Common Event Expression

Common Log Transport (CLT) Requirements Specification

Version 0.6

The CEE Editorial Board
October 2011

This page intentionally left blank.

This document is provided by the copyright holders under the following license.

LICENSE

Disclaimers

THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS. COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.

This page intentionally left blank.

Abstract

MITRE, in collaboration with industry and government, offers the CEE Common Log Transport (CLT) requirements to define the mandatory and preferred capabilities for a log transport protocol. Such protocols enable CEE Log Syntax (CLS) encoded event records to be shared between parties in a universal, machine-readable manner. The intent of CLT is to provide guidance for vendors and end users regarding how event records should be reliably and securely shared.

KEYWORDS: CEE, CLT Protocol CLT, CLTP, Logs, Event Logs, Audit Logs, Log Protocol, Audit Protocol

This page intentionally left blank.

Table of Contents

Table of Figures

1 Introduction

The CEE Log Transport (CLT) provides the technical support necessary for a secure, interoperable, and reliable log infrastructure. A log infrastructure requires more than just standardized event records, support is needed for international string encodings, secure logging services, standardized event interfaces, and secure, verifiable log trails.

The CLT defines a listing of requirements that a **CLT Protocol** must meet. For example, a CLT Protocol must be able to transmit a Common Log Syntax (CLS) Encoded CEE Event. More advanced CLT Protocols may provide things like encryption and full acknowledgments. A CLT Protocol may be able to specify or transmit Common Event Log Recommendation (CELR) Profiles and additional event-related information, such as packet captures or file data.

CLT also defines transport mappings. A **CLT Mapping** defines a standardized way for CEE Events to be transmitted over a certain CLT Protocol. One use for a CLT Mapping is to define how to send CEE Events over the RFC5425 TLS Syslog protocol [1]. This Mapping would define that the CEE Event must be encoded using an RFC5424 Syslog-compatible [2] CLS Encoding and placed at a certain point in the Syslog message. The CLT Mapping may need to define additional indicators, such as flags to indicate that the data is an encoded CEE Event and the character encoding used (e.g., UTF-8).

The CLT provides the features necessary to support the end-to-end audit process by extending the event record representation to include the essential confidentiality, integrity, and availability of audit services.

MITRE coordinates the CLT Protocol as part of the CEE Architecture, which is one piece of MITRE's *Making Security Measurable* initiative (http://measurablesecurity.mitre.org).

1.1 Background

The purpose of the CLT Protocol component of the CEE Architecture is to establish a heterogeneous environment by which logs can be shared. The intent of this CLT specification is to provide a list of requirements to allow vendors and end users to build CLT-compliant protocols to be used to transport CEE event records.

The log recommendations are provided in the form of machine-readable "profiles." A profile can define Dictionary fields and field types, Event Taxonomy tags and tag types, along with the events for a specific product, function, or capability [3]. For example, operating systems, applications, and firewalls utilize user accounts for authentication. A CELR authentication profile is available to use for all devices utilizing an authentication mechanism. Using the authentication profile would ensure the correct events are captured and the events contained the recommended level of detail.

Profiles are critical to understanding what is occurring on a device because a profile identifies event fields of importance associated with each event. For example, it is crucial to include the username associated with an attempted user login as this information allows analysts to distinguish between someone trying to find valid usernames versus someone who might have forgotten their password.

Each CELR profile is developed by subject-matter experts and validated against related best practices, including requirements documents, information assurance guidance, forensics guidance, and inputs from the CEE Community.

1.2 Scope

This document is an introduction to CLT and details requirements for CLT-compliant log protocols. CLT was defined with the assistance of and input from a community of vendors, researchers, end users, the CEE Editorial Board, and The MITRE Corporation.

1.3 Conformance

To be conformant with the CLT Protocol, applicable products and processes must provide support for a publicly available protocol that meets the requirements laid out by this document.

When using a syntax there should be options, depending on the environment and objectives, as to how information is transmitted. An administrator should be able to choose the best transport, regardless of whether it is an encoded binary syntax, name-value pairs, or an XML-based one. Common Log Transport (CLT) will be used to define the potential mediums through which CLS can be expressed and transmitted. Below are three possibilities for CLT that address issues of speed, ease-of-use, and expressiveness:

Speed — A binary log format (and corresponding syntax of fixed sized fields in a binary file) can express comprehensive information and is the fastest way to log and exchange data. When wanting to minimize size and network impact, compressed binary is the best option. However, binary syntaxes are not designed for human readability and require conversion libraries for encoding and decoding logs.

Ease-of-Use — Plaintext syntaxes include delimited and key-value pairs such as Comma-Separated Values (CSV) and ArcSight's Common Event Format (CEF), which humans and machines can more easily read and understand. With a fairly basic syntax, this format is very practical and would most likely have the best overall acceptance by event producers and consumers. Additionally, this type of syntax offers compatibility with a majority of transports. At the same time, this format is not as speed-efficient as a binary format.

Expressiveness — Syntaxes based on structures such as XML are comprehensive and capable of representing complex data structures, such as lists and nested object relationships. Similar to ease-of-use syntax options, an XML-based syntax would be a desirable option for some event producers and consumers. Some drawbacks include a limited choice of compatible transports as well as the extra space for storage and transmission and possible difficulties with human understanding of such logs. Since most event data is fairly straightforward, forcing it into an expressive syntax would be excessive.

An important feature of CLT Protocol is that many current log transport options can be adopted as a supplemental "standard." For example, Syslog over port UDP 514 is used by millions of UNIX-derived systems and thus can probably be considered as a standard log transport mechanism.

The CEE Community or other organizations may define additional validation processes, which impart further requirements or conformance tests.

1.4 Purpose

This document defines the requirements for CLT protocol to the CEE Community for validation and approval. Additionally, this document defines CLT Implementation Requirements. The CEE Community is vital to the success and adoption of CEE; therefore feedback and discussion is needed in order to produce an open, practical, and industry-accepted standard. Comments and recommendations should be submitted to the CEE Discussion List (`cee-discussion-list@lists.mitre.org`) or to the MITRE CEE Team (cee@mitre.org).

2 CLT Protocol Model

The CLT Protocol consists of a **session** of at least one event **channel** (Figure 1). Each channel consists of zero or more protocol **packets**. Each packet has a packet header and body (Figure 2). The body contains the event data within zero or more CLS encoded CEE event records.

The CLT Protocol session is a full-duplex pipe where the computer at either end is capable of initiating a message exchange. A CLT Protocol session carries one or more channels simultaneously. A channel is a stream of "typed" messages.

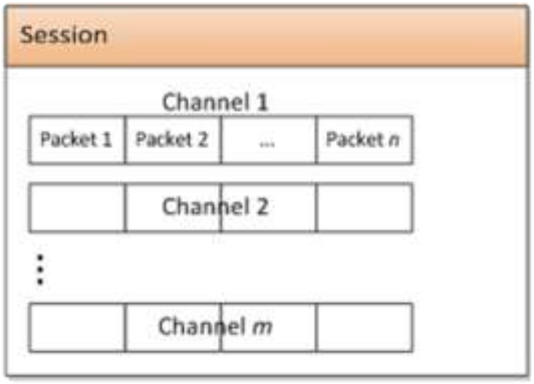

Figure 1: CLT Session Model

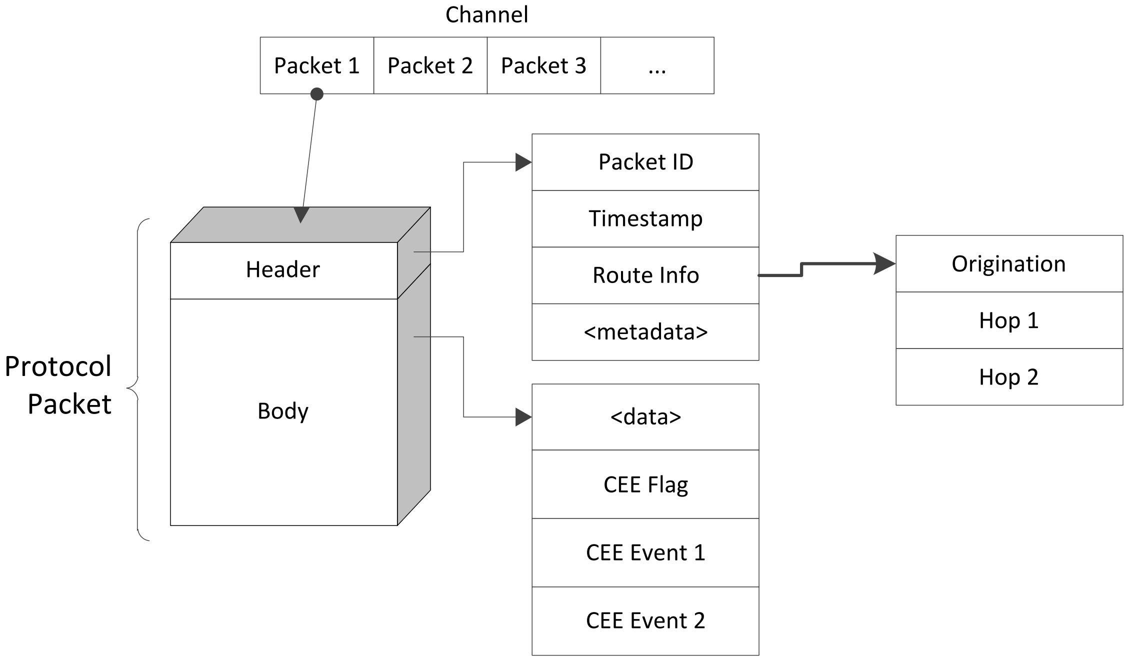

Figure 2: CLT Packet Model

The actual CEE events are sent from the Event Sender and are received by the Event Receiver. Each session communication may be sent directly from the Sender to the Receiver (Direct Transmission) or transmitted by way of one or more session-forwarding Relays (Multi-Hop Transmission). Figure 3, on the next page, shows as system view of the CTL Protocol.

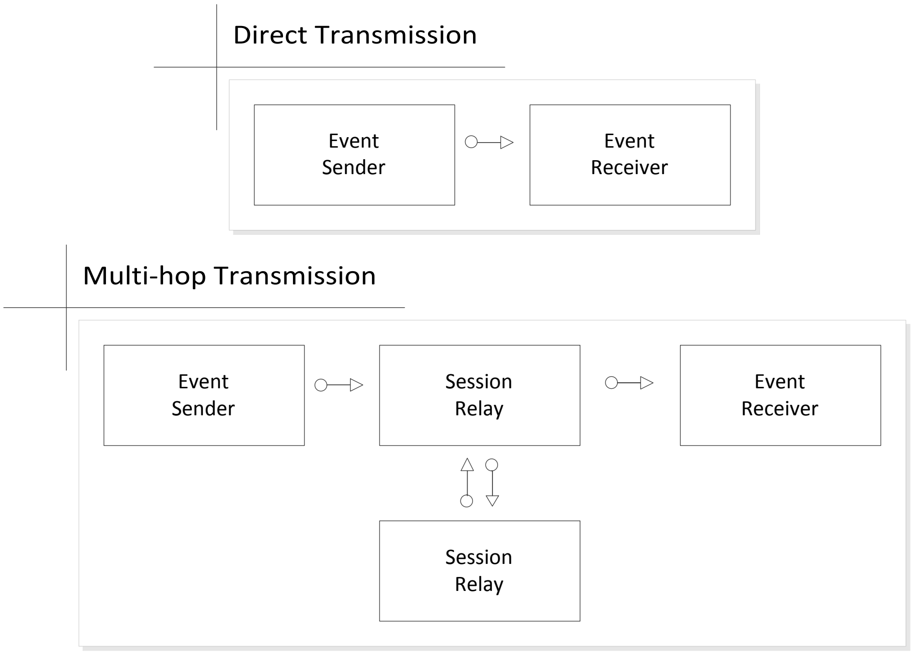

Figure 3: CLT Transmission Models

3 CLT Protocol Requirements

Many uses for events records require them to be transported from the originating system. CLT Protocols must meet some basic requirements in order to do this reliably and efficiently. This requirements document categorizes requirements in four groups based on conformance levels. Conformance Level 0 is the mandatory conformance level and includes basic capability. Further conformance levels describe optional capabilities that more advanced CLT protocols should support. Conformance Level 1 is core capability providing the minimum set of requirements for robustness. Conformance Level 2 contains a set of additional requirements that address logging in the presence of attackers. Conformance Level 3 contains an additional set of requirements that address local administrative attack scenarios. Conformance Level 3 is the most robust requirements set.

3.1 Conformance Level 0 – Core Requirements

3.1.1 Publish

The CLT Protocol shall be a published protocol specification with no licensing barriers to interoperability, no royalties and no approval process. The CLT Protocol shall utilize only those protocols and standards which are openly available.

3.1.2 Transport

The CLT Protocol shall be able to transport at least one form of CEE encoded event records within the body of the protocol packet.

3.1.3 Self-Identification

The CLT Protocol shall make evident by the protocol as to which data belongs to CEE event records and the CEE defined encoding used when a transmission is made consisting of CEE and non-CEE data.

3.1.3.1 Identification of CEE Events

The CLT Protocol shall have a mechanism that identifies the CEE Events within the protocol packet body.

3.1.3.2 Encoding Identifier

A CLT Protocol capable of handling more than one (1) data encoding method shall provide evidence as to the encoding being used.

3.1.4 Time Stamp

Each protocol packet shall have timestamp that indicates the date and time the packet was transmitted. A valid timestamp must indicate year, month, day, hour, minute, and second. Including sub-seconds is encouraged.

3.2 Conformance Level 1 – Basic Capabilities

One of the core capabilities is for the log transport to perform in limited bandwidth environments. Additional core capabilities include log transport tamper detection, Confidentiality, and Authenticity.

3.2.1 Event Record Delivery

The CLT Protocol shall preserve the integrity of the logical order of a channel's packets such that the Event Receiver will be able to reconstitute the original logical order.

3.2.2 Compression of Records

The CLT Protocol shall provide a method that allows the Event Sender to compress the packet body prior to transmission.

3.2.3 Missing Record Detection

The CLT Protocol shall be able to accurately and reliably detect missing transport packets.

3.2.4 Transmission Encryption

The CLT Protocol shall support transmission encryption using best practice encryption algorithms. One way to achieve this requirement is by using Transport Layer Security (TLS).

3.2.5 Confidentiality

The CLT Protocol shall maintain confidentiality of data, minimally within the packet body.

The CLT Protocol cryptography modules shall be capable of performing data encryption using data encryption best practices.

3.2.6 Message Identification

The CLT Protocol shall support message identifiers.

3.2.6.1 Packet Duplication

The CLT Protocol shall be able to identify when duplicate packets have been received.

3.2.6.2 Packet Acknowledgement

The CLT Protocol shall support at least one method of allowing the Event Receiver to acknowledge (ACK) that a packet has been received.

3.2.6.3 Packet Retransmission

The CLT Protocol shall be able to retransmit individual packets on request at least until the Event Receiver has acknowledged reception or until a nominal time has elapsed.

3.2.7 Packet Traversal Traceability

The CLT Protocol shall be capable of tracing and recording the path a packet traverses. The intent of this requirement is to allow packets to be traced through Session Relay devices such as NATs.

3.2.8 Tamper Detection

The CLT Protocol shall accurately and reliably detect any evidence of tampering or data corruption through the use of digital signatures or other anti-tamper mechanisms.

3.2.9 Authenticity

The CLT Protocol shall maintain authenticity of the data in transit.

3.2.9.1 Use of SASL, GSS-API, and Kerberos

The CLT Protocol shall support authenticity using Simple Authentication and Security Layer (SASL), Generic Security Services Application Program Interface (GSS-API), and Kerberos.

3.3 Conformance Level 2 – Securely Log in the Presence of Attackers

The core theme for the Conformance Level 2 transport requirements is the capability for CLT Protocol to securely log in the presence of attackers.

3.3.1 Full Integrity Acknowledgements

The CLT Protocol shall be capable of producing packet level acknowledgements that contains data through which the sender can verify the integrity of a received packet. This requirement is an extension of that of Requirement 3.2.6.2: Packet Acknowledgement.

3.3.2 Message Replay Protection

The CLT Protocol shall protect against message replay.

3.3.3 Event Integrity

The CLT Protocol shall accurately and reliably detect any repeated or unexpected message identifiers.

3.3.3.1 Chain of Modification

The CLT Protocol shall maintain the chain of modification of CEE Event data that is modified while in transit.

3.3.3.2 Reproduction of Original Event

The CLT Protocol shall be able to reproduce by request, the original CEE Event that is modified while in transit.

3.4 Conformance Level 3 – Secure Against Local Administration Attacks

The core theme for the Conformance Level 3 transport requirements is the capability for CLT Protocol to securely log in the presence of local administration attacks.

3.4.1 Tamper Resistant

The CLT Protocol shall maintain integrity mechanisms resistant to tampering by local administrator (e.g. perfect forward secrecy).

3.4.2 Record Channels

The CLT Protocol shall be able to provide support for multiple channels within a session.

3.4.3 Profile Channels

A CLT Protocol channel shall be able to have CEE-specific metadata bound to it, such as a CEE Event Profile. This metadata can be used by the Event Sender/Receiver to exchange record format data or reduce duplicative data from being sent.

4 CTL Implementation Requirements

4.1 Conformance Level 0 – Core Requirements

4.1.1 Support CLT Protocol Level 0

The implementation must support at least a Conformance Level 0 of the CLT Protocol.

4.2 Conformance Level 1 – Basic Requirements

4.2.1 Support CLT Protocol Level 1

The implementation must support at least a Conformance Level 1 of the CLT Protocol.

4.2.2 Sender-side Buffering

The CLT Protocol shall be capable of sender-side buffering, e.g. event record is retained in a recoverable fashion on sender until server indicates that event has been received.

4.2.2.1 Single Log Record Buffering

The client shall retain each log record until the server has acknowledged (ACKed) the reception of the message; ideally this ACK should include a hash or signature whereby the sender can validate the message was correctly received .

4.2.2.2 Batch Log Record Buffering

The client shall retain each batch of log records until the server has ACKed the reception of the batch message; ideally this ACK should include a hash or signature whereby the sender can validate the message was correctly received.

4.2.2.3 Enable and Disable Switch

The CLT Protocol shall have a switch that enables and disables send-side buffering.

4.2.3 Log in Limited Network Environments

The CLT Protocol shall be capable of reordering the event transmission queue so that the most important messages receive priority. Many environments utilize Network Address Translation (NAT). The CLT Protocol shall be capable of functioning correctly in that environment.

4.2.3.1 Retransmission Priority

The Priority of the event retransmission queue shall be settable by the application.

4.2.3.2 Network Address Translation (NAT)

The CLT Protocol shall be capable of communicating in a Network Address Translation (NAT) environment.

4.3 Conformance Level 2 – Log In The Presence of Attackers

The implementation must support at least a Conformance Level 2 of the CLT Protocol.

4.4 Conformance Level 3 – Secure Against Local Administration Attacks

4.4.1 Support CLT Protocol Level 3

The implementation must support at least a Conformance Level 3 of the CLT Protocol.

4.4.2 Secure against local administration attacks

The CLT Protocol shall, on average, transmit event records within 1 second of event record creation.

4.4.3 Event Source Channel Binding

Applications sending events over the CLT Protocol shall be able to bind the event records to a CLT Protocol channel.

4.4.4 Event Destination Channel Binding

Applications receiving events over a CLT Protocol channel shall be able to reconstruct the full event record based on the channel contents and previously exchanged channel metadata.

4.4.5 Channel Profiles

Applications shall be able to bind one or more event profiles to a registered channel.

4.4.6 Continuous Operation

The CLT Protocol shall be able to support load balancing and gracefully failover to backup servers when the primary is lost.

5 Summary

The Common Log Transport (CLT) provide features necessary to support the end-to-end audit process by extending the event record representation to include the essential confidentiality, integrity, and availability of audit services. This allows systems to share log information with each other, a repository, or end user in a standard way. A CLT Protocol must meet a given set of tiered requirements, which are based on an enterprise’s particular environment. These include core, basic, and optional (optional because these requirements will not be applicable to all environments). For example, a CLT Protocol core requirement is to be able to transmit a CLS Encoded CEE Event. More advanced CLT Protocols may provide things like encryption and full acknowledgments. The CEE CLT component also defines transport mappings. A CLT mapping defines a standardized way for CEE Events to be transmitted over a certain CLT Protocol.

This document defines the feature and implementation requirements for a CEE CLT Protocol. MITRE provides this document to the CEE Community for their validation and approval.

This page intentionally left blank.

Appendix A References

[1] F. Miao, Y. Ma, and J. Salowey. (2009, March) Transport Layer Security (TLS) Transport Mapping for Syslog (RFC 5425). [Online]. http://tools.ietf.org/html/rfc5425

[2] Rainer Gerhards. (2009, March) The Syslog Protocol (RFC 5424). [Online]. http://tools.ietf.org/html/rfc5424

[3] CEE Editorial Board. (2010, November) Common Event Expression: CEE Dictionary and Event Taxonomy Specification v0.5.1.

[4] CEE Editorial Board. (2010, February) Common Event Expression: CEE Architecture Overview.

[5] The MITRE Corporation. (2008, June) Common Event Expression: CEE, A Standard Log Language for Event Interoperability in Electronic Systems. [Online]. http://cee.mitre.org

[6] W3C. (2008, November) Extensible Markup Language (XML) 1.0 (Fifth Edition). [Online]. http://www.w3.org/TR/2008/REC-xml-20081126/

[7] CEE Editorial Board. (2010, November) Common Event Expression: CEE Log Syntax (CLS) Specification, Version 0.5.1.

[8] Andrew Buttner and Neal Ziring. (2009, March) Common Platform Enumeration (CPE) - Specification, Version 2.2. [Online]. http://cpe.mitre.org/files/cpe-specification_2.2.pdf

[9] Brant A. Cheikes and David Waltermire. (2010, August) Common Platform Enumeration: Naming Specification, Version 2.3 (DRAFT). [Online]. http://csrc.nist.gov/publications/drafts/nistir-7695/draft-nistir-7695_cpe-naming-2_3.pdf

[10] The MITRE Corporation. (2008, June) Common Event Expression: CEE, A Standard Log Language for Event Interoperability in Electronic Systems. [Online]. http://cee.mitre.org

[11] International Organization for Standardization (ISO), "Data elements and interchange formats - Information interchange - Representation of dates and times," ISO, Geneva, ISO 8601:2004(E), 2004.

[12] D. Crockford. (2006, July) The application/json Media Type for JavaScript Object Notation (JSON). [Online]. http://www.ietf.org/rfc/rfc4627.txt?number=4627

[13] W3C. (2008, November) Extensible Markup Language (XML) 1.0 (Fifth Edition). [Online]. http://www.w3.org/TR/2008/REC-xml-20081126/

[14] W3C. (1996, March) Extended Log File Format. [Online]. http://www.w3.org/pub/WWW/TR/WD-logfile.html

This page intentionally left blank.

Appendix B Definitions

audit	The process of evaluating logs within an environment (e.g., within an electronic system). The typical goal of an audit is to assess the overall status or identify any notable or problematic activity.
category	*see event category*
channel	*See full-duplex pipe*
event	A single occurrence within an environment, usually involving an attempted state change. An event usually includes a notion of time, the occurrence, and any details the explicitly pertain to the event or environment that may help explain or understand the event's causes or effects.
event category	Groups events based upon one or more event categorization methodologies. Example methodologies include organization based upon what happened during the event, the involved parties, device types impacted, etc.
event field	One characteristic of an event. Event fields are defined in the CEE Dictionary and used in event records. Examples of an event field include date, time, source IP, user identification, and host identification. An event field relates a name identifier with a single field value.
event record	A collection of event fields that, together, describe a single event. Terms synonymous to event record include "audit record" and "log entry".
field	*see event field*
full duplex pipe	Messages can be exchanged in both directions simultaneously.
log (n)	A collection of event records. Terms such as "data log," "activity log," "audit log," "audit trail," "log file," and "event log" are often used to mean the same thing as log.
log (v)	The act of recording events into logs. Examples of logging include recording events into records a text log file, or storing the data in binary files or databases.
profile	A description of events, including event fields, event categories, and tags, that are generated by a product or relate to a specific capability (e.g., authentication or configuration management, firewall, signature detection, routing).
record (n)	*see event record*
record (v)	The act of saving the details of an event; recording an event as an event record.

This page intentionally left blank.