Modify Connector**, then click **Next**.
3. Select **Add, Modify, or Remove Destinations**, then click **Next**.
4. Select the destination for which you want to configure batching, then click **Next**.
5. Select **Modify destination settings**, then click **Next**.
6. Select **Field Based Aggregation**, then click **Next**.
7. Specify the following information, then click **Next**.

   Configurable Settings

Name Field	Value Field
Time Interval	Choose a time interval, if applicable, to use as a basis for aggregating the events the connector collects. Aggregation time interval and threshold settings must be set to enable aggregation. Default is **Disabled**.
Event Threshold	Select the maximum number of events to use as a basis for aggregating the events the connector collects. This is the maximum count of events that can be aggregated; for example, if 900 events were found to be the same within the time interval selected (for example, contained the same selected fields) and you select an event threshold of 500, you then receive two events, one of count 500 and another of count 400. This option is exclusive of Time Interval. Default is **Disabled**.
Field Name	Choose one or more fields, if applicable to use as the basis for aggregating the events the connector collects. The result is a comma-separated list of fields to monitor. Do not include ESM derived fields (in Italic) in Field Based Aggregation.

Configurable Settings, continued

Name Field	Value Field
Fields to Sum	Choose one or more fields, if applicable, to use as the basis for aggregating the events the connector collects. If specified, this set of numeric fields is summed rather than aggregated, preserved, or discarded. The most common fields to sum are `bytesIn` and `bytesOut`. If any of the fields listed here are also in the list of field names to aggregate, they are aggregated and not summed.
Preserve Common Fields	(Yes | **No**) Choosing **Yes** adds fields to the aggregated event if they have the same values for each event. Choosing **No** ignores non-aggregated fields in aggregated events. Default is **No**.

8. Select **Done with editing destination settings**, then click **Next**.
9. Click **Exit**.

Configuring Filter Aggregation

Filter Aggregation is a way of capturing aggregated event data from events that would otherwise be discarded due to an agent filter. Only events that would be filtered out are considered for filter aggregation (unlike Field-based aggregation, which looks at all events).

To configure Filter Aggregation:

1. Run the `runagentsetup.bat` file.
2. Select **Modify Connector**, then click **Next**.
3. Select **Add, Modify, or Remove Destinations**, then click **Next**.
4. Select the destination for which you want to configure batching, then click **Next**.
5. Select **Modify destination settings**, then click **Next**.
6. Select **Filter Aggregation**, then click **Next**.
7. Specify the following information, then click **Next**.

Configurable Settings

Name Field	Value Field
Time Interval	Choose a time interval, if applicable, to use as a basis for aggregating the events the connector collects. It is exclusive of Event Threshold. Select from the options available in the drop-down list. Default is **Disabled**.
Event Threshold	Choose a number of events, if applicable, to use as a basis for aggregating the events the connector collects. This is the maximum count of events that can be aggregated; for example, if 900 events were found to be the same within the time interval selected (for example, contained the same selected fields) and you select an event threshold of 500, you then receive two events, one of count 100 and another of count 400. This option is exclusive of Time Interval. Select from the options available in the drop-down list. Default is **Disabled**.
Fields to Sum	(Optional) Choose one or more fields, if applicable, to use as the basis for aggregating the events the connector collects.

8. Select **Done with editing destination settings**, then click **Next**.
9. Click **Exit**.

Configuring Processing

To configure processing:

1. Run the `runagentsetup.bat` file.
2. Select **Modify Connector**, then click **Next**.
3. Select **Add, Modify, or Remove Destinations**, then click **Next**.
4. Select the destination for which you want to configure batching, then click **Next**.
5. Select **Modify destination settings**, then click **Next**.
6. Select **Processing**, then click **Next**.
7. Specify the following information, then click **Next**.

Configurable Settings

Name Field	Value Field
Preserve Raw Event	(Yes | **No**) Some devices contain a raw event that can be captured as part of the generated alert. If that is not the case, most connectors can also produce a serialized version of the data stream that was parsed/processed to generate the ArcSight event. This setting allows the connector to preserve this serialized "rawEvent" as a field in the event inspector. This setting is disabled, by default, since using raw data increases the event size and therefore requires more database storage space. Select from the options available in the drop-down list. If you select **Yes**, the serialized representation of the "rawEvent" is sent to the selected destination and preserved in the `rawEvent` field. Default is **No**.  **Note**: When selecting **Aggregate Events**, the **Preserve Raw Event** feature is disabled.

Configurable Settings, continued

Name Field	Value Field
Check Event Integrity Method	This parameter enables you to specify a method that you want to use to verify the integrity of events. It is recommended to configure this parameter for only one destination. Because, for a given set of events, a verification event (also known as an agent:040 event) is generated per destination and transmitted to all the configured destinations. If you configure the parameter for only one destination, then it avoids duplication of verification event on all the destinations and reduces the extra load on the connector. If the **Preserve Raw Event** parameter is selected as **Yes** and a valid event integrity algorithm is selected, the connector will generate additional verification events that contain a crypto signature field. This crypto signature field can be used to verify whether the raw event field of a normal event was tampered with, after the normal event was generated by the connector. The crypto signature field has the following format: `"#seq(alg):digest"` where, • **`seq`** is a persistent event sequence number. • **`alg`** is the event integrity algorithm. • **`digest`** is a hexadecimal message digest. If you select **Connector Validation Tool**, the crypto signature field is populated in normal events and the digest is the result of hashing the **raw event** field of a normal event. If you select **Recon**, the crypto signature is still computed, but as an optimization, it is not stored in the normal event. The verification event provides integrity check for the normal events. The crypto signature field of the verification event is the result of hashing the crypto signature field of the normal events. The crypto signature field of these verification events is always populated. Select one of the following options: • **Disabled**: For backwards compatibility purposes, if the **Preserve Raw Event** parameter is selected as **Yes** and a valid event integrity algorithm is selected, then selecting the **Disabled** option for this parameter will be considered as **Connector Validation Tool**. • **Recon**: Verification events are formatted to be compatible with, and optimized for the event integrity check capability of Recon. It is mandatory to specify a unique Generator ID, set the **Preserve Raw Event** parameter to **Yes**, and have a valid event integrity algorithm selected. In the **Recon** mode, the crypto signature field contains globally-unique event ID (GEID) and has the following format: `"#Geid (alg):digest"`

Configurable Settings, continued

Name Field	Value Field
	• **Connector Validation Tool**: Verification events are formatted to be compatible with the EventIntegrityChecker validation tool provided with the connector. It is mandatory to set the **Preserve Raw Event** parameter to **Yes** and have a valid event integrity algorithm selected. Default is **Disabled**.
Event Integrity Algorithm	(**Disabled** | SHA-256 | SHA-1 | MD5 | SHA-512) Supported algorithms are: SHA-256, SHA-1, MD5, and SHA-512. Default is **Disabled** (that is, no algorithm is applied).
Turbo Mode	(**Complete** | Faster | Fastest) If your configuration, reporting, and analytic usage permits, you can greatly accelerate the transfer of a sensor's event information through SmartConnectors by choosing one of two "turbo" (narrower data bandwidth) modes. Select one of the following options: **Complete**, **Faster**, **Fastest**. For information about these modes, see "Understanding the turbo mode" on page 131. Default turbo mode is **Complete**.
Enable Aggregation (in secs)	**Note:** If you have already used this settings for setting up previous SmartConnectors, you can continue to do so. However, ArcSight recommends that you use the new "Configuring Field-Based Aggregation " on page 182feature as a more flexible option. Here is the description of the legacy “Enable Aggregation” feature, for those who are still using it: When enabled, **Enable Aggregation (in seconds)** aggregates two or more events based on the selected time value. (**Disabled**, 1, 2, 3, 4, 5, 10, 30, 60) The aggregated event shows the event count (how many events were aggregated into the displayed event) and event type. The rest of the fields in the aggregated event take the values of the first event in the set of aggregated events. Default is **Disabled**.
Limit Event Processing Rate	You can moderate the SmartConnector's burden on the CPU by reducing its processing rate. This can also be a means of dealing with the effects of event bursts. The choices range from **Disabled** (no limitation on CPU demand) to **1eps** (pass just one event per second, making the smallest demand on the CPU). Be sure to note that this option's effect varies with the category of SmartConnector in use, as described in the SmartConnector Processing Categories table. Default is **-1** [0 minute(s)] that means set to disabled.

Configurable Settings, continued

Name Field	Value Field
Fields to Obfuscate	Using MD5 hashing, this option lets you to specify a list of fields for obfuscation in a security event. In FIPS mode, SHA-256 is used.
Store Original Time In	(Disabled | Flex Date 1) This parameter lets you to move the original device receipt time to a specified field if altered by the time correction. Default is **Disabled**.
Enable Port-Service Mapping	(**No** | Yes) If set to Yes and one of the two fields destination port and application protocol is set, and the other is not, the one that is set is used to set the other. For example, if the destination port is 22 and application protocol is not set, then the application protocol is set to ssh. Default is No.
Uppercase User Names	(**Disabled** | Enabled (orig to ID) | Enabled(orig to ID or Flex) | Enabled(orig to Add. Data)) If set to any of the *enabled* settings, the two user name fields are automatically changed to uppercase. The original values are saved as follows: • **Enabled (orig to ID)** saves the original values to the **sourceUserID** and **destinationUserID** fields, respectively, overwriting any values that may have been there previously. • **Enabled(orig to ID or Flex)** saves the original values in the same fields if they do not already contain values, or to the `flexString1` (source) and `flexString2` (destination) fields if the ID fields do contain values. • **Enabled(orig to Add. Data)** saves the original values to additional data fields called `OrigSrcUsrName` and `OrigDstUsrName`, respectively. **Note**: The uppercase operation is typically done using the default Locale for the chosen platform. You can set this to a Locale by setting the **connector.uppercase.user.name.locale** property in `agent.properties` to the desired Locale (For example: using "en_US" for U.S. English). Default is **Disabled**.

Configurable Settings, continued

Name Field	Value Field
Enable User Name Splitting	(Yes | **No**) If this is set to yes and the destination user name contains commas in the event, this parameter duplicates that event. Each user name in the list is placed in one of the events. For example, if the destination user name in an event is “User 123, User 456”, then that event is sent twice, with the destination user name set to “User 123” in the first and “User 456” in the second. Default is **No**.
Split File Name into Path and Name	(Yes | **No**) If this is set to *yes* and an event’s file name field is set but its file path field is not, this parameter splits the file name into a path and a name, placing each part into appropriate fields. For example, if the file name field is set to `C:\dir\file.ext` and the file path is not set, then the file path is set to `C:\dir` and the file name to file.ext. The separator character can be either \ or / as the system looks to the SmartConnector to determine its platform. Default is **No**.
Generate Unparsed Events	(Yes | **No**) If set to yes and some incoming event data cannot be parsed (perhaps because a device has been upgraded since the SmartConnector parser was written), then a special event named “Unparsed Event” is generated. The raw event appears in the event message field. If set to **No**, the SmartConnector log files indicate the unparsed events. Default is **No**.

Configurable Settings, continued

Name Field	Value Field
Preserve System Health Events	(Yes | **No**) If set to yes, internal system health events are preserved. SmartConnectors generate system health events that provide information about the systems on which they are installed (for example, disk usage, network memory, JVM memory, percentage of processing of CPU memory usage, and so forth). By default, these events are not retained or passed on to ArcSight destinations and, therefore, not available for viewing. Setting this option to yes makes them available in the Console or any destination like Logger. Default is **No**.
Device Status Monitoring	If set to <NumberOfMilliseconds>, the selected SmartConnector generates internal events periodically 1 minute (60000 milliseconds) or greater with the status of the devices for which the connector is receiving normal events. These events have the name "Connector Device Status." Enabling periodic device status monitoring events helps monitor both the SmartConnector and device uptime. Device status monitoring events include this information, if available: • Event name (Connector Device Status) • Vendor and Product information • Source Address and Host Name • Zone • Last event received • Total number of events for the device since the connector started • Event count since last call Device status monitoring events can be set to generate events for every 1 minute (60000 milliseconds), or less frequently (that is, a greater number of milliseconds than the minimum). If you specify less than 60000, you get a warning in the log that the minimum is 60000 milliseconds (1 minute) and the system uses the minimum. If you enter a non-number in the field, it generates an error in the log that the value could not be parsed. In this case, the feature is disabled (and logged as such). In such cases, there is no indication on the Console that anything went wrong because there is no way for the Connector to convey that error. Default is **-1** [0 minute(s)] that means set to disabled.

8. Select **Done with editing destination settings**, then click **Next**.
9. Click **Exit**.

Configuring Payload Sampling

Some SmartConnectors use Payload Sampling to send a portion of the packet payload (as opposed to the complete payload) along with the original event. This portion is retrieved using the on-demand payload retrieval in the event inspector.

Overview of Payload Sampling

Many customers use ArcSight for security event analysis, including investigating the packet records data that triggered the security event. In ArcSight terms, these packet records are called *payload*. Payload refers to the information carried in the body of an event's network packet, as distinct from the packet's header data. While security event detection and analysis usually centers on header data, packet payload may also be forensically significant.

ArcSight supports the following ways to retrieve payload:

- **Payload Sampling** allows up to 1023 bytes of the payload to be retrieved and displayed as ASCII characters in a custom string field for ***each*** event. An option is also provided to display up to 511 bytes in hexadecimal format. By default, the payload sampling feature is not enabled due to its potentially large storage requirements. To enable payload sampling, select **true** for the Enable payload sampling parameter during connector installation.
- **On-Demand Payload Retrieval** lets you retrieve the entire payload if the payload is still held on the device.

You can retrieve, preserve, view, or discard payloads using the ArcSight Console. Because event payloads are relatively large, ArcSight does not store them by default. Instead, you can request payloads from devices for selected events through the Console. If the payload is still held on the device, the ArcSight SmartConnector retrieves it and sends it to the Console.

Payloads are downloaded and stored only on demand. You must configure ESM to log these packets. By default, 256 bytes of payload will be retrieved.

Whether an event has a payload to store is visible in event grids. Unless you specifically request to do so, only the event's "payload ID" (information required to retrieve the payload from the event source) is stored. Payload retention periods are controlled by the configuration of each source device.

Locate Payload-Bearing Events

The first step in handling event payloads is to be able to locate payload-bearing events among the general flow of events in a grid view. In an ArcSight Console Viewer panel grid view, right-click a column header and choose **Add Column > Device > Payload ID**. Look for events showing a Payload ID in that column.

Retrieve Payloads

In a Viewer panel grid view, double-click an event with an associated payload. In the Event Inspector, click the **Payload** tab, then click **Retrieve Payload**.

Preserve Payloads

In a grid view, right-click an event with an associated payload, select **Payload**, then **Preserve**. Alternatively, in the Event Inspector, click the **Payload** tab, then **Preserve Payload**.

Discard Payloads

In a grid view, right-click an event with an associated payload and select **Payload**, then **Discard Preserved**. You also can use the Event Inspector: In a grid view, double-click an event with an associated payload. In the Event Inspector, click the **Payload** tab. Click **Discard Preserved Payload**.

Save Payloads to Files

In a grid view, double-click an event with an associated payload. In the Event Inspector, click the **Payload** tab. Click **Save Payload**. In the **Save** dialog box, navigate to a directory and enter a name in the **File name** text field. Click **Save**.

Configuring Payload Sampling

To configure payload sampling when available:

1. Run the `runagentsetup.bat` file.
2. Select **Modify Connector**, then click **Next**.
3. Select **Add, Modify, or Remove Destinations**, then click **Next**.
4. Select the destination for which you want to configure batching, then click **Next**.
5. Select **Modify destination settings**, then click **Next**.
6. Select **Payload Sampling (When Available)**, then click **Next**.
7. Specify the following information, then click **Next**.

Name Field	Value Field
Maximum Length	You can configure the maximum length of the payload sample using the following values: • Discard • 128 bytes • 256 bytes • 512 bytes • 1 Kbyte When the Discard option is chosen, no payload sample is sent inside the original event. Default is **256 bytes**.
Mask Non-Printable Characters	(**False** | True) This setting allows you to mask the non-printable characters in the payload sample. Default is **False**.

8. Select **Done with editing destination settings**, then click **Next**.
9. Click **Exit**.

Configuring Filters

Agent severity is the translation of the device severity into normalized values. For example, some connectors use a device severity scale of 1-10, whereas others use a scale of high, medium and low. These values are normalized into a single agent severity scale. The default scale is Low, Medium, High, and Very High. An event can also be classified as Unknown if the data source did not provide a severity rating.

To configure Filters:

1. Run the `runagentsetup.bat` file.
2. Select **Modify Connector**, then click **Next**.
3. Select **Add, Modify, or Remove Destinations**, then click **Next**.
4. Select the destination for which you want to configure batching, then click **Next**.
5. Select **Modify destination settings**, then click **Next**.
6. Select **Filtering**, then click **Next**.
7. Specify the following information, then click **Next**.

Configurable Settings

Name Field	Value Field
Filter Out	Filters for SmartConnectors are exclusive (filter out). Events that meet the connector filtering criteria are **not** forwarded to the destination. During SmartConnector set up, you can configure the connector to use filter conditions that do **not** pass events to the destination according to specific criteria. For example, you can use filters to exclude events with certain characteristics or events from specific network devices.
Very High Severity Event Definition	A filter condition to sort for very high severity events.
High Severity Event Definition	A filter condition to sort for high severity events.
Medium Severity Event Definition	A filter condition to sort for medium severity events.
Low Severity Event Definition	A filter condition to sort for low severity events.
Unknown Severity Event Definition	A filter condition to sort for unknown severity events.

8. Select **Done with editing destination settings**, then click **Next**.
9. Click **Exit**.

Managing SmartConnector configurations

Modifying SmartConnector settings

You can modify the connectors parameters you initially configured through the wizard, including destination parameters, service settings, and setting global parameters.

- **To change configured settings:**

  After installing a connector, you can run the installation wizard either by running the executable or by using the following command from `$ARCSIGHT_HOME/current/bin`, in a command prompt:

  `runagentsetup`
- **To Modify Connector Parameters:**

  Select **Modify Connector**, select **Modify connector parameters**, then make the relevant changes.
- **To Modify Destination Setting:**

  Select **Modify Connector** > **Add, modify, or remove destinations**, select the destination that you want to modify, then click **Modify destination Settings**.
- **To Modify Destination Parameters:**

  Select **Modify Connector** > **Add**, **modify**, **or remove destinations**, select the destination that you want to modify, then click **Modify destination Parameters**.

  Note: The **Modify destination Parameters** option is not applicable for the ArcSight SaaS destination.
- **To Modify the Unique Generator ID**

  Open the `agent.properties` file and add the following line:

  ```
  agent.generator.id=<unique_generator_id>
  ```

Managing SmartConnector filter conditions

For SmartConnectors that have ArcSight Manager as the destination, you can apply filters through the ESM Console. For more information, see Managing SmartConnector Filter Conditions in the ArcSight Console User's Guide for ESM. A filter applied through the ESM Console only applies to the events sent to that ESM.

Note: When selecting **Aggregate Events**, the **Preserve Raw Event** feature is disabled.

For all other destinations, the filter must be expressed in text. For example, you can write filtering strings such as:

```
Name EQ “Agent”
(name Contains “Super”) Or (name EQ “Agent”)
attackerAddress Between (“10.0.0.1”, “10.0.0.10”)
destinationAddress Is “NOT NULL”
```

The following table lists operators that can be used:

Usable Operations	Description
EQ	equal to
NE	not equal to
LT	less than
LE	less than or equal to
GE	greater than or equal to
GT	greater than
Between	compares any specified range
ContainsBits	equal to, for bitmap fields
In	standard CCE operator for membership test
Contains	contains the specified string
StartsWith	starts with the specified string
EndsWith	ends with the specified string
Like	standard CCE operator for simple pattern matching for string type: _ wildcard for single character, % wildcard for any number of characters
InSubnet	for IP address that is not the specified subnet
InGroup	for asset in the specified asset category or zone in the specified zone group
Is	tests true for the selected state, “NULL” or “NOT NULL” . Do not use all uppercase of “Is”.

For more information about data fields, event mappings, and CEF fields, see the “Data Fields,” “Audit Events,” “Cases,” and “Events” sections in *ArcSight ESM User’s Reference*.

Managing customized event filters

Use customized events filtering to remove events that are not of interest, or include only the events that are of interest, to your organization before they are counted. Filtering is performed based on certain predefined patterns.

By default, this feature is not enabled. If enabled, you can either include only the events that have a specific pattern in the raw event field, or exclude all the events that have a specific pattern. Use the Get Status command at any point in time when the connector is running to see:

- the total number of events filtered out after starting the last connector
- the status of the events filtering

All connector destinations subsequently receive only the relevant events based on the filtering defined.

Configuring Custom Event Filter

The custom event filtering feature applies to the raw event field in the ArcSight security event. During the flow of the security events through the connector, the raw event field is extracted and evaluated to apply the filter.

This feature only impacts the events that have a non-empty `rawEvent` field. All device events have the raw events field present when they reach the connector, and will be impacted by using this feature. Some internal events, such as `agent:017` (get status), also have the `rawEvent` field present in the event and will be impacted by the filtering feature. Most of the internal events, such as `agent:030`, `agent:031`, or `agent:050` do not have the `rawEvent` field in the event and will not be impacted.

By default, the feature is disabled (`customeventsfilter.regex.enabled=false`) and no filtering is applied to any events.

To use the feature, add the following line to the `agent.properties` file.

```
customeventsfilter.regex.enabled=true
```

Enter a valid Java Regex Pattern in **one** of the following properties:

```
customeventsfilter.regex.pattern.include=
customeventsfilter.regex.pattern.exclude=
```

For examples about Java Regex Patterns, see Examples of Patterns.

If a bad regex (un-compilable by JAVA Pattern class) is used, an error message is logged in the `agent.log` file. See Log Messages.

Note:

1. If the feature is enabled and both patterns are inadvertently defined, the exclude pattern takes precedence and the include pattern is ignored.
2. Enabling the filter through an include pattern filters out all the events in the raw event field that do not have the pattern in question. Therefore, be certain of the outcome that you want to achieve before enabling the include filter.
3. All properties are considered unique to the agent. Therefore, avoid defining any property multiple times for either the include or exclude patterns.

If the feature is enabled and the pattern specified for both include and exclude pattern fields is invalid, then the Get Status command shows a message similar to the following for the filtering state:

```
Custom Filtering: Events Filtering State.....Events Filtering Disabled
Due to Syntax Error in User Defined Regex
```

The following table shows the various states of the filter under different user entry combinations:

customeventsfiler.regex.enabled	customeventsfilter.regex. pattern.exclude	customeventsfilter. pattern.include	Result
false	Any pattern (valid, invalid, or empty)	Any Pattern (valid, invalid, or empty)	The filtering is disabled.
true	Valid and non-empty pattern	Any Pattern (valid, invalid, or empty)	The filtering is enabled with exclude filter. Include pattern has no impact.
true	Empty or invalid	Valid pattern	The filtering is enabled with include filter.
true	Empty or invalid	Empty or invalid	The filtering is disabled.

Get Status

From the ESM Console

Use the `Get Status` command from the ESM Console to get the current filtering state and also the number of events filtered out by the feature since the last connector start.

In the ESM Console, right-click the connector and select **Send Command** > **Status**> **Get Status**.

The command is sent to the connector and the result set is displayed. In the results, contain two rows pertaining to the custom filtering feature.

From the Command Line

To get status from the connector command line, enter this command from the <ARCSIGHT_HOME>/current/bin:

```
arcsight agentcommand –c status
```

Examples of patterns

Patterns are compiled through the `java.util.regex.Pattern` class. Any non-empty pattern that can be compiled is considered a valid pattern. The following table shows a few examples of valid patterns and their results:

Example of Valid Pattern	Result
`customeventsfilter.regex.pattern.exclude= IPSec\\s+tunnel`	Filters out all the events that have the pattern `IPsec tunnel` in the raw event.
`customeventsfilter.regex.pattern.exclude= "Bad\\s+\\S+"`	Filters out all the events that have the pattern `"Bad anyWord"` in the raw event (including the double quotes).
`customeventsfilter.regex.pattern.exclude= 111.112.113.114`	Filters out all the events that have the IP `111.112.113.114` in the raw event.
`customeventsfilter.regex.pattern.include= remote_peer-_ip\\s*\=\\s*\\d+\.\\d+\.\\d+\.\\d+`	The filtering feature is enabled (provided that the exclude pattern is empty) through the include filter to allow only the events that have the pattern, for example, `remote_peer-_ip = 11.12.13.14` in the raw event to pass through.

The following 10 messages are actual raw events. Examples of how the filtering can be used to include or exclude events from these 10 raw events are provided in the four cases that follow this list.

1. Nov 28 22:03:21 10.0.111.2 Nov 28 2016 22:02:17: %PIX-6-106015: Deny TCP (no connection) from 101.102.103.104/3671 to 10.0.111.22/80 flags RST ACK on interface inside
2. Nov 28 22:03:21 10.0.111.2 Nov 28 2016 22:02:17: %PIX-2-106006: Deny inbound UDP from 10.0.65.116/2908 to 10.0.126.55/123 on interface outside
3. Nov 28 22:03:53 10.0.111.2 Nov 28 2016 22:02:49: %PIX-2-106020: Deny IP teardrop fragment (size = 32, offset = 0) from 101.102.103.104 to 10.0.126.55

4. Nov 28 22:04:09 10.0.111.2 Nov 28 2016 22:03:04: %PIX-2-106001: Inbound TCP connection denied from 10.0.65.116/3694 to 10.0.126.55/23 flags SYN on interface outside
5. Nov 28 22:04:10 10.0.111.2 Nov 28 2016 22:03:05: %PIX-3-305005: No translation group found for tcp src inside:10.0.112.9/37 dst outside:10.0.65.116/3562
6. Nov 28 22:04:44 10.0.111.2 Nov 28 2016 22:03:39: %PIX-2-106001: Inbound TCP connection denied from 10.11.12.13/3699 to 10.0.126.55/8080 flags SYN on interface outside
7. Nov 28 22:05:07 10.0.111.2 Nov 28 2016 22:04:02: %PIX-4-500004: Invalid transport field for protocol=17, from 10.0.142.116/1234 to 10.0.126.55/0
8. Nov 28 22:05:25 10.0.111.2 Nov 28 2016 22:04:20: %PIX-2-106020: Deny IP teardrop fragment (size = 36, offset = 0) from 10.11.12.13 to 10.0.126.55
9. Nov 28 22:06:01 10.0.111.2 Nov 28 2016 22:04:57: %PIX-2-106012: Deny IP from 10.0.142.116 to 10.0.126.55, IP options: "0x1f"
10. Nov 28 22:06:10 10.0.111.2 Nov 28 2016 22:05:05: %PIX-3-305005: No translation group found for tcp src inside:10.0.112.9/37 dst outside:101.102.103.104/3562

The following cases describe the results of four distinct filtering cases on the above raw events.

Case 1:

```
customeventsfilter.regex.enabled=true
customeventsfilter.regex.pattern.exclude=Deny IP.*from \\d+\.\\d+\.\\d+\.\\d+
```

Events #3, #8, and #9 will be dropped (excluded) from the flow. This pattern is meant to exclude all raw events that have both the patterns <Deny IP> and <from IPaddress> in the same raw event.

Case 2:

```
customeventsfilter.regex.enabled=truecustomeventsfilter.regex.pattern.exclude=
(10.11.12.13)|(101.102.103.104)
```

Events #1, #3, #6, #8, and #10 will be dropped (excluded) from the flow. The pattern is meant to exclude raw events that have the IPs 10.11.12.13 or 101.102.103.104.

Case 3:

```
customeventsfilter.regex.enabled=truecustomeventsfilter.regex.pattern.include=
(10.11.12.13)|(101.102.103.104)
```

Events #2, #4, #5, #7, and #9 will be dropped (excluded) from the flow. The pattern is meant to include raw events that have the IPs 10.11.12.13 and 101.102.103.104 in them (both IPs do not need to be in the same pattern). All other events that do not have either of the IPs will be dropped.

Case 4:

```
customeventsfilter.regex.enabled=false
customeventsfilter.regex.pattern.include=(10.11.12.13)|(101.102.103.104)
```

No filtering will be done because the enabled property is false.

Log Messages in `agent.log`

During connectors initialization, information and error messages regarding the filtering states and the patterns are printed in the `agent.log` file. The following lines are excerpts from the `agent.log` file. This shows an instance when the user defined an invalid regex in the exclude pattern:

```
[2017-03-24 16:07:54,485][INFO ][default.com.arcsight.agent.loadable._
CustomEventsRegexFilter][init] CustomEventsRegexFilter Initialized: Filtering Enabled
=true, Exclude Regex =remote_peer_ip\s+\is\s+\d+.\d+.\d+.\d+, Include Regex =

[2017-03-24 16:07:54,485][ERROR][default.com.arcsight.agent.loadable._
CustomEventsRegexFilter][init] Unable to compile custom filter exclude regex=remote_
peer_ip\s+\is\s+\d+.\d+.\d+.\d+

[2017-03-24 16:07:54,500][INFO ][default.com.arcsight.agent.loadable._
CustomEventsRegexFilter][init] Events Filtering Disabled Due to Syntax Error in User
Defined Regex
```

Configuring Log Rotation

You can configure log rotation in file based SmartConnectors to limit log data, avoid overflow of record store, and maintain smaller log files.

Log Rotation Types

There are three mechanisms for rotating log files:

Name Following Log Rotation: An example would be, the device writes to xyz.log. At rotation time, the device renames xyz.log to xyz1.log and creates a new xyz.log and begins to write to it. The connector detects the drop in size of xyz.log and terminates the reader thread to the old

xyz.log after processing is completed. The connector creates a new reader thread to the new xyz.log and begins processing that file.

Daily Rotation: A typical scenario could be, the device writes to xyz.timestamp.log on a daily basis. At a specified time, the device creates a new daily log and begins to write to it. The connector detects the new log and terminates the reader thread to the previous log after processing is complete. The connector then creates a new reader thread to the new xyz.timestamp.log and begins processing that file.

Index Rotation: In this case, the device writes to indexed files - xyz.log.001, xyz.log.002, xyz.log.003 and so forth. At startup, the connector processes the log with highest index. When the device creates a log with a greater index, the connector terminates the reader thread to the previous log after processing completes, creates a thread to the new log and begins processing that log. To enable this log rotation, set`rotationscheme` to Index.

Configuring Log Rotation

To configure log rotation, edit the agent.properties file after the installation of SmartConnector

1. Open the `agent.properties` file located at `$ARCSIGHT_HOME\current\user\agent`.
2. To enable Daily log rotation, set `rotationscheme` to Daily, and set `rotationschemeparams`, as shown in the following example:

   ```
   agents[x].rotationscheme="Daily"
     agents[x].rotationschemeparams="FilePrefix,DateFormat,FileSuffix"
   ```

   Where, for a data file name of `foo.2013-09-23.log`

   ```
   FilePrefix = foo
     DateFormat = yyyy-mm-dd
     FileSuffix = .log
   ```
3. To enable Index log rotation, set `rotationscheme` to Index, and set `rotationschemeparams`, as shown in the example below:

   ```
   agents[x].rotationscheme="Index"
     agents
   [x].rotationschemeparams="FilePrefix,FileSuffix,Digits,Count,Optional true
   or false"
   ```

   Where for a data file name of `foo.log.%03d,001,999,false`
4. To enable Name Following log rotation, set `followexternalrotation` to true.
5. Save the file and restart the connector for your changes to take effect.

Configuring the Reconnecting Feature for Load Balancer

If you have a multiple tier connector installation where there is a Load Balancer between tiers, you can use the reconnect feature for better load balancing . For example, without the reconnect feature, tier 1 connectors start up and make a connection to the CEF syslog destination (tier 1). The load balancer makes a load balancing decision at the time of the initial connection and the tier 1 connector always sends to the same tier 2 connector.

With the `reconnect` parameter, the tier 1 connector makes an initial connection to the tier 2 connector as before and the Load Balancer makes a load balancing decision and picks a tier 2 connector. After the reconnect timeout, the tier 1 connector makes a new connection and the Load Balancer makes a new load balancing decision and selects a tier 2 connector, that could be different from the previous tier 2 connector. This distributes the load evenly across the tier 2 connectors over time.

To configure the reconnect parameter:

1. Open the `$ARCSIGHT_HOME/current/user/agent/agent.properties` file.
2. Locate the following parameter to edit:
   `agents[0].destination[0].params`
3. Change the value for the reconnect parameter from **-1** to a value, which indicates the time in seconds the CEF Syslog destination must stay open before it gets disconnected and attempts to reconnect.

   For example, to disconnect and reconnect every 60 seconds, change as follows:

   ```
   <Parameter Name\="reconnect" Value\="-1"/>\n
   ```

   to

   ```
   <Parameter Name\="reconnect" Value\="60"/>\n
   ```
4. Save and close the `agent.properties` file.

Configuring Persistent SmartMessage Transport

You can configure the SmartMessage transport to be persistent to achieve higher throughput for Logger destinations. This is useful if SmartConnectors experiences any issues in sending a batch of events to Logger due to network errors.

When a connector is unable to send events to Logger, the following symptoms might be noticed in the log:

- Logger ping tests fail frequently
- EPS drops down

- Heartbeat transport and event transport links sporadically go up and down
- Longer roundtrip times might be observed for 'event sent' acknowledgment
- Events fail to be sent
- Caching

Note: Changing the persistent value to **true** is not recommended if there are more than 250 Logger connections.

To configure persistent SmartMessage Transport for logger pool destinations:

1. Open the `$ARCSIGHT_HOME/current/user/agent/agent.properties` file.
2. Set the following parameters:

```
transport.loggersecure.connection.persistent=true
```

```
transport.loggersecurepool.connection.persistent=true
```

Specifying IP address on devices with Multiple Network Interfaces

You can configure devices with multiple network interfaces or multihomed hosts, to let the connector choose the interface with the lowest numerical IP address to report its host address.

If the interface is not the one your device is using to communicate with another device, set the `connector.network.interface.name` property as follows:

To set this parameter:

1. After the connector installation, go to `$ARCSIGHT_HOME\current\user\agent`.
2. Go to the end of the `agent.properties` file and add the `connector.network.interface.name` parameter.
3. Enter the interface name.
4. Save the file.
5. Restart the connector.

To get the Name of the Network Interface:

- **On Linux:** Run the **`ifconfig`** command in the terminal.
  The interface names ethø and lo appear in the left column.

- **On Windows:**: Do the following:

  From `$ARCSIGHT_HOME\current\system\agent\config\list_net_interfaces`, run the `run.bat` file.

  The interface names appear in the first brackets of each interface (lo, eth3, net5, eth4).

Defining default and alternate configurations from ArcSight Console

A SmartConnector can have a default and multiple alternate configurations.

An alternate configuration is a set of runtime parameters that is used instead of the default configuration during a specified time. For example, you can specify different batching schemes (by severity or size) for different time ranges during a day. You can define more than one alternate configuration per destination and apply them to the destination for different time ranges during the day. For example, you can define a configuration for 8 am to 5 pm time range and another configuration for the 5 pm to 8 am time range.

If the time ranges of the combined alternate configurations do not span 24 hours, the default parameters will be used to cover the time intervals not already defined in the alternates.

To define default configurations:

1. In the Navigator panel, choose the **Connectors** resource tree.
2. Right-click the SmartConnector you want to manage and select **Configure**.

   This opens the **Inspect/Edit** panel for the **Connector Editor**.
3. On the **Connector** tab, type the **Connector Location** and the **Device Location**. All events are tagged with these fields by the SmartConnector. The creation date and other information are automatically populated.
4. On the **Default** tab, change any additional Batching, Time Correction, or other parameters as desired. See the *ArcSight Console User's Guide*, "Managing SmartConnectors", for configuration field descriptions in the "Connector Editor Option Tabs" and "Connector Tab Configuration Fields" sections.
5. Click **Apply** to add your changes and to keep the Connector Editor open.

The description entry associated with the setting provides tool tip information. These parameters are not localized since they come directly from the connector and the connector might contain new resources or might be a newer version.

The framework for connector commands operates similarly. Configuration of the connector command menu is achieved by sending the list of commands that are supported on the connector at registration time.

There are several controls you can adjust in the Connector Editor. The variety of options are best summarized by briefly describing what's available at each of the editor's tabs or subtabs.

To create alternate configurations:

1. Open the Inspect or Edit panel of the SmartConnector.
2. On the **Default** tab, click **Add Alternate**.

   A new tab, Alternate #1, is added to the edit panel. The alternate tab provides fields for entering a time interval.
3. Under **Time Interval**, enter times for **From** and **To**. Make additional changes as required, then click **Apply**.
4. Repeat the process if you want additional alternates using different time intervals and different parameters. For example, create alternates if you want the varying batching schemes based on the severity or size on certain times of the day.

For more information about ArcSight Management Center implementations, see the *ArcSight Management Center Administrator's Guide*, "Managing Alternate Configurations".

Configuring multiple lines of table parameters

During connector installation, some connectors require table parameters to be entered. SmartConnectors for which parameter tables are used includes multiple files, multiple sites or servers, and multiple database instance connectors.

You can either enter parameters manually for few lines or use a `.csv` file to populate many lines of parameter data. You can also export the populated data into a `.csv` file.

Note the following when using this feature:

- After importing a `.csv` file, data in private columns remain hidden.
- Although you can manually enter a private column either by adding the column to your `.csv` file or by filling it through the Configuration Wizard, the column will not appear in any exported files. This is a precautionary measure.
- Importing data from a `.csv` file overwrites the existing data in the table.
- After exporting the connector host list (table parameters) that is longer than 588, all 588 records are downloaded appropriately. However, record 589 gets truncated and all other information past entry 589 is lost. Note that, there is no error message from ArcMC for this limitation.

To configure multiple lines of parameters:

1. Enter multiple lines of parameter data into a spreadsheet and save it as a `.csv` file.
2. During connector installation, click **Import** to locate and import the `.csv` file you created.
3. To add more rows manually, click **Add**, then specify the relevant details.

   Note: The example above shows a "Password" column within the Configuration Wizard that does not appear in the original `.csv` file. This private column does not contain actual password data and **will not be included in an exported file**.

4. To export data into a `.csv` file, click **Export**.

Configuring Connector with third-party application

Server Name Indication (SNI) is a TLS extension, which is defined in RFC 4366. It enables TLS connections to virtual servers in which multiple servers with different network names are hosted under a single IP address.

In the SmartConnector environment, the `Djsse.enableSNIExtension` property, which is used to enable or disable SNI, is disabled by default.

Enabling SNI Manually

Based on the connector installation, use any of the following options to enable SNI:

Note: For REST API connector, ensure that the events URL is correct.

1. **For standalone installations**: Open the following files from the `current\bin\script` folder, then set the `-Djsse.enableSNIExtension` property to **True**:
   - `connectors.bat`
   - `jvmcommonparams.bat`
2. **For service installations**: Open the `agent.wrapper.conf` file from the `current\user\agent` folder, then set the `-Djsse.enableSNIExtension` property to **True**.

Because of SNI, the following certificate exception might be displayed while configuring the connector with third-party application:

```
Error[1]: RemoteException: cause[javax.net.ssl.SSLHandshakeException: PKIX
path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find
valid certification path to requested target
```

To fix this issue, see Certificate Issue while Integrating Connector with Third-party Application.

Managing Compression

Compression lowers the overall network bandwidth used by connectors dramatically without impacting their overall performance. By default, all connectors have compression enabled. Connectors send event information to the Manager in a compressed format using HTTP compression. The compression is provided at the rate of 1 to 10 or greater, depending on the input data or the events sent by the connector.

To disable compression, add the following line to the `ARCSIGHT_HOME\current\user\agent\agent.properties` file:

```
http.transport.compressed = false
```

Enabling FIPS Support

Federal Information Processing Standards (FIPS) are a set of rules and regulations defined by the United States government that specify the security requirements for data processing and communication between the components. To know more about FIPS, see Understanding FIPS.

Important: Before installing any connector, ensure that the random number pool (also known as entropy pool) of Operating System must not be less than the ideal lower limit of 3290. For more information, see SmartConnector Remote Connections Failing Due to Low Entropy.

To enable FIPS mode during installation, you can select **Global Parameters** > **FIPS Mode**. You can also enable this option after installation, by running the `<Installation_directory>/Current/Bin/runagentsetup.bat` file.

- Manually Enabling FIPS Support
- Enabling FIPS Suite B
- Limitations

Manually enabling FIPS support

You can enable FIPS support to connectors in the installation wizard during installation process. If you are installing SmartConnector on an appliance, you can enable FIPS support through the user interface. To do this, enable support on the container or containers containing the connector for which you want to enable support.

If you did not enable FIPS support during installation, you can manually enable support after the installation by using the following procedures:

Note: Refer to FIPS Compliance Limitation to understand the limitations for some of the SmartConnectors.

Manually enabling FIPS mode

1. From `$ARCSIGHT_HOME/current/user/agent`, open the `agent.properties` file.
2. Enter the following property: `fips.enabled=true`
3. Save and close the `agent.properties` file.
4. Restart the connector to apply the changes.

Enabling FIPS Suite B mode

To enable FIPS Suite B Mode from the Installation Wizard:

1. After completing the installation, execute `runagentsetup` from the `$ARCSIGHT_HOME\current\bin` directory.
2. On the window displayed, select **Modify Connector**.
3. Select **Add**, **Modify**, or **remove destinations** and click **Next**.
4. Select the destination for which you want to enable FIPS Suite B mode and click **Next**.
5. Select **Modify destination parameters** and click **Next**.
6. When the parameter window is displayed, select **FIPS with Suite B 128-bits** or **FIPS with Suite B 192 bits** for the FIPS Cipher Suites parameter. Click **Next**.
7. The window displayed shows the editing changes to be made. Confirm and click **Next** to continue. (To adjust changes before confirming, click **Previous**.)
8. A summary of the configuration changes made is displayed. Click **Next** to continue.
9. Click **Exit** to exit the configuration wizard.

Manually enabling FIPS Suite B support

If you have installed a SmartConnector in FIPS-compliant mode, you can manually enable FIPS Suite B support by modifying the ESM destination parameters in the `agent.properties` file as follows:

Note: The destination must also be installed in FIPS Suite B mode.

1. From `$ARCSIGHT_HOME\current\user\agent`, open the `agent.properties` file for editing.
2. Locate the following property for destination parameters (approximately, line 10 in the file):

   ```
   agents[0].destination[0].params=<?xml version\="1.0" encoding\="UTF-8"?>\n<ParameterValues>\n <Parameter Name\="port" Value\="8443"/>\n <Parameter Name\="filterevents" Value\="false"/>\n <Parameter Name\="host" Value\="samplehost.sv.arcsight.com"/>\n <Parameter Name\="aupmaster" Value\="false"/>\n <Parameter Name\="fipsciphers" Value\="fipsDefault"/>\n</ParameterValues>\n
   ```

3. The destination parameters are specified here as an XML string where each element is one parameter. Based upon the Suite B mode of the destination, change `fipsDefault` to `suiteb128` (for 128-bit security) or `suiteb192` (for 192-bit security).

4. Save and close the `agent.properties` file.
5. Restart the connector for your changes to take effect.

Limitations

There are certain limitations in implementing FIPS compliance for the following connector destinations:

CEF Syslog as the Destination

If you choose **CEF Syslog** (with TLS protocol) as the destination for the connector, the wizard attempts to retrieve the security certificate from the destination and import it based upon your input. Although the CEF Syslog destination works as expected in FIPS-compliant mode, when you edit `agent.properties` to enable FIPS-compliant mode, the certificate retrieved from the destination may not be imported properly into the truststore.

If the SmartConnector wizard is unable to fetch and import the destination certificate, you can import the certificate manually:

1. Copy the certificate from the destination to a temporary location.
2. From the `$ARCSIGHT_HOME/current/bin` directory, execute the following command to import the certificate:

   ```
   arcsight keytoolgui
   ```
3. Open the keystore in `$ARCSIGHT_HOME/jre/lib/security/cacerts` (the password will be `changeit`).
4. From the **Menu** bar, select **Tools** and **Import Certificate**. Upload the certificate file.
5. Trust the certificate.
6. Start the connector and the device.

Microsoft SQL JDBC Driver

If you are running a database connector that uses the SQL JDBC driver *with encryption enabled*, the connector cannot be installed in FIPS-compliant mode.

See the configuration guide for the database connector you are installing for instructions for downloading and installing a Microsoft SQL Server JDBC driver.

Password management

Use the commands below to change your key and trust store passwords. Then update the `agent.properties` file with the new value.

To change password on a keystore or truststore:

1. Run the following command (see the table in the section that follow for store values):
   `bin/arcsight agent keytool -store <store value> -storepasswd`
2. Enter the new password as prompted.
3. Update the `agent.properties` file, according to the table below.

Note: Keystore files will not exist unless client authentication has been setup.

To change password of a key inside the key store:

A key entry uses the same password as the key store, so when changing the key store password, also change the key's password.

`bin/arcsight agent keytool -store agentkeys -keypasswd -alias <alias of key>`

Store Values

Key Store (for Client Authorization)	Trust Store
agentkeys	agentcerts

Entries for the `agent.properties` File

When changing passwords, make sure to add or update the corresponding property value in the `agent.properties.` file.

	Key Store (for Client Authorization)	Trust Store
FIPS	ssl.fips.keystore.password.encrypted=<new password>	ssl.fips.truststore.password.encrypted=<new password>
Non-FIPS	ssl.keystore.password.encrypted=<new password>	ssl.truststore.password.encrypted=<new password>

Upgrading Connectors

You can upgrade a smart connector to implement the newly implemented features, mapping improvements and overall functionality of a smart connector. You can upgrade connectors either locally or remotely. Connectors automatically determine their upgrade status when they start.

Note: For connectors running on Windows platforms, there is a known limitation for upgrading the connector from its ESM destination.

As part of the connector upgrade, some folders or files are moved from the old to the new version. Because, Microsoft Windows locks the folders or files even they are opened for a read, upgrades could fail if locked folders or files associated with the connector installation are accessed during the upgrade. To prevent this issue, start the connector from **Start > Programs**, so that no windows are opened to run the connector, thus reducing the possibility of locked folders or files.

Important: If you are running a 32-bit version of a SmartConnector, you upgrade it to the 64-bit version. To run the 64-bit implementation of the connector, you must install the 64-bit version of the SmartConnector.

Upgrade Considerations

- The versions of connectors that you want to upgrade must be available on the Manager to which you are connected. The remote upgrade option is available only for ESM 4.0 or later and connectors 4.0.2 or later.
- Both the Manager and the connectors you want to upgrade must be running.
- You must download current versions of the Connectors Configuration Guides from the support website and review information specific to the connector device that you are planning to upgrade.
- You must have the required administrative permissions.
- The supported upgrade path of a SmartConnector is from 8.x to any later versions. For example, you can directly upgrade from 8.0 to 8.4.

After Upgrading

- If the upgrade is successful, the new connector starts and reports successful upgrade status.

- If the upgraded connector fails to start, the original connector restarts automatically as a failover measure.

  Tip: If the connector fails to start:
  - Select **Send Command > Tech Support > Get Upgrade Logs** from the Console menu, to review the logs.
  - Use the Send Logs Wizard to collect and send logs, including upgrade logs, to support for help

Rolling Back to the Previously Installed Version

Perform the following steps to roll back hotfix SmartConnector 8.x.x.xxxx.0 updates:

1. Stop the SmartConnector.
2. Rename the sub folder from **current** to **8.x.x.xxxx.0**.
3. Unzip the previous release install folder (folder name with previous release build number).
4. Rename the subfolder from <***previous release build number***> to **current**.
5. Restart the SmartConnector.

Deleting Older Vulnerable Libraries after Upgrading a Connector

When you upgrade a Connector from local, ArcMC, or ESM, it creates a backup of the install directory of the existing connector to facilitate rollback in unforeseen scenarios.

Earlier versions of the connector might have libraries that were vulnerable and were upgraded to non-vulnerable later versions. This might require cleaning all vulnerable libraries from the system manually.

Note: Though the vulnerable libraries are present in the backup folder, the active connector instances do not use these files. Whether you delete the vulnerable libraries or not, these static files will not cause any harm.

Perform the following steps to delete the older vulnerable libraries manually:

Note: This disables the rollback ability. However, you can retain the backup of certain configurations, if required.

Option 1 – Delete only the vulnerable libraries

For Linux:

1. Run the following command: `cd $Arcsight_Home`

   The following folders will be displayed:

   - **current** (upgraded version of the connector)
   - **Xxxxx** (xxxx refers to the build number of connector before upgrade, for example: X8444)
2. Run the following command: `cd Xxxxx/lib/agent`
3. Run the following command to remove the log4j libraries: `rm -rf *log4j*`
4. Run the following command: `cd Xxxxx/system/agent/web/webapps/axis/WEB-INF/lib/`
5. Run the following command to remove the log4j libraries: `rm -rf *log4j*`
6. Run the following command: `cd Xxxxx/lib/agent/axis`
7. Run the following command to remove the log4j libraries: `rm -rf *log4j*`

For Windows:

1. Go to `$Arcsight_Home`.

   The following folders will be displayed:

   - **current** (upgraded version of the connector)
   - **Xxxxx** (xxxx refers to the build number of connector before upgrade, for example: X8444)
2. Open the `Xxxxx\lib\agent` folder.
3. Search for **log4j** and delete all the entries.
4. Open the `Xxxxx\system\agent\web\webapps\axis\WEB-INF\lib\` folder.
5. Search for **log4j** and delete all the entries.
6. Open the `Xxxxx\lib\agent\axis` folder.
7. Search for **log4j** and delete all the entries.

Option 2 - Delete the complete backup folder of the existing connector

For Linux:

1. Run the following command: `cd $Arcsight_Home`

   The following folders will be displayed:

- **current** (upgraded version of the connector)
- **Xxxxx** (xxxx refers to the build number of connector before upgrade, for example: X8444)

2. Run the following command to delete the backed up folder: `rm -rf Xxxxx` (for example: `rm-rf X8444`)

For Windows:

1. Go to `$Arcsight_Home`.

   The following folders will be displayed:

   - **current** (upgraded version of the connector)
   - **Xxxxx** (xxxx refers to the build number of connector before upgrade, for example: X8444)

2. Delete the **Xxxxx** folder manually.

Upgrading Connectors locally

To locally upgrade a connector:

Note: If you are upgrading from a Connector version that does not support GEID to a version that supports, add the following line to the agent.properties file before upgrading the connectors:

```
agent.generator.id=<generator_id>
```

1. Stop the connector.
2. Run the connector installer.
3. Select the location of the connector that you want to upgrade.
4. Select the option to continue and upgrade the connector.

   The upgraded connector is installed in the `$ARCSIGHT_HOME\current` folder.

   After the upgrade is complete, the previously installed folder is zipped and renamed to **X<build_number>.zip**. For example, **X8833.zip**.

Upgrading Connectors remotely from ArcSight Management Center

Important: Only Windows, Linux, and Solaris platforms are supported for connectors remote upgrade from the Console.

Note: If you are upgrading from a connector version that does not support GEID to a version that supports it, add the following line to the `agent.properties` file before upgrading the connectors:

```
agent.generator.id=<generator_id>
```

ArcSight Management Center (ArcMC) provides the ability to upgrade SmartConnectors remotely. Use the **Upgrade** command on the Console to launch, manage, and review the status of upgrades for all connectors. All communication and upgrade processes between components such as Console, Manager, and connectors) take place over secure connections.

The Console reflects current version information for all your connectors.

After the upgrade is complete, the previously installed folder is zipped and renamed to **X<build_number>.zip**. For example, **X8833.zip**.

Upgrading Connectors from ESM

1. Download the latest connector upgrades to the Manager From the OpenText SSO site. Upgrade version files are delivered as `.aup` files (a compressed file set).
2. Copy the `.aup` file to `ARCSIGHT_HOME\updates\` on a running Manager. The Manager automatically unzips the `.aup` file and copies its content to `ARCSIGHT_HOME\repository\`.
3. From the Console, select connectors to be upgraded (one at a time) and launch the **upgrade** command for each of them.

   The selected connectors upgrade themselves, restart, and send upgrade results (success or failure) back to the Console through the Manager.

   Caution:

   - If you have installed multiple connectors in a single JVM, select the first connector installed in the JVM and launch the upgrade command to upgrade all connectors in the JVM. If you select any other connector, the upgrade fails.
   - If your connector has multiple Manager destinations, you must perform this process from the primary Console. Any attempt to upgrade from a secondary or non-primary Console destination fails.

   After the upgrade is complete, the previously installed folder is zipped and renamed to **X<build_number>.zip**. For example, **X8833.zip**.

Upgrading to the New AES-GCM Data Encryption Scheme

From SmartConnector 8.1.0 and on, connectors use a new AES-GCM data encryption scheme, which replaces AES-CBC. If you cannot use a more recent version of CEF Encrypted Syslog (UDP)

as a destination, complete the following steps to change the encryption scheme back to AES-CBC.

To upgrade the connector from both source and destination to CE 24.4 or higher:

1. Go to `$ARCSIGHT_HOME$\current\user\agent\agent.properties`.
2. Find your destination configuration. For example:

   ```
   agents[0].destination[0].params=<?xml version\="1.0" encoding\="UTF-
   8"?>\n<ParameterValues>\n <Parameter Name\="cefver" Value\="0.1"/>\n
   <Parameter Name\="sharedKey"
   Value\="OBFUSCATE.4.9.0\:YRav1HUqxp8+D0V+oEbyRv11noZhnMzjUwdfIhDcmE1HGM/6H
   qymDOu8dYk\="/>\n <Parameter Name\="protocol" Value\="Encrypted UDP"/>\n
   <Parameter Name\="reconnect" Value\="-1"/>\n <Parameter Name\="port"
   Value\="514"/>\n <Parameter Name\="host" Value\="<Your host>"/>\n
   <Parameter Name\="forwarder" Value\="true"/>\n <Parameter
   Name\="encryptionScheme" Value\="AES-GCM"/>\n</ParameterValues>\n
   ```

3. Change `Value\="AES-GCM"` to `Value\="AES-CBC"`.
4. Save the `agent.properties` file.

ArcSight Update Packs (AUPs)

This section describes the ArcSight Update Packs (AUPs) used to update content between the ESM Manager and connectors. AUP files may contain information that applies to connectors or ESM related updates.

ArcSight Content AUPs

AUP files provide a way to collect a set of files together and update ArcSight resources as well as distribute parsers to connectors. ArcSight continuously develops new connector event categorization mappings, often called "content." This content is packaged in ArcSight Update Packs (AUP) files. All existing content is included with major product releases, but it is possible to stay current by receiving up-to-date, regular content updates from OpenText Subscribers Choice. Contact OpenText SSO for details.

Content updates (`ArcSight-xxxx-ConnectorContent.aup`) can be downloaded from the Support website. They contain data that is then transferred to registered connectors. An AUP can provide updates for:

1. Event categorizations (Category Behavior, Category Object, etc.)
2. Default zone mappings (what IP maps to which zone by default)
3. OS mappings (when a network is scanned, where the asset is created)

Content such as filters, rules and dashboards are not provided by the AUP.

Note: ArcSight Management Center does not support the automatic deployment of an AUP.

As shown below, the method of uploading an AUP varies depending on the ArcSight product.

ESM

Content updates are available from support. To update:

1. Download the latest AUP release.
2. Copy the `.aup` file to `ARCSIGHT_HOME\updates\` onto a running ESM Manager. SmartConnectors registered to this ESM automatically download the `.aup` and, once completed, an audit event is generated.

ESM or Logger

A Connector can send events to ESM and Logger simultaneously. In this configuration, it's helpful to use the AUP Master Destination feature. AUP Master Destination allows ESM to push AUP content to the Connector used for its Logger destination(s). Logger is not capable of storing or pushing its AUP content.

1. Using the Connector Configuration Wizard, add the ESM destination and set the AUP Master Destination parameter to **true** (the default is false).
2. If you have not already done so, you can also add the Logger destination.
3. Copy the `.aup` file to `ARCSIGHT_HOME\updates\` on the running ESM Manager you added in step 1.

Connector

The AUP content is pushed from ESM to the connector, which then sends an internal event to confirm. If the AUP Master Destination flag was set for the ESM destination, that AUP content is used by the connector for Logger or any other non-ESM destinations.

> **Caution:** The AUP Master Destination flag should be set to **true** for only one ESM destination at a time. If more than one ESM destination is set and the flag is true for more than one, only the first is treated as master.
>
> Failover ESM destinations cannot be AUP Masters.

Logger

Logger has no facility to store or forward AUPs to connectors.

ESM Generated AUPs

Some AUPs are generated by ESM itself for internal maintenance and operation.

System Zones Updates

System Zones updates (for example, `system-zone-mappings_00000000000000000001.aup`) are generated by ESM when a change to the ArcSight System zones is detected, then transported to the necessary connectors. It contains the new System-Zone mappings so incoming events are attached to the correct zones or assets in ESM.

As System Zones are always present, all connectors connected to ESM routinely receive them as an AUP.

User Categorization Updates

User Categorization Updates, (for example, `user-categorizations_user_supplied_00000000001300014581.aup`) are generated by ESM when a user modifies the way an event is categorized through the Console tools. These updates are then transferred to the registered connectors to update the way the newly sent events will be categorized. This is generally used for categorizing custom signatures for which ArcSight does not provide categorization.

User Zones Updates

User Zones updates(for example, `user-zone-mappings_3RxkkOxYBABDRZlZyr6nrWg==_00000000001700001895.aup`) are generated by ESM when a change to a user-created zone configuration is detected, then transported to the necessary connector. It contains the new zone mappings so that incoming events are attached to the correct zones or assets in ESM.

Uninstalling a SmartConnector

Before uninstalling a connector that is running as a service or daemon, stop the service or daemon. Also, be sure to remove the service files using the following command: `$ARCSIGHT_HOME/current/bin/arcsight agentsvc -r`

The Uninstaller does not remove all the files and directories under the connector home folder. After completing the uninstall procedure, manually delete these folders.

To uninstall on Windows:

1. Open the **Start** menu.
2. Run the Uninstall SmartConnectors program found under **All Programs** > **ArcSight SmartConnectors** (or the name you used for the folder during connector installation).
3. If connectors were not installed on the **Start** menu, locate the `$ARCSIGHT_HOME/current/UninstallerData` folder and run the following command: `Uninstall_ArcSightAgents.exe`

Note: To perform a silent uninstall, run the command with the following parameters: `Uninstall_ArcSightAgents.exe -i silent`

To uninstall on UNIX hosts:

1. Change to the `$ARCSIGHT_HOME/UninstallerData directory`.
2. Run the following command: `./Uninstall_ArcSightAgents`

Note:

- The **UninstallerData** directory contains `file .com.zerog.registry.xml` with Read, Write, and Execute permissions for everyone. On Windows platforms, these permissions are required for the uninstaller to work. However, on UNIX platforms, you can change the permissions to Read and Write for everyone (that is, 666).
- To perform a silent uninstall, run the command with the following parameters: `./Uninstall_ArcSightAgents -i silent`

Appendix - SmartConnector Audit Events

SmartConnector Audit Events

Audit events are events generated within the SmartConnector to mark a wide variety of routine actions that can occur manually or automatically.

DeviceEventClassId	Description
agent:000	AGENT
agent:001	Agent Connection
agent:002	Agent Reconnected
agent:003	Agent Zombie
agent:004	Agent Disconnect
agent:006	Unknown Agent Attempted to Connect
agent:007	AGENT_REGISTRATION_SUCCESS Agent was successfully registered with Manager
agent:008	AGENT_REGISTRATION_FAILURE Agent was not successfully registered with Manager
agent:009	AGENT_CONNECTION_REFUSED Manager rejected a connection attempt from an Agent for reasons other than authentication failure
agent:010	AGENT_UPGRADE_SUCCESS Agent upgrade succeeded
agent:011	AGENT_UPGRADE_FAILURE Agent upgrade failed
agent:012	AGENT_TIME_DEVICE_FAILURE Agent detected source events from a sensor device containing incorrect time stamps
agent:013	AGENT_DEVICE_FOUND Agent noted that a new sensor device is sending events
agent:014	AGENT_SYSLOG_AGGREGATION_FAILURE Agent could not find a base event referenced in a syslog aggregate event

DeviceEventClassId	Description
agent:015	AGENT_CONNECTION_DEVICE_FAILURE Agent could not connect to the sensor device's log
agent:016	AGENT_CONNECTION_DEVICE_SUCCESS Agent successfully connected to the sensor device's log
agent:017	AGENT_COMMAND_SUCCESS Agent successfully executed a command
agent:018	AGENT_COMMAND_FAILURE Agent could not execute a command
agent:019	AGENT_CACHE_CACHING Agent is caching events because they could not be immediately transmitted to the Manager
agent:020	AGENT_CACHE_EMPTY Agent has emptied its cache of events
agent:021	AGENT_NTCOLLECTOR_ERROR Agent could not communicate with an NT collector sensor
agent:022	AGENT_CONFIGURATION_FAILURE Agent could not process a reconfiguration request
agent:023	AGENT_CHECKPOINT_ERROR Agent could not communicate with a CheckPoint sensor
agent:024	AGENT_CHECKPOINT_WARN Agent is having difficulty communicating with CheckPoint
agent:025	AGENT_UPDATE_SUCCESS Agent content was successfully updated
agent:026	AGENT_UPDATE_FAILURE Agent content update failed
agent:027	AGENT_ACS_ERROR
agent:028	AGENT_UNEXPECTED_ERROR Agent experienced an unexpected problem
agent:029	AGENT_CACHE_DROPPED - Agent was forced to drop some of its cached data
agent:030	AGENT_STARTED Agent started

DeviceEventClassId	Description
agent:031	AGENT_SHUTTINGDOWN Agent shutdown
agent:032	AGENT_CONFIGURATION_CHANGED Agent configuration was successfully changed
agent:033	AGENT_DATABASE_PASSWORD_CHANGED The password used by an Agent to access a database has changed
agent:034	AGENT_DEVICE_UPDATED The Agent has been directed to monitor a different device (sensor)
agent:035	AGENT_TIME_FAILURE The Agent has detected event time stamps that fall outside the valid range
agent:036	AGENT_UPGRADE_STARTED
agent:037	AGENT_UPGRADE_ROLLBACK_STARTED
agent:038	AGENT_UPGRADE_ROLLBACK_SUCCESS
agent:039	AGENT_UPGRADE_ROLLBACK_FAILURE
agent:040	AGENT_INTEGRITY These warn about incoming non-internal events that have no raw event data. If the user does want to protect his event integrity, then these alerts should be given attention since they probably imply that a Connector has been improperly written such that events are being generated without raw event data
agent:041	AGENT_COMMAND_SENTTOAGENT
agent:042	AGENT_UNPARSED_EVENT
agent:043	AGENT_DEVICE_STATUS_MONITOR
agent:044	AGENT_FILE_STARTED
agent:045	AGENT_FILE_ENDED_SUCCESS
agent:046	AGENT_FILE_ENDED_FAILURE
agent:047	AGENT_FILE_COUNT_INCORRECT
agent:048	AGENT_LOG_ROTATION_ERROR
agent:049	AGENT_OVERRIDE_MISMATCH
agent:050	AGENT_RAWEVENT_STATISTICS
agent:100	AGENT_CONNECTION
agent:101	AGENT_CONNECTION_ESTABLISH Agent has just connected to Manager

DeviceEventClassId	Description
agent:102	AGENT_CONNECTION_ZOMBIE Agent is sending events but no heartbeats
agent:103	AGENT_CONNECTION_DROP Agent is sending neither events nor heartbeats
agent:104	AGENT_CONNECTION_UNKNOWN_AGENT An unknown Agent attempted to connect to the Manager
agent:105	AGENT_CONNECTION_ID_MISMATCH An Agent presented an incorrect shared secret when authenticating
agent:106	AGENT_SIDETABLE_OVERFLOW
agent:107	AGENT_SIDETABLE_OVERFLOW_DETECTE D_ON_AGENT_SIDE
agent:108	AGENT_CONNECTION_BLACKLISTED_AGE NT

agent:049

The **agent:049** event provides additional information about the parser file that is identified by the **parserIdentifier** field in security event. By default, the **agent:049** event is generated periodically that is once every 12 hours.

Note: To enable or disable the **agent:049** event, open the **agent.default.properties** file of the SmartConnector and set the **parser.versioning.identifier.map.enabled** parameter value to **true** or **false** respectively.

The **agent:049** event provides the following information about the parser file:

CEF Key Name	Value	CEF Key Name	Description of the Value
parserVersion	<The version of the parser file>	parserIdentifier	The unique id value representing the parser file.
cs1Label	Parser file	cs1	Name of the parser file.
cs2Label	Override status	cs2	True if the parser file is an override file.
cs3Label	Default signature	cs3	The signature of the parser file bundled inside AUP present in the connector.
cs4Label	Override signature	cs4	The signature of the Overridden parser file. This field will be absent in case of no override.
cs5Label	Override Parser file	cs5	The override parser file. This field will be absent in case of no override.
cs6Label	Override Parser version	cs6	The Override parser file version. This field will be absent in case of no override.

Troubleshooting

This section includes the following troubleshooting information:

The Raw Syslog destination is not available while deploying the Connectors in CHA

The Raw Syslog destination is not available for selection from the list of supported destinations for SmartConnectors that are freshly installed or upgraded through ArcMC Connector Host Appliance (CHA).

Workaround

You must add the **transport.types** property for each container available in CHA to enable the **Raw Syslog** destination in the destinations list.

To add the **transport.types** property (the container property located in the **agent.properties** file):

1. Log in to the ArcSight Management Center.
2. Click **Configuration Management** > **Bulk Operations** from the top-level menu bar.
3. Click the **Container** tab.
4. In the **Manage Containers** table, select the required container, and then click **Properties** at the top right of the table. The **Container Property Update** dialog box opens.
5. Click **Edit**.
6. Under **Property List**, enter a new property as **transport.types**, and then enter its **Value** as follows:

   ```
   http,loggersecure,cefsyslog,encryptedcefsyslog,loggersecurepool,cefkafka,simplesyslog
   ```
7. Click **Save**. The SmartConnetor will restart automatically.

The **transport.types** property is added to the list and the **Raw Syslog** destination will be available in the list of supported destinations for the SmartConnectors.

Events are not sent from SmartConnector to ArcSight SaaS

When **ArcSight SaaS** is configured as a destination, events are not sent from the SmartConnector to ArcSight SaaS. This is because the connector's access to **ArcSight SaaS**

might have been revoked. The events, instead, are cached in the connector and the following error message is displayed in the log file:

"Your access has been revoked. Please register again"

Workaround

To ensure that **ArcSight SaaS** receives events from the connector, you must re-register the **ArcSight SaaS** destination with a new registration URL. For more information, see "Re-registering a destination" on page 168.

Note: If you re-register the **ArcSight SaaS** destination, all cached events in the connector will be lost.

Connector upgrade remains incomplete with Azure Event Hub as destination through ArcMC

While performing the connector upgrade with Azure event hub as the destination through ArcMC, the connector is not restarting with all the new parameters. This is leading to an incomplete connector upgrade.

Workaround:

You must log in to the machine where the connector is installed, change the destination parameters and then restart the connector.

Certificate Issue while Integrating Connector with Third-party Application

Because of SNI, the following certificate exception might be displayed while configuring the connector with third-party application:

```
Error[1]: RemoteException: cause[javax.net.ssl.SSLHandshakeException: PKIX
path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find
valid certification path to requested target
```

Perform the following steps to fix this issue:

1. Stop the SmartConnector.
2. Enable SNI.
3. Restart the SmartConnector after enabling SNI to apply the changes.

Diagnosing Common Transformation Hub Issues

The following can help to diagnose common Transformation Hub issues.

502 Gateway Error (On Multi-Master Install)

If the initial master node is disabled and later re-enabled, the user may see a 502 Gateway error when attempting to log in to the Installer UI, preventing login. If this happens, restart all the `nginx-ingress-controller` pods in the cluster as follows:

1. Determine the **nginx** pod names: `kubectl get pods -n default`
2. For each **nginx** pod, run the following command: `kubectl delete pod <nginx name>`

Potential DNS Resolution Issue

The Transformation Hub application pods that depend on hostname resolution from DNS could fail. For example, the Schema Registry pod will be in a crash loop status, with the following error message in the Schema Registry logs:

```
# kubectl logs eb-schemaregistry-1138097507-1jxbn -n eventbroker
```

```
...
```

```
org.apache.kafka.common.config.ConfigException: No resolvable bootstrap urls
given in bootstrap.servers
```

```
...
```

The following steps will be useful in debugging this DNS resolution issue. The key is that the bootstrap host name given should be resolvable from within the pod, which can be verified as follows. Find the schema registry pod name:

```
# kubectl  get pods -n eventbroker1 | grep schemaregistry
```

```
eb-schemaregistry-2567039683-9l9jx   1/1       Running   1          18d
```

Find the configured bootstrap server:

```
# kubectl logs eb-schemaregistry-2567039683-9l9jx -n eventbroker1 | grep
"bootstrap.servers ="
```

```
bootstrap.servers = [<hostname>]
```

Use the `ping` command to check if the host name is DNS resolvable. If it is resolvable, you will see an output similar to the following example:

```
# kubectl exec  eb-schemaregistry-2567039683-9l9jx -n eventbroker1  -- ping -c
1 n15-214-137-h51.arst.usa.microfocus.com | grep transmitted
```

```
1 packets transmitted, 1 packets received, 0% packet loss
```

If the ping command is not successful, you will see an error:

```
# kubectl exec  eb-schemaregistry-2567039683-9l9jx -n eventbroker1  -- ping -c
1 bad.dns.hostname.arst.usa.microfocus.com | grep transmitted
```

```
ping: unknown host
```

If the host name is not resolvable, please check the DNS configuration on the system.

Transformation Hub Cluster Down

The number of nodes required to keep a Transformation Hub (TH) cluster operating depends on the replication factor. If the replication factor is only 1, which is not recommended, then all Kafka nodes in the TH cluster need to be up to make the TH cluster function correctly. In general, if the replication factor is N, then the system will tolerate up to N-1 server failures without losing any records committed to the log.

Pod Start Order

After deployment, pods are configured to start in the following order (downstream pods will not start until the dependencies are met.)

1. A quorum of ZooKeeper pods in the cluster must be up (2 of 3, or 3 of 5). The total number of ZooKeepers must be odd.
2. All Kafka pods must be up
3. Schema Registry pod must be up
4. Bootstrap Web Service, Transformation Hub Manager
5. Transformation Stream Processor, Routing Stream Processor

Cannot query ZooKeeper

This can occur when running the `kubectl` get pods command to get the status of the pods, downstream pods (as defined in the pod start order) does not stay up, and status is a 'CrashLoop'-type error.

- Ensure ZooKeeper pods are running.

- If the ZooKeeper pod status is **Pending**, you may not have labeled the nodes correctly (zk=yes). Verify that the nodes are labeled using the `kubectl get nodes -L=zk` command.
- Verify that you configured an odd number of ZooKeepers in the arcsight-installer.properties eb-zookeeper-count attribute.
- Check the ZooKeeper pod logs for errors using `kubectl logs <pod name> -n eventbroker1`.

Common Errors and Warnings in ZooKeeper logs

- **Quorum Exceptions:** A leader cannot be elected. If you see this type of error, check the conditions above.
- **Socket error:** this can occur if there are too many connections. The solution is to restart the pod using the `kubectl delete pod <pod_name> -n eventbroker1` command. The pod will be recreated automatically.

Common Errors and Warnings in Kafka logs

Cannot Register ID: In some cases, a broker node cannot register its ID. This can be caused by multiple broker nodes with the same ID. This is a rare situation that can occur when you are adding and removing nodes from the cluster and you do not define the cluster properly. Connect to each system running a Kafka broker and check the assigned `broker.id` value of each, in `/opt/arcsight/k8s-hostpath-volume/eb/kafka/meta.properties`. The broker.id value defined on each Kafka node must be unique.

SSL Connection Errors: These are warnings that occur if there is a connection issue between Kafka and consumer or producer.

Cannot communicate with other brokers: Host names may not be configured properly. It is possible that the node cannot perform reverse lookup or that DNS is not set up properly.

Transformation Hub default topics not created on first deployment: In this instance, the Bootstrap Web Service log contains 500 response code (the response from the Schema Registry), and topics are not created. Try undeploying the Transformation Hub containers, and then redeploy them.

One or more connectors cannot send data to Kafka: Check the following:

- The connection configuration is set properly in the connector.
- The encryption mode (TLS, TLS+FIPS, TLS+CA, TLS+FIPS+CA) is the same for both the Connector and Transformation Hub.
- Make sure you can connect to the Kafka port on the system and that there are no network issues.

Cannot retrieve the certificate error when connecting: Make sure that time is synced across all systems in the data pipeline.

• Check whether the Kafka pod is down. If you configured the connector with only one broker address, check whether the broker is down. If there are multiple brokers, they must be all configured in the connector as a comma-separated list.

• If the replication factor is set to 1 and a Kafka broker is down, data will not be sent through Transformation Hub. Fix the broker issue to bring it back up. In general, topics should be configured with a replication factor greater than **1** to prevent this scenario.

Kafka is resyncing: This may cause event throughput slowdown, but will not stop event flow.

An EB component crashes: Check the following:

- Check the container start up order (above). Have any of the dependency pods not started or crashed?
- It could be that the JVMs require more memory than the system has available.
- Check the number of open sockets.

Transformation Hub EPS is lower than expected: Check resource constants on Transformation Hub nodes, such as CPU, memory, or disk space. Also, check usage with ArcMC.

Network bottleneck: In this case, the Stream Processor is not able to keep up with transformation, or is resource-constrained in some way. In ArcSight Management Center, the Stream Processor metric will be lower than the connector EPS. Check that you have sufficient resources, memory, CPU.

Continuous network failures: This may be related to the management of TCP/IP resources. **TIME_WAIT** is the parameter that indicates the amount of time the node will take to finish closing a connection and the amount of time before it will kill a stale connection. Try reducing the value from its default. Edit the file `/etc/sysctl.conf` and add these lines to the end of it (or edit the existing values):

```
Decrease TIME_WAIT seconds
```

```
net.ipv4.tcp_fin_timeout = 10
```

```
Recycle and Reuse TIME_WAIT sockets more quickly
```

```
net.ipv4.tcp_tw_recycle = 1
```

```
net.ipv4.tcp_tw_reuse = 1
```

After editing the file, the following command:

```
$ sysctl --system
```

Diagnostic Data and Tools

Transformation Hub includes a diagnostic script (`eb-diag.sh`) for the collection of diagnostic data. **Diag.sh** is found in the web service container.

To run Transformation Hub diagnostic tools:

1. On the master server, find the Docker container ID of the web service container. (In this example 278e86760803)

```
$ docker ps | grep atlas_web-service
```

```
278e86760803        localhost:5000/arcsightsecurity/atlas_web-
service@sha256:c25b023afa7b7054de6aa188ed2802d24312f3c5de87b6537aa3e937476376d
8   "/bin/bash -c 'source"
```

```
$
```

2. Copy the script archive from web service container. (In this example 278e86760803)

```
$ docker cp 278e86760803:/eb/ws/eb_diag/eb_diag.tgz
```

```
$ tar -tzf eb_diag.tgz
```

```
vertica-diag.sh
```

```
eb-diag.sh
```

```
eb-sys-diag.sh
```

```
$
```

3. Extract the diagnostic script and run it.

```
$ tar -xvf eb_diag.tgz eb-diag.sh
```

```
$ sh eb-diag.sh
```

SmartConnector Installed on Windows Servers Taking Up Disk Space

SmartConnectors installed on Windows servers sometimes generate large .att files under the `<connector_home>\current\system\agent\web\webapps\axis\WEB-INF\attachments\` folder.

To reduce disk space consumption:

Edit the JAVA_TOOLS_OPTIONS variable by adding this value:
`-Djava.io.tmpdir=<newpath>\tmp\dir`

The path is changed to a non-existing folder and .att files are no longer generated. For more information, see the Oracle Documentation.

SmartConnector Remote Connections Failing Due to Low Entropy

All SmartConnector remote connections go through SSL and they depend on the Operating System random number pool (entropy pool) to generate private keys for secure communication. When the entropy pool is less than the ideal lower limit of 3290, the keys are not generated, communication cannot be established, and the SmartConnector does not start. In cloud hosted Linux instances, the entropy pool value can be less than 3290.

To increase the entropy pool value:

1. Install the `rng-tools` package by the following command:
   `sudo yum install -y rng-tools`
2. Check the entropy availability in the system by the following command:
   `cat /proc/sys/kernel/random/entropy_avail`
3. Enable and start the `rngd` service at boot by the following commands:
   `systemctl enable rngd.service`
   `systemctl start rngd.service`
4. Check the entropy availability in the system, after starting the `rngd` service by the following command:
   `cat /proc/sys/kernel/random/entropy_avail`

Master or Worker Nodes Down

This section describes the expected behavior if a master node or one or more worker nodes goes down.

- Kubernetes worker nodes will continue running even when the master is down, but if they reboot then they will not be able to find the master node and will fail.
- All services running on the master node will become unavailable.
- Transformation Hub Web Service running on the master node becomes unavailable.
  - The services (Routing Stream Process) and integration (ArcMC management) that depend on the Web Service will fail.

  - Any other Transformation Hub (Transform Stream Process, Schema Registry, Kafka Manager) that was running on the master will get scheduled by Kubernetes on other worker nodes depending on system resources available.
  - If the master node was labeled for Kafka and/or ZooKeeper deployment, then those instances will fail but the cluster will still work with the rest of the instances on worker nodes.
- The NFS server, which runs on the master node, will become unavailable.
  - Kafka and ZooKeeper do not depend on NFS storage and use local Kubernetes worker node storage. They will be available for event processing with some limitation.
  - The beta feature Connector in TB (CTB) will be affected, since it depends on NFS storage, which is configured on master server.
- DNS service (`kube-dns`) runs on the master server will become unavailable.
  - Worker nodes will lose the ability to resolve host names, except for those that had already been resolved, and which may be cached for some period.
- Any of the Transformation Hub service instances running on the worker node which is down and these instance are not tied to a worker node (such as Transform Stream Process, Routing Stream Process, Schema Registry, or Kafka Manager) will be scheduled by Kubernetes on other worker nodes, depending on system resources available on other worker nodes.
- Depending on system resources on other worker nodes, Transformation Hub service instances that are labeled for Kafka and ZooKeeper will be automatically scheduled by Kubernetes on other worker nodes (if there are additional worker nodes already labeled for Zookeeper and kafka).
  - Likewise, the c2av-processor may cease if the worker node containing the `eb-c2av-processor` goes down and system resources prevent Kubernetes from automatically rescheduling processing on another worker node.
  - If automatic re-scheduling of service instances does not occur for the Zookeeper, Kafka, or `eb-c2av-processor` (that is, the node is not recoverable), run the following manual command from the master node to delete all service instances from the failed node and force Kubernetes to move the services to other nodes:

```
# kubectl delete node <Failed_Node_IP>
```

Note: There must be another node available in the cluster, with the `zookeeper` and `kafka` labels, for the service instances to be migrated from the failed node.

Tuning Transformation Hub Performance

The following can help improve the performance of Transformation Hub.

Increasing Stream Processor EPS

You can increase Stream Processor EPS by adding more stream processor instances using the ArcSight installer configuration UI. When you change this value, you do not need to redeploy Transformation Hub.

Note: This change will increase the number of pods. You will see this difference when you run the `kubectl get pods --all-namespaces` command.

Increasing Kafka Retention Size or Time

You can change the value of retention size or time in any topic using Transformation Hub Manager after deploying Transformation Hub containers. You can change the value while events are flowing through the topic.

To change the default values before you deploy, change the values in the `arcsight-installer.properties` file. The changes are applied immediately.

Adding a New Worker Node

To add a new worker node, label the new node (delete or overwrite the existing label with a different label). Remove the label from the old node. Kubernetes should start Kafka on the new node. Then, reassign partitions on the new node. Data copying will take some time to complete

Verifying the Health of the Transformation Hub Cluster

Verify the health of each container

Run the following command to list pods and their status:

```
#kubectl get pods --all-namespaces -o wide to list pods and their status.
```

View Kubernetes logs for each container

Run the following command to list pods and their status:

```
# kubectl logs -n eventbroker1 [WEB SERVICE POD ID/NAME] -c atlas-web-service
```

Verify data flows through the system

Check any of the following:

- In ArcMC, review the EPS graph and verify whether events are flowing through the stream processor (routing and transforming).
- All topics: Check the offset for each topic in Transformation Hub Manager and verify whether the offset value increasing.

Verify that Web Service APIs are healthy:

- Check logs of the web service container (see command above).
- Make sure the port is bound by running the following command:

```
# netstat -lntp | grep 38080
```

- Check the Kakfa Scheduler status by running the following command:

```
# watch ./root/install-vertica/kafka_scheduler  status
```

- Check whether the offset is increasing in the status output. If not, then there might no data in the Avro topic, or if Avro contains data there may be a problem.
- Verify the topic partition count and distribution.
- Check that the configured partition count matches its expected value.
- Check the partition count or replication factor for the topic using Transformation Hub Manager.

Self-Healing for Unparsed Events

If the **Generate Unparsed Events** feature is on, and there is a high traffic of events, the destinations might receive less valuable information (unparsed events).

Users can now set a limit for unparsed events. When the number of unparsed events reaches the limit, the **Self-Healing** feature disables the unparsed event generation and as soon as the event queue normalizes, it is re-enabled and the destinations start receiving unparsed events again.

If the **Self-Healing** feature is active, an internal event is sent with the status of the **Generate Unparsed Events** feature.

New Properties

- `unparsed.events.self.healing.enabled` — The default value is **False**. Set to **True** to turn on the functionality.
- `unparsed.events.self.healing.threshold.limit`— The default value is **60%**. The

functionality is enabled when the limit is reached.

Note: The limit refers to the percentage of unparsed events in a given time.

To enable Self-Healing:

From the `agent.properties` file, ensure the `unparsed.events.self.healing.enabled` flag in is set to **True**.

SmartConnector Commands Queue

Commands Queue prevents from executing commands that might conflict if executed at the same time. Conflicting commands are sent to a queue and executed later.

The ESM Console can track commands sent to an agent. When the console finds a conflict, it does not send the command. However, if another console sends the command to the agent at the same time, there might be a conflict among each other.

Some command groups might present an error when executed from different ESM Consoles in parallel processes, or, if the same command is executed more than once, at the same time.

Commands received by a connector are now placed in a queue and categorized, according to their priority.

Command Type	Description
"Get" Commands	Always safe to execute. Skip the queue. Executed at the moment they arrive.
Any other commands	Low conflict probability, unless another process is running the same command.
Event Flow Commands	Change the connector status (start, stop, pause).
Upgrade Commands	Change the connector version.
Connector Process Commands	Turn off the connector (terminate, restart). Its priority can be changed to always. Choose these commands above the others. To change the priority: 1. Go to the `agent.properties` file. 2. Add the following property to the file: `commands.queue.connector.process.high.priority=true` 3. Ensure the property status is set to **True**.

TLS Warning when Running a SmartConnector

The following warning can be displayed without affecting the performance of the connector:

```
[WARNING: The protocol [TLSv1.1] was added to the list of protocols on the
SSLHostConfig named [_default_]. Check if a +/- prefix is missing.]
```

If you want to remove the warning:

Modify the `agent.default.properties` file as shown below:

```
remote.management.ssl.enabled.protocols=TLSv1.2,+TLSv1.1,+TLSv1
```

```
remote.management.ssl.fips.enabled.protocols=TLSv1.2,+TLSv1.1,+TLSv1
```

Handshake Error when Configuring Connector 7.15 or older with ESM 7.6

When configuring Connector version 7.15 or older with ESM 7.6, the installation program returns a handshake error.

Workaround:

1. In the `C:\arcsight\Connectors\current\config\agent\agent.defaults` properties file, go to the **# The following cipher suites are supported:** > **# In FIPS mode** section.
2. Add the following FIPS cipher to an existing list of ciphers:

   ```
   ssl.fips.cipher.suites=TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
   ```

A Non-administrator User Unable to Run Connectors on Windows and the Log File has Permission Error

Issue: If any user other than the administrator tries to run any of the Windows connectors, the connector does not run and the log file shows the following error:

```
[FATAL][default.com.arcsight.agent.am.e][init] Could not initialize the
Obfuscation key manager
[FATAL][default.com.arcsight.agent.am.e][init]
com.arcsight.common.config.n: An error occured in configuration. Unable to
load properties from file ‘<install
path>\current\user\agent\keys\obfuscationkey'.
Error was: ‘<install path>\current\user\agent\keys\obfuscationkey (Access is
denied)'
```

Workaround: This issue occurs because only the administrators are authorized to access the `<install path>\current\user\agent\agent.properties` and `<install path>current\user\agent\keys\obfuscationkey` files in the SmartConnector 7.15.0 or later. However, if you add a user or a users group to access these files, permission entries of these newly added users (except the owner of the files) will be automatically removed after you restart the connector.

For a non-administrator user to run this connector, change the **ownership** of the **agent.properties** and **obfuscationkey** files to a corresponding user with the **Full control** permission. If there are more than one users who need permission to run the connector, add these users in the same group so that the **ownership** of the **agent.properties** and **obfuscationkey** files can be assigned to this group.

For information about taking ownership and full control of files, refer to the Microsoft documentation.

Frequently Asked Questions

The section contains a list of frequently asked questions.

Note: This section is periodically updated.

My machine is in a different location than 'en_US' and my connectors display parser errors when parsing timestamp fields

The connector assumes a default locale of 'en_US'. If your machine is running in a different locale, your connector might display parsing errors when parsing timestamps. Try changing the parser locale by adding the property 'agent.parser.locale.name=<locale of your machine>' into the user/agent/agent.properties file, and restart your connector.

For example, China and France would have the following locales:

```
agent.parser.locale.name=zh_CN
agent.parser.locale.name=fr_FR
```

To use the default locale for the connector machine, you can leave the locale blank. For example:

```
agent.parser.locale.name=
```

What if my device is not one of the listed Connectors?

- ArcSight offers an optional feature called the FlexConnector Development Kit (SDK), which can assist you in creating a custom connector for your device.
- ArcSight can create a custom connector; contact customer support for more information.

My device is on the list of supported products; why doesn't it appear in the Connector Configuration Wizard?

Connectors are available for installation based upon the operating system you are using. If your device is not listed, either it is not supported by the operating system on which you are attempting to install, or your device is served by a Syslog server and is, therefore, a Syslog sub-connector.

To install a Syslog connector, select **Syslog Daemon**, **Syslog Pipe**, or **Syslog File** during the installation process.

Why isn't the SmartConnector reporting all events?

Check that event filtering and aggregation setup is appropriate for your needs.

Why are some event fields not showing up in the Console?

Check that the two separate turbo modes for the connector and the Manager are compatible for the specific connector resource. If the Manager is set for a faster turbo mode than the connector, some event details will be lost.

Why isn't the SmartConnector reporting events?

Check the Connector log for errors. Also, if the Connector cannot communicate with the Manager, it caches events until its cache is full. A full cache can result in the permanent loss of events.

How can I get my database SmartConnector to start reading events from the beginning?

- If it is a FlexConnector for Time-Based DB, set the following parameter in the `agent.properties` file:

  `agents[0].startatdate=01/01/1970 00:00:00`
- If it is an FlexConnector for ID-Based DB, set the following parameter in the `agent.properties` file:

  `agents[0].startatid=0`

When events are cached and the connection to the Manager is re-established, which events are sent?

Events are sent with a 70% live and 30% cached events ratio. If live events are not arriving quickly, the percentage of cached events can be higher. This can reach 100% if there are no live events.

Also, if the settings dictate that certain event severities are not sent at the time connection is restored, those events are never sent. This is true even if they were originally generated (and cached) at a time when they would ordinarily go out.

Why does the status report the size of the cache as smaller than it should be? For example, I know that a few events have been received by the SmartConnector since the Manager went down, yet the report marks events as zero

Some of the events are in other places in the system, such as the HTTP transport queue. Shut down the connectors and look at the cache size in the `.size.dflt` file to confirm that the events are still there.

Why does the estimated cache size never change in some connectors? Why is the estimated cache size negative in others?

The estimated cache size is derived from a size file that gets read at startup and written at shutdown. If the connectors could not write the size at shutdown (for example, due to an ungraceful shutdown, disk problem, or similar problem) the number could be incorrect. Newer versions will attempt to rebuild this cache size if they find it to be incorrect, but older builds do not.

To rebuild the cache file:

1. Stop the connector.
2. Delete the size file (a file with extension `.size.dflt`) under `current\user\agent\agentdata`.
3. Re-start the connector.

The connector detects that there is no size file and re-builds the cache size by reading all the cache files.

Can the SmartConnector cache reside somewhere other than user/agent/agentdata?

You can change the folder to contain the connector cache by adding the following property in the agent.properties file:

agentcache.base.folder=<relative-folder-path>

where, <relative-folder-path> is the path of the folder relative to $ARCSIGHT_HOME.

Why is my end time always set to a later date and time?

The Manager performs auto time correction for older events. If the end time is older than your retention period, it is set automatically to that lower bound. A warning is displayed and an internal event with the same message is sent to you.

Do our Syslog connectors support forwarded messages from KIWI or AIX?

Yes.

The property related to KIWI is

syslog.kiwi.forwarded.prefix=KiwiSyslog Original Address

Kiwi adds a prefix with the original address. For example, the message:

Jan 01 10:00:00 myhostname SSH connection open to 1.1.1.1

is converted to

Jan 01 10:00:00 myhostname KiwiSyslog Original Address myoriginalhost: SSH connection open to 1.1.1.1

The Connector strips out the prefix and uses myoriginalhost as the Device Host Name.

The property related to AIX is

syslog.aix.forwarded.prefixes=Message forwarded from,Forwarded from

Similar actions are performed for messages forwarded using AIX.

What does the T mean in the periodic SmartConnector status lines?

"T" is shorthand for "throughput (SLC)." The following lines are in the `agent.defaults.properties` file:

```
status.watermark.stdoutkeys=AgentName,Events
Processed,Events/Sec(SLC),Estimated Cache
Size,status,throughout(SLC),hbstatus,sent
status.watermark.stdoutkeys.alias=N,Evts,Eps,C,ET,T,HT,S
```

The SLC stands for Since Last Check, which means "in the last minute," assuming `status.watermark.sleeptime=60` has not been overridden.

What do Evts and Eps refer to?

Evts is an acronym for Events Processed and **Eps** is an acronym for Events/Sec(SLC).

Does a file reader SmartConnector reading files over a network share display errors when the network share is disconnected? How can I recognize which error message refers to which file in `agent.log` and `agent.out.wrapper.log`?

If the network share is a Linux/UNIX NFS mount or a Windows network mapped drive, the file reader connector displays errors in the agent log.

If files are being read using a Windows UNC path that does not require network mapping, the file reader connector cannot detect a network connection loss.

Error messages related to file access contain the file name, but error messages related to log line parsing does not.

Are log files accessed sequentially or in parallel?

This depends upon the connector you are using. Some log file connectors process files sequentially and others process log files in parallel.

After reading a log file, can a SmartConnector move them using NFS?

Yes. Folder Follower connectors can rename or move the files using NFS, if the folders containing the log files give the correct permissions for the connector.

My SmartConnector must read log files from a remote machine through a network share. How can I do this?

To establish a network share to a remote machine, you can use network mapping on Windows platforms, and NFS or Samba mounting on Linux/UNIX platforms.

If you are running the connector as a Windows service, access privileges to the network share are required. To access the user name and password panel:

1. From the **Start** menu, select **Control Panel**.
2. Double-click **Administrative Tools**.
3. Double-click **Services**.
4. Right-click the name of the appropriate connector and select **Properties**.
5. Click the **Log on** tab and enter the user name and password for the user with access permissions to the file share. Specify the file path using UNC notation, not as a network mapped drive.

Is there any limitation on performance relating to EPS?

These limitations are subjective and depend upon system resources, number of devices, number of events, and so on.

How many log files can a SmartConnector access at one time?

The connector can access as many log files as it is configured with. The folders are processed in parallel.

What is the recommended maximum number of connectors per Manager?

There is no hard and fast maximum. The Manager has a restriction of 64 concurrent Connector threads by default. The more threads you add, the more it affects performance, because there is more thread context-switching overhead. The recommendation is to stay lower than the triple-digit range.

When configuring the connector to run as a service (for Windows) or daemon (for Unix), you may encounter the following error message: An issue has been encountered configuring the connector to run as a service. Check agent.log (Service Installation) for details

There may be different reasons for you to get this message when you cannot configure the connector to run as a service or daemon. It may be that you installed a second connector on Windows or Unix with the same name and type, such as when using the default options. More information is included in the *agent.log*, including the specifics for <Service Installation>. For example: <Service Installation> - SE:wrapperm | Unable to install the ArcSight Syslog NG Daemon service - The specified service already exists. (0x431).

You can fix this issue by manually deleting the `agent.wrapper.conf` file from the second or additional connectors. The file is present in the `$ARCSIGHT_HOME/current/user/agent` folder.

When configuring multiple connectors, use a different name and type to avoid duplication.

Which is the default cache size limit?

SmartConnectors use a compressed disk cache to hold large volumes of events when the destination (such as ArcSight Manager) is down or when the SmartConnector receives bursts of events. This parameter specifies the disk space to use. The default is 1 GB which, depending on the connector, can hold about 15 million events (though this can vary dramatically depending on the event type), but it also can go down to 200 MB. When this disk space is full, the SmartConnector drops the oldest events to free up disk cache space. The default is value **1 GB**.

How are the cached DNS entries managed? If there is a size limit, how would it be handled? Will it rotate when the limit exceeds?

If the Name Resolution Host Name Only parameter is set to No, then the source, destination, and device host name and DNS domain event fields are looked at. If in any of the three cases the DNS domain event field is empty and the corresponding host name event field contains a host name that is not an IP address (IPv4 or IPv6) and does have a dot in it, then the host name event field is split at the first dot, with the latter part being moved to the corresponding DNS domain event field.

If, on the other hand, the corresponding DNS domain and host name event fields are both set for the source, destination, or device, and the host name event field does not contain any dots, then they are combined for purposes of any name resolution that may be done later.

Next, if the Name Resolution Host Name Only parameter is set to No, and the agent host name event field is set (which is normally done for all Connectors based on a lookup done at startup time), contains a dot, and is not an IP address, then it is split up, with the part after the first dot being moved to the agent DNS domain event field.

Lastly, if the Name Resolution Domain From E-mail parameter is set to Yes, then the source and destination DNS domain event fields are looked at. If either is empty, the corresponding user name event field is not empty and contains an "@" character, and the corresponding host name event field is empty, then the DNS domain event field is set to the part of the user name event field after the "@" character.

On the other hand, the default limit is 50000 entries for each of the two caches (names => IPs, and IPs => names). An expiration daemon runs periodically (normally once per minute) to check for stale cache entries, as defined by the shortest TTL set for all destinations. Older entries are simply queued for re-resolution. If the queues are at all backed up, these refresh requests may take some time, which is why entries that are up to twice the TTL are not considered stale when Wait For Resolution is disabled. Note that previously, arbitrarily old entries would continue to be used, so the behavior in the case of severely backed up queues has changed.

Under what circumstances can an entry be removed from the cache?

Cache entries are removed if 1) the size limit is reached or 2) the name or IP address is explicitly configured to not be looked up but it was previously in the cache. Additionally, an entry can be removed after DNS lookup fails if it has been previously configured by removing Unresolvable Names or IPs from cache.

How should an entry be manually removed from the cache?

It is not possible to manually remove a single entry from the cache. It is only possible to flush the cache files while the connector is stopped.

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 computer, click the link above and an email window opens with the following information in the subject line:

Feedback on SmartConnector Installation and User Guide (SmartConnectors CE 24.4)

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 MFI-Documentation-Feedback@opentext.com.

We appreciate your feedback!