



Common Event Format Configuration Guide

Layer 7 Technologies

SecureSpan Gateway / CloudSpan Gateway

Date: Friday, October 07, 2011



CEF Connector Configuration Guide

This document is provided for informational purposes only, and the information herein is subject to change without notice. Please report any errors herein to ArcSight, LLC. ArcSight does not provide any warranties covering this information and specifically disclaims any liability in connection with this document.

Certified CEF Compatible:

The event format complies with the requirements of the ArcSight Common Event Format. The ArcSight CEF connector will be able to process the events correctly and the events will be available for use within ArcSight products.

Certified CEF Compliant:

The event format complies with the requirements of the ArcSight Common Event Format. The ArcSight CEF connector will be able to process the events correctly and the events will be available for use within ArcSight products. In addition, the event content has been deemed to be in accordance with standard SmartConnector requirements. The events will be sufficiently categorized to be used in correlation rules, reports and dashboards as a proof-of-concept (POC) of the joint solution

Layer 7 SecureSpan Gateway / Layer 7 CloudSpan Gateway

September 9, 2011

Revision History

Date	Description
[09/09/2011]	First edition of this Configuration Guide.

Layer 7 Gateway Configuration Guide

This guide provides information for configuring the Layer 7 SecureSpan or CloudSpan Gateway for syslog event collection. This Connector is supported on all supported platforms. Device versions starting at 6.1 are supported.

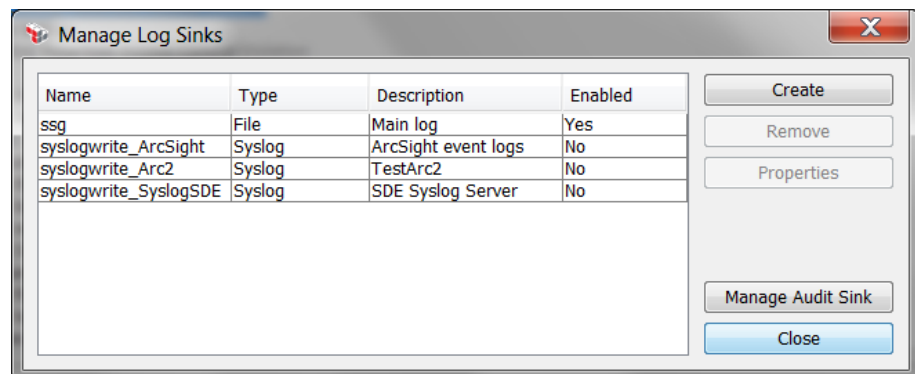
Overview

Layer 7 Gateways provide a central Policy Enforcement Point between service providers and consumers, allowing organizations to secure, monitor, and adapt their enterprise applications exposed as web services, APIs, or other standards-based interfaces. Deployed as hardware appliances, virtual appliances, cloud-resident appliances or installable software, Layer 7 Gateways offer credential-based access control, data and transport security, message format and protocol mediation, threat protection, routing, service adaptation and end-to-end API governance.

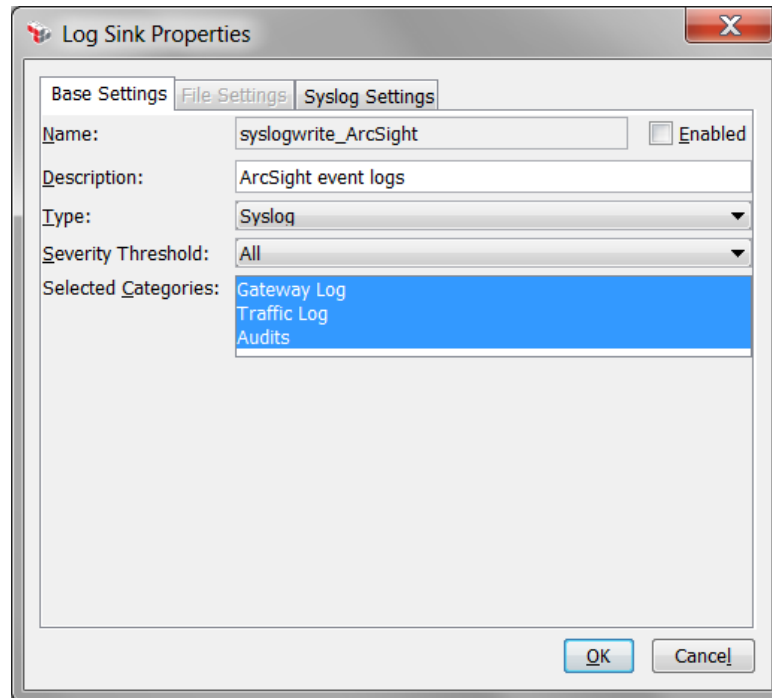
Configuration

Layer 7 Gateways connect to ArcSight using the CEF specification delivered via syslog-formatted event messages. This functionality can be added to any Gateway version 6.1 and above by requesting the “Log Message to Syslog” Assertion from Layer 7 support. The assertion should be installed using the *Layer 7 Custom Assertion Installation Manual*.

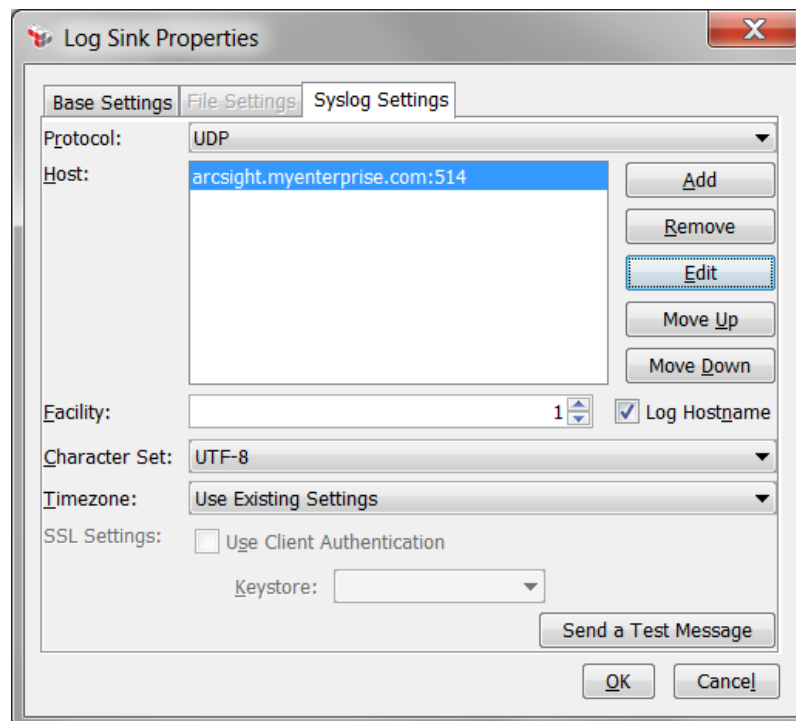
Initial connectivity to the CEF Connector is configured via the “Manage Audit/Log Sinks” task within the Layer 7 Policy Manager.



Click ‘Create’ to configure a new Log Sink for ArcSight. Name the new Log Sink with the naming convention “syslogwrite_ SinkName” and provide a description. Select ‘Syslog’ as the Type and ‘All’ as the Severity Threshold. Select all three category types, and do not click the ‘Enabled’ checkbox.



When the Type has been set to 'Syslog,' the Syslog Settings tab will become available. Set the Protocol to 'UDP' and click 'Add' to provide the ArcSight host and port information. Leave the remaining fields at their default values. Click 'OK' to complete the ArcSight connector configuration.

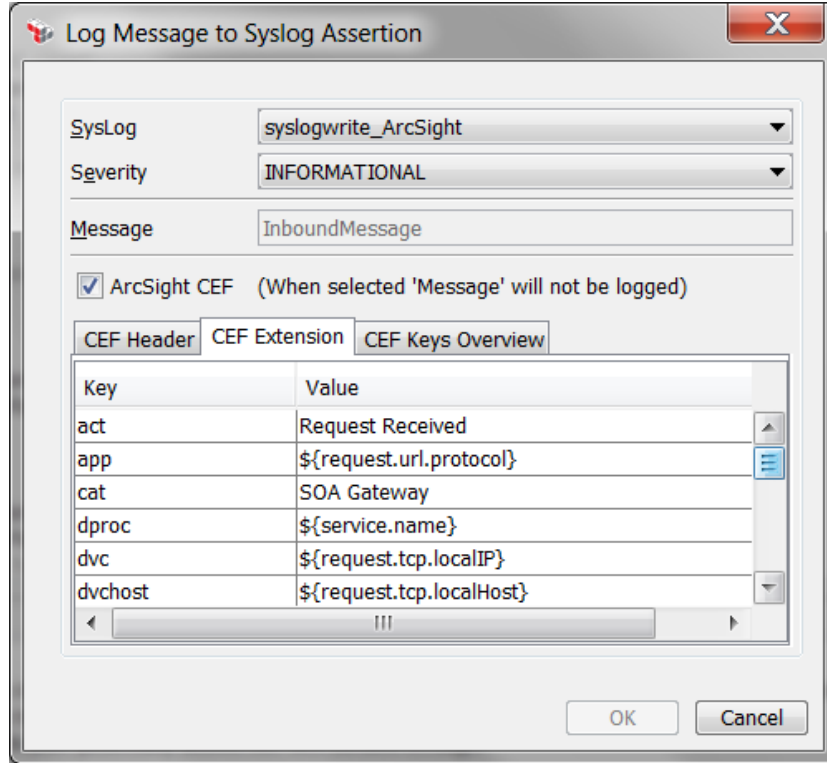


If the Log Message to Syslog Assertion has been installed correctly, it will appear in the Assertion palette under Logging, Auditing, and Alerts. This assertion can be used in any policy to log relevant events relating to the application being managed.

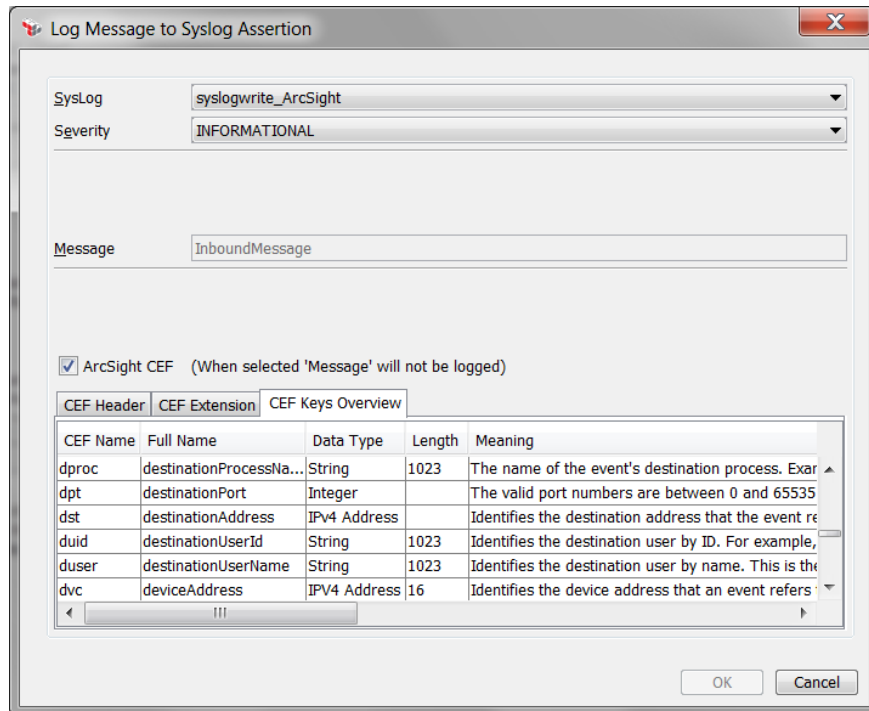
Upon adding the assertion to the Policy Development window, the configuration interface will appear. Select the Syslog connection previously created in the Manage Audit/Log Sinks task and click the 'ArcSight CEF' checkbox to make the assertion produce CEF-formatted output. Most of the CEF header fields are predetermined, as are the host and timestamp that are sent with any syslog-formatted event. Select a Signature-ID, Name, and Severity. While these fields are customizable, we suggest defining a convention for which events will be recorded and at what severity level.

The screenshot shows the 'Log Message to Syslog Assertion' configuration window. The 'SysLog' dropdown is set to 'syslogwrite_ArcSight' and 'Severity' is 'INFORMATIONAL'. The 'Message' field is empty. The 'ArcSight CEF' checkbox is checked. The 'CEF Header' tab is active, showing 'CEF Version' as 0, 'Device Vendor' as Layer7 Technologies Inc., 'Device Product' as SecureSpan Gateway, and 'Device Version' as 6.1-2. The 'Signature-ID|Name' field contains '100 | Message Received' and 'CEF Severity' is set to 2. 'OK' and 'Cancel' buttons are at the bottom right.

Select the CEF Extension tab to define the additional fields and values to be submitted to the ArcSight server for this event. Again, the value chosen for these fields are completely customizable, but suggestions are provided in the Device Event Mapping section at the end of this document. Data can either be hard-coded or determined dynamically through the use of predefined or user-defined context variables. This allows insertion of transaction metadata that will update automatically with each request.



For more information on the Keys available for use in the CEF Extension tab, select the CEF Keys Overview tab. This provides an alphabetical list of the keys supported by the specification and a brief description of their meanings.



Usage of this CEF-specific assertion will vary greatly according to the policy being constructed and the event logging requirements defined by the ArcSight administrator. Some options for deployment are:

- At the beginning of the policy to keep a record of every request
- At the end of the policy to summarize the request-response conversation
- In conjunction with threat protection assertions to log incoming threats
- In conjunction with access control assertions to record user identities
- In conjunction with credential mapping to record both internal and external credential tokens
- In conjunction with routing assertions to track the flow of inbound traffic
- Some combination of any/all of the above, using `$(requestId)` to correlate events relating to a single transaction

Example

The following ArcSight channel displays events generated by a Layer 7 Gateway. Inbound and outbound events are provided for successful transactions, access control and routing events are included for informational purposes, and validations of threat protection or authentication policies are

Displaying: All

Field Set: Extended (18 Pages Total)

	Manager Receipt Time	End Time	Name	Device Vendor	Device Product	Device Severity	Device Action	Device Event Class ID	Device Host Name	Device Address	Attacker Host Name	Attacker Address
<input type="checkbox"/>	9/10/2011 4:01:57 AM PDT	9/10/2011 4:01:46 AM PDT	Request Metrics Inbound	Layer7 Technologies Inc.	SecureSpan Gateway	2	Request Received	100		192.168.206.196		192.168.206.1
<input type="checkbox"/>	9/10/2011 4:01:57 AM PDT	9/10/2011 4:01:46 AM PDT	Access Control	Layer7 Technologies Inc.	SecureSpan Gateway	4	Authentication	200	99-204-150-151.pools.spcsdns.net	99.204.150.151		
<input type="checkbox"/>	9/10/2011 4:01:57 AM PDT	9/10/2011 4:01:46 AM PDT	Theat Protection Violations	Layer7 Technologies Inc.	SecureSpan Gateway	8	Request processing aborted	520	99-204-150-151.pools.spcsdns.net	99.204.150.151		
<input type="checkbox"/>	9/10/2011 4:01:57 AM PDT	9/10/2011 4:01:46 AM PDT	Request Metrics Inbound	Layer7 Technologies Inc.	SecureSpan Gateway	2	Request Received	100		192.168.206.196		192.168.206.1
<input type="checkbox"/>	9/10/2011 4:01:57 AM PDT	9/10/2011 4:01:46 AM PDT	Access Control	Layer7 Technologies Inc.	SecureSpan Gateway	4	Authentication	200	99-204-150-151.pools.spcsdns.net	99.204.150.151		
<input type="checkbox"/>	9/10/2011 4:01:57 AM PDT	9/10/2011 4:01:46 AM PDT	Message Validation Violation	Layer7 Technologies Inc.	SecureSpan Gateway	8	Request processing aborted	540	99-204-150-151.pools.spcsdns.net	99.204.150.151		
<input type="checkbox"/>	9/10/2011 4:01:57 AM PDT	9/10/2011 4:01:48 AM PDT	Request Metrics Inbound	Layer7 Technologies Inc.	SecureSpan Gateway	2	Request Received	100		192.168.206.196		192.168.206.1
<input type="checkbox"/>	9/10/2011 4:01:57 AM PDT	9/10/2011 4:01:48 AM PDT	Access Control	Layer7 Technologies Inc.	SecureSpan Gateway	4	Authentication	200	99-204-150-151.pools.spcsdns.net	99.204.150.151		
<input type="checkbox"/>	9/10/2011 4:01:57 AM PDT	9/10/2011 4:01:48 AM PDT	Message Routing	Layer7 Technologies Inc.	SecureSpan Gateway	2	Delivered	300	99-204-150-151.pools.spcsdns.net	99.204.150.151		
<input type="checkbox"/>	9/10/2011 4:01:57 AM PDT	9/10/2011 4:01:48 AM PDT	Request Metrics Outbound	Layer7 Technologies Inc.	SecureSpan Gateway	2	Response Sent	150	99-204-150-151.pools.spcsdns.net	99.204.150.151		

Events

Events and Signature-IDs are customizable by the Layer 7 administrator, but one sample event list could be as follows. Additional message codes and values are available in the *Layer 7 Policy Manager User Manual*, in the *Audit Message Codes Appendix*.

- 100 – Inbound Request
- 150 – Outbound Response
- 200 – Access Control Operation
- 300 – Message Routing Operation
- 500 – Access Control Violation
- 520 – Threat Protection Violation
- 540 – Message Validation Violation
- 560 – Message Routing Violation

Device Event Mapping to ArcSight Data Fields

Information contained within Layer 7 event definitions is sent to the ArcSight SmartConnector, and then mapped to an ArcSight data field.

The following table lists some suggested mappings from ArcSight data fields to the Layer 7 values and/or context variables.

Layer 7 Gateway Connector Field Mappings

Vendor-Specific Event Definition	ArcSight Event Data Field
<code>\${request.url.protocol}</code>	app
SOA Gateway	cat
<code>\${requestId}</code>	externalId
<code>\${request.tcp.localIP}</code>	dvc
<code>\${request.tcp.localHost}</code>	dvchost
<code>\${request.size}</code>	in
<code>\${request.url.protocol}</code>	proto
<code>\${request.url}</code>	request
<code>\${gateway.time}</code>	rt
<code>\${service.name}</code>	dproc
<code>\${request.tcp.remotePort}</code>	spt
<code>\${request.tcp.remoteHost}</code>	src
<code>\${request.http.method}</code>	requestMethod
<code>\${httpRouting.url.host}</code>	dst
<code>\${gateway.time}</code>	end
<code>\${response.size}</code>	out
<code>\${authenticatedUser.id}</code>	duid
<code>\${request.authenticateduser}</code>	duser
<code>\${request.mainpart}</code>	msg
<i>errorSpecificValue</i>	reason