opentext™

ArcSight SmartConnectors

Software Version: CE 24.3

Overview of SmartConnectors

Document Release Date: July 2024
Software Release Date: July 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. 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 11.

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 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

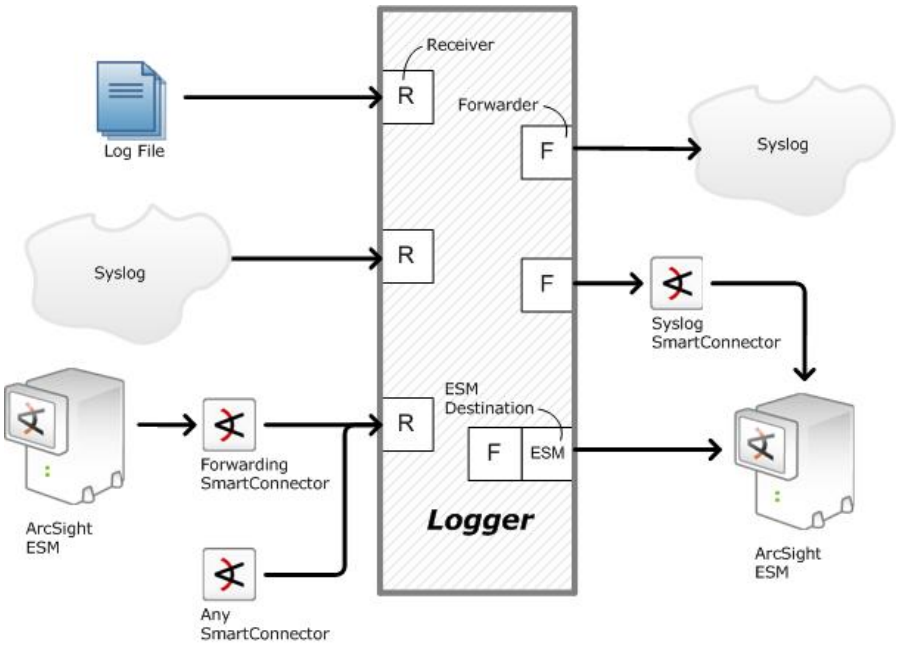
Sending Events from Logger to a Manager

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.

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.

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.

This 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.

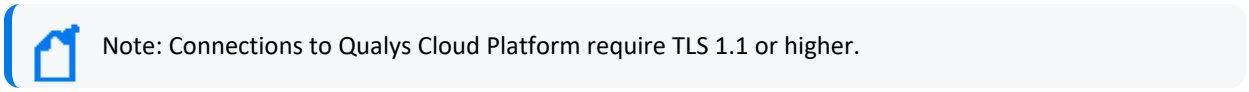
A Connector can also be configured to send events to a CSV file and an ESM Manager at the same time.

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.



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

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 Overview of SmartConnectors (SmartConnectors CE 24.3)

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!