MITRE

Common Event Expression

Architecture Overview

Version 0.6

The CEE Board
May 2011

This page intentionally left blank.

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

LICENSE

The MITRE Corporation (MITRE) hereby grants you a non-exclusive, royalty-free license to use CEE for research, development, and commercial purposes. Any copy you make for such purposes is authorized provided that you reproduce MITRE's copyright designation and this license in any such copy.

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.

Acknowledgments

This architecture document is the result of the combined efforts of the CEE Board and the individuals from the CEE Community who provided ideas, recommendations, and inputs. The CEE Board wishes to thank their colleagues who reviewed drafts of this document, providing comments that contributed to its development. Specific contributions the CEE Board would like to acknowledge include Raffael Marty for the concept of the Field Dictionary and Taxonomy, and Eric Fitzgerald for the idea of event profiles and recommending the use of conformance levels. The CEE Board also gratefully acknowledges and appreciates the many contributions from individuals representing government agencies and the private sector, including Sanford Whitehouse, Dr. Anton Chuvakin, and David Corlette. Additionally, the CEE Board would like to thank Francis Duff (MITRE) and Linda K. Jones (MITRE) for helping organize and draft this document.

This page intentionally left blank.

Abstract

This Common Event Expression (CEE™) Architecture defines the structure and components that comprise the CEE event log standard. This architecture was developed by MITRE, in collaboration with industry and government, and builds upon the Common Event Expression Whitepaper [1]. This document defines the CEE Architecture for an open, practical, and industry-accepted event log standard.

This document provides a high-level overview of CEE along with details on the overall architecture and introduces each of the CEE components including the data dictionary, syntax encodings, event taxonomies, and profiles. The CEE Architecture is the first in a collection of documents and specifications, whose combination provides the necessary pieces to create the complete CEE event log standard.

KEYWORDS: CEE, Logs, Event Logs, Audit Logs

This page intentionally left blank.

Table of Contents

1 Introduction 1

1.1 Scope 1

1.2 Purpose 1

1.3 Document Organization 1

2 Event and Audit Basics 2

2.1 Background 2

2.2 Definitions and Terminology 2

3 CEE Architecture 4

3.1 Approach 4

3.2 Design Goals 4

3.3 Architecture Components 5

3.4 CLS: CEE Log Syntax 6

3.5 CDET Dictionary 6

3.6 CDET Taxonomy 7

3.7 CELR: CEE Event Log Recommendations 7

3.8 CEE Profile 8

3.9 CLT: CEE Log Transport 8

4 CEE Management 9

4.1 CEE Management Process 9

4.2 CEE Adoption 9

5 Summary 10

Appendix A References A-1

Appendix B Definitions B-1

List of Figures

Figure 1: Standardizing Event-Log Relationships 2
Figure 2: Definitions Mapped to CEE 3
Figure 3: CEE Architecture and Components 5
Figure 4: Event Field Instance 6
Figure 5: CEE Profile Contents 8
Figure 6: CEE Request Management Process 9

List of Tables

Table 1: Example CEE Tag Types and Tags 7

1 Introduction

The Common Event Expression (CEE™) Architecture enables an open, practical, and industry-accepted event log standard. This architecture is a coordinated industry initiative, developed by a community of vendors, researchers, and end users. The primary goal of this document is to introduce an open, practical, and industry-accepted framework that standardizes the description, representation, and exchange of event records between electronic systems.

The overall purpose of CEE is to improve the audit process and users' ability to effectively interpret and analyze event log and audit data. Through CEE, the limited interoperability offered by current event and log formats will be corrected.

MITRE coordinates the CEE Architecture as part of its larger *Making Security Measurable* initiative (http://measurablesecurity.mitre.org).

1.1 Scope

This document is an introduction to the CEE Architecture. This document is not the complete architecture, as each component will be further detailed in its own, subsequent specification. The architecture is based on inputs from the CEE Community, the CEE Editorial Board, and The MITRE Corporation.

1.2 Purpose

This document introduces CEE and the CEE Architecture to the CEE Community for validation and approval. The CEE Community is vital to the success and adoption of CEE; therefore feedback and discussion is needed 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).

1.3 Document Organization

This document is organized into the following sections:

Introduction: This section identifies the scope, purpose, and approach for this document.

Event and Audit Basics: This section defines some basic terminology and provides an overview of event management and audit requirements.

Architecture: This section provides an overview of the CEE Architecture, including its components and design considerations.

Management: This section provides an overview of the CEE change management and conformance processes.

Summary: This section identifies the components that were discussed and summarizes the next steps to ensure the success of CEE.

2 Event and Audit Basics

In order to understand the CEE Architecture, an agreement must be reached as to the definition of terms and the design goals.

2.1 Background

Organizations routinely undertake the expensive task of auditing their electronic systems. Some audits are performed to identify problems, reduce unnecessary overhead, or to maintain compliance with regulatory laws. Every log may contain critical information about prior and ongoing electronic events. Examples of some of these events include electronic events such as logon, connect, and write, or physical events such as building access or equipment pressure readings. These electronic events reflect status, threats, and other observable environment changes that allow an enterprise to maintain constant situational and informational awareness. Today, there is no standard for representing and describing these events in logs. This is a significant data management problem, since enterprise-wide situational awareness depends on the ability to process and analyze event data. The CEE Architecture addresses this audit problem by standardizing the event-log relationship by normalizing the way events are recorded, shared, and interpreted (Figure 1).

Figure 1: Standardizing Event-Log Relationships

The CEE Architecture standardizes the representation of events by providing tools similar to the dictionaries, grammar books, communication mediums (e.g., letters, e-mails, newspapers), and guidance used to support natural languages (e.g., English, Spanish, Latin). The architecture groups these tools across four areas: terminology dictionaries, representation (e.g., grammar rules), transport (e.g., e-mails, web services), and recommendations. These areas map directly to the four CEE Architecture components: CEE Dictionary and Event Taxonomy (CDET), CEE Log Syntax (CLS), CEE Log Transport (CLT), and the CEE Event Log Recommendations (CELR). The CEE Dictionary and Event Taxonomy provide a controlled vocabulary for the consistent description of event details. CLS defines the event language and event encodings for representing CEE events. CLT defines the requirements for secure, reliable recording and transmission of events. The CELR provides profiles for commonly used and product-generated CEE events.

2.2 Definitions and Terminology

This document uses the terms **event**, **event category**, **event field**, **event record**, **log**, **audit**, **recording**, and **logging**, which are defined below.

> An **event** is 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. CEE is concerned only with those events that are recording within an event record.
>
> **Event categories** group 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. The CDET Taxonomy defines a listing CEE Tags, which represent common event categories.

An **event field** describes one characteristic of an **event**. Examples of an **event field** include date, time, source IP, user identification, and host identification. A collection of commonly used fields are defined by the CDET Dictionary.

An **event record** is a collection of **event fields** that, together, describe a single **event**. Terms synonymous to **event record** include "audit record" and "log entry". In CEE, the format of an event record is defined by the CLS and represented using a CLS Encoding.

A **log** is 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**.

An **audit** is 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.

Recording is the act of saving the **event fields** associated with a single **event** as an **event record**.

Logging is the act of collecting **event records** into **logs**. Examples of **logging** include storing log entries into a text log file, or storing audit record data in binary files or databases.

The relationship between these terms can be summarized as: an **event** is described via its **fields**, which are chronicled in a **record** that is collected in a **log** and evaluated during an **audit**. Figure 2 shows the connection between the definitions and their corresponding CEE component.

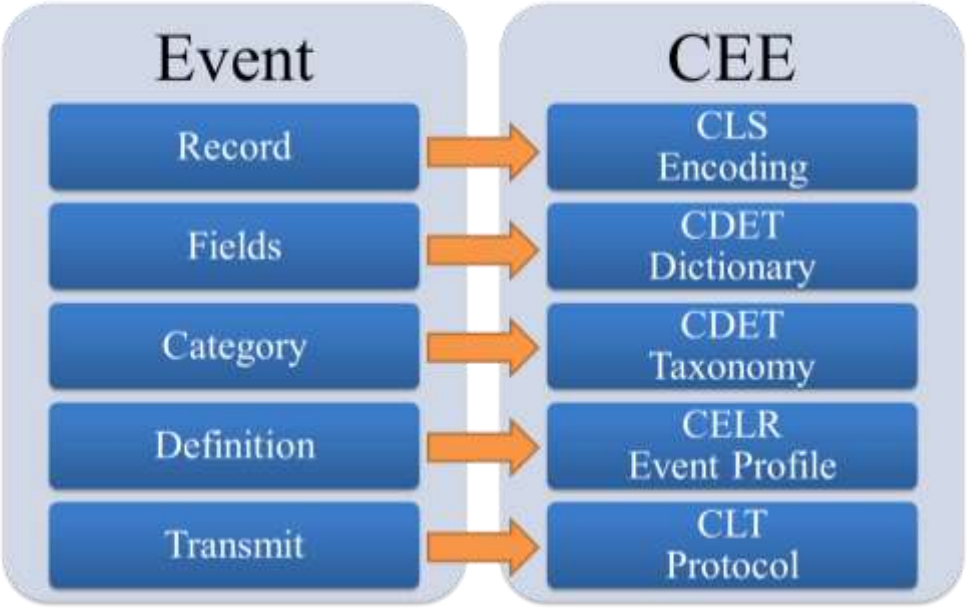

Figure 2: Definitions Mapped to CEE

3 CEE Architecture

The CEE Architecture provides the specifications and documents necessary to improve event management by standardizing the creation and interpretation of event records. Though this architecture could be adapted to events in any environment, the principal focus of CEE is to standardize event records produced within electronic systems such as computers and sensors. The CEE Architecture standardizes the event-to-record-to-log process. The use of CEE enables a reversible process, to allow the five architecture components to interact, which provides the functions necessary for the two event management activities: logging events and automating log interpretation.

3.1 Approach

The CEE Architecture and event model was developed using an iterative approach. An initial CEE Whitepaper [1] was published in 2008. Additional information was gathered from the community of interest and existing event log standards (e.g., IDMEF, SDEE, CIEL, CEF) were researched. A preliminary architecture was defined by analyzing the research results and by identifying the shortfalls of existing approaches to the event standardization problem. To elicit input, the preliminary architecture was provided for community review and comment. In addition, the preliminary architecture was prototyped and validated through MITRE research activities, tested during multiple exercises and vetted by the CEE Community via the CEE mailing list.

3.2 Design Goals

Due to the many uses of event records, CEE is designed to address many diverse log and audit needs. To encourage widespread adoption, the criteria and considerations for the architecture must address current community requirements as well as deficiencies identified with previous standardization attempts. Described below are general design goals for the CEE Architecture. Included with each goal is an explanation why it is important to the success of CEE.

1. **Syntax Neutral**: While the CEE Architecture is built upon XML, users of CEE shall be able to exchange additional syntaxes in order to be able to support various event log formats. Attempts to create XML-exclusive log specifications (e.g., SDEE, IDMEF) had limited success, and investigation to date has shown the CEE Community needs more options. The CEE Community supports interchangeability between XML and lower overhead text-based and binary protocols. To maximize usability and promote adoption, the CEE standard shall focus on the event records and the event attributes, while allowing multiple syntax options.
2. **Flexible**: CEE Architecture shall provide options in the choice of event fields and syntax encodings to provide flexibility to the end user. Providing some flexible options shall ensure producers and consumers can implement CEE in a manner that makes sense for their intended use or environment.
3. **Extensible:** CEE Architecture shall focus on addressing representative event records for a typical organization’s environment. Therefore, CEE may not address all events and event fields. Vendors and users shall be provided the ability to define additional events and event fields without compromising CEE device compatibilities.
4. **Compatible:** CEE Architecture shall strive to utilize or provide compatibility with widely used standards, such as Syslog. By dividing CEE into multiple components and abstracting features such as the event terminology, a level of backwards-compatibility is provided. Future changes shall strive not to compromise compatibility with previous versions, yet the

compatibility goal shall not prevent future additions and changes to CEE when the CEE Community believes it is necessary. Devices supporting older versions of CEE shall always be able to receive and process event records conforming to recent specification versions.

5. **Comprehensive:** All events, event fields, and field values shall be represented within the CEE Architecture or shall be capable of being specified within a compatible extension.
6. **Maintainable:** CEE Architecture shall be defined in a manner to ensure that maintenance and updates have minimal impact on CEE and component specifications. While the future of the events and their uses is unpredictable, it is certain that quantity of events and dependency upon them will continue to grow.
7. **Easily Implementable**: CEE Architecture and its components shall be defined such that it is easy to implement by both event record producers and consumers.

As the CEE Architecture and specifications evolve, the CEE Editorial Board and CEE Community shall ensure the design goals identified in this section are used to vet future architecture modifications and enhancements.

3.3 Architecture Components

The CEE Architecture is comprised of four (4) components: the CEE Dictionary and Event Taxonomy (CDET), CEE Log Syntax (CLS), CEE Log Transport (CLT), and the CEE Event Log Recommendations (CELR). The dictionary portion of the CDET defines the event terminology (i.e., field names and value types), while the CDET Taxonomy provides entries with which to categorize events. The CLS defines the event structure and representation. CLT allows for the sharing of event information. Finally, the CELR provides guidance as to which events and related fields should be logged.

The CEE Architecture (Figure 3) can combine these components to record an event into a log. First, an event occurs. The CELR, using the CDET Dictionary and Taxonomy, specifies which events and event fields are recorded. These fields are recorded into a record according to the CLS Encoding. Finally, a CLT-defined standard protocol can share these records or transmit them to a log repository.

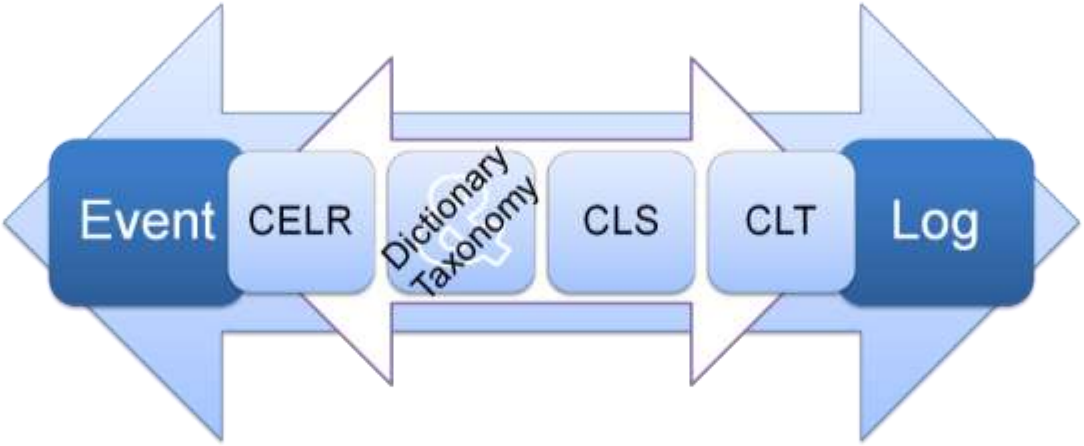

Figure 3: CEE Architecture and Components

Since the CEE process is bidirectional, the reverse process can also occur – recreating events based on event logs. The log is received using a standard protocol defined by the CLT component and is parsed according to a CLS Encoding. Once the fields have been parsed, systems and end users can understand what happened during the event by cross-referencing the event record fields with those defined in the CDET Dictionary. Finally, the event record, along with the associated fields, can be validated against the CELR to determine whether they adhere to audit policy or best practice recommendations.

3.4 CLS: CEE Log Syntax

The CEE Common Log Syntax (CLS) is how each CEE Events are represented. Each CEE Event can be represented using one or more CLS Encodings. These CLS Encodings are well-defined syntaxes that CEE event producers write and CEE consumers process.

In general, each event record describes how an event is categorized and a collection of relevant event data. Each of these pieces of data is represented by an instance of an **event field**. A field instance is a combination of a field name, such as those defined in the CDET Dictionary, and one or more values (Figure 4).

`

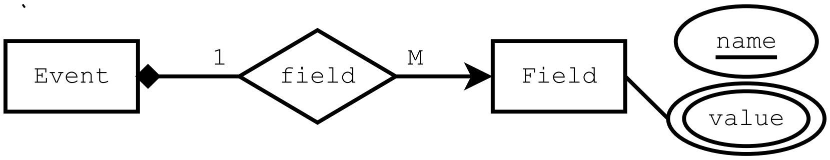

Figure 4: Event Field Instance

In addition to defining the general CEE event language, the CLS component defines a number of different encodings for a CEE Event. Since each encoding is based on the same event structure, translating between different CLS Encodings is efficient and straightforward. Based on CEE Community inputs, CLS will minimally support eXtensible Markup Language (XML) [2] and JavaScript Object Notation (JSON) [3]. Consideration will be given to providing compatibility with other syntaxes, such as RFC5424 Syslog [4] compatible Structured Text, binary, or the W3C Extended Log Format (ELF) [5] syntax.

CEE Events should use the event field names and associated value types defined by the CDET Dictionary and categorize events via the event category tags of the CDET Taxonomy.

3.5 CDET Dictionary

The CDET Dictionary defines a collection of event fields and field value types for use with CEE Events. The fields and types are used within event records to specify the values of an event property associated with a specific event instance.

Each field type definition has a name, format, and description. The type format is either a union or restriction. The restriction type places a limitation on the acceptable values supported by that type. One restriction is a pattern indicating how the character sequence an acceptable value should match, such as an IPv4 address field type must have a specific pattern: `\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}`. The format may also specify the minimum or maximum value for an integer, such as a network port is a number between 0 and 65535. The union type defines entry types as combination of other types (e.g., an IP address type may be a union of an IPv4 and IPv6 address type).

Each **event field** definition is represented by a name that identifies one event characteristic (e.g., source IPv4 address, filename, username, destination port number). Each field is defined by a unique name, definition, and is associated with one field type (e.g., integer, string, timestamp, IPv4 address).

For both fields and field types, the name is a unique character sequence used to reference a specific definition. Using a name in the CDET Dictionary is similar to using a standard dictionary. Users can look up the meaning of, or locate the proper term to describe a certain event characteristic. For example, if a product wants to provide the account name of a user involved in an event, a search through the CDET Dictionary entries would inform that the correct event field to use would be *acct.name*. Similarly, if a product records an event field entitled *proc_id*, the Dictionary would explain that this field describes the numerical process identifier of an executing process.

3.6 CDET Taxonomy

The Event Taxonomy is the other half of the CDET component. The CDET Taxonomy defines a collection of **CEE Tags** that can be used to systematically categorize events. Its goal is to support common event categorization methodologies and identify records that pertain to similar types of events. CEE Tags are grouped by common event categories based on their tag type, which are used by event producers can provide obvious and consistent event categorization. Users and event consumers can leverage these categories to improve event correlation or easily locate certain classes of events.

The CDET Taxonomy defines a tag type as way to categorize events. Each tag type consists of one or more CEE Tags. Each tag represents on event classification concept and is associated with a unique name. These tag types allow each event to be associated with multiple tags representing multiple categories. This gives the event consumers the flexibility to identify similar events based upon their needs.

Common tag types include event action, status, and object, and might include other categorizations such as attack type, device type, or other categorizations that are required by the event consumer. An example list of event tag types and names are in Table 1.

Table 1: Example CEE Tag Types and Tags

Tag Type	Tag Name
action	start, stop, execute, read, delete, login
object	file, acct, app, db, system, malware
status	success, failure, error
attack	dos, exploit, xss, buffer-overflow
device-type	finance, dev, prod, test, dmz

These CEE Tags can be specified within an event record to indicate that event's categorization. For example, an event could be tagged with a `login`, `success`, and `db` tag, indicating that the event probably pertains to a successful login to a database.

3.7 CELR: CEE Event Log Recommendations

The purpose of the CEE Event Log Recommendations (CELR) component of the CEE Architecture is to provide recommendations as to which events and fields should be recorded in certain situations. CELR provides this guidance in the form of a machine-readable Event Profiles. An **Event Profile** defines the optional and mandatory fields for a specific CEE Event. To conform to an Event Profile, a CEE Event must contain the event fields and values defined as part of the Event Profile.

Each CEE-defined Event 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.

3.8 CEE Profile

A **CEE Profile** is a document that defines Event Profiles, CDET Dictionary Fields, and Taxonomy Tags. This allows for all of the CEE event profiles and vocabularies to be packaged with a single document, as shown in Figure 5.

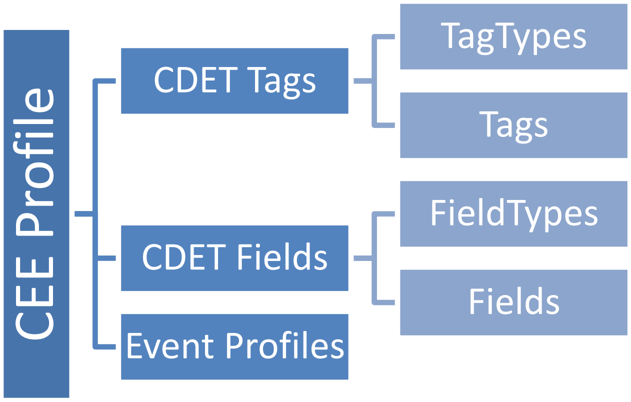

Figure 5: CEE Profile Contents

There are three (3) different types of CEE Profiles: base, functional, and product. A **Functional Profile** is a CEE Profile that defines Event Profiles for events that comprise a certain functional capability. For example, a "firewall" functional profile can be defined consisting of "connection allow" and "connection block" events. Similarly, an "authentication management" function can be composed of "account login," "account logout," "session start," and "session stop." A **Product Profile** allows vendors to define their own CEE Profiles to describe the events produced by their products. The **Base Profile** is a special CEE Profile provided by CEE that defines the base CEE Event structures. The Base Profile must be used by all CEE Events.

Tools can be used to check whether a CEE Event is compliant with a CEE Profile. If the event contains all of the fields required by the profile, and each field's value corresponds to the field's value type defined in the profile, then the event record is said to be compliant with the profile.

3.9 CLT: CEE Log Transport

The CEE Log Transport (CLT) provides the technical support necessary for a secure 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 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 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 [6]. This Mapping would define that the CEE Event must be encoded using an RFC5424 Syslog-compatible [4] 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 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 audit services.

4 CEE Management

4.1 CEE Management Process

CEE "products" can be defined as anything created in support of the CEE Architecture, including this document, component specifications, the CDET Dictionary and Taxonomy, CLS Encodings, Event Profiles, etc. The CEE Editorial Board maintains oversight for all new products as well as updates to existing products. The CEE Editorial Board reviews inputs and requests from the community regarding new product recommendations, updates and, eventually, product retirement. Figure 6 outlines the change management process that will be used for all CEE products.

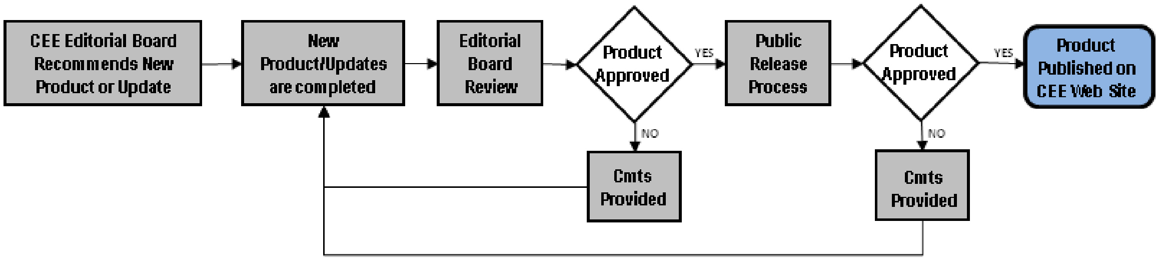
CEE Editorial Board Recommends New Product or Update
New Product/Updates are completed
Editorial Board Review
Product Approved
YES
NO
Cmts Provided
Public Release Process
Product Approved
YES
NO
Cmts Provided
Product Published on CEE Web Site

Figure 6: CEE Request Management Process

Prior to new products becoming finalized, a draft product will be made available to the CEE Community for review and comment. Once the comment period has ended, per the direction of the CEE Editorial Board, the draft document will be updated to address comments, re-reviewed, finalized, and published.

4.2 CEE Adoption

CEE adoption is defined as supporting the CEE effort by having the capability to properly process CEE-compatible event records. In order to promote interoperability and encourage vendor adoption, the CEE adoption program will establish requirements and measures for evaluating product and process conformance to the CEE specifications.

As not all products require all of the CEE specification, the adoption program allows vendors and customers to conform to only those event-processing capabilities that make the most sense. Currently, CEE recognizes six (6) event-related capabilities:

- **Event Producer**: A product that generates event records according to the CEE Log Syntax
- **Event Consumer**: A product that receives and validates CEE event records
- **Event Manager**: A product that allows users to search, correlate, or otherwise manage event records using CEE field and tag definitions
- **Event Translator**: A product that converts other event records into CEE-compliant records
- **Profile Producer**: A formal, public specification of the event records and associated event fields and tags within an event profile
- **Profile Validator**: A product that validates the content and format of CEE event records against one or more event profiles

The full CEE adoption program, capabilities, and conformance requirements will be defined once the CEE component specifications have been finalized.

5 Summary

The successful adoption of CEE depends on its ability to meet the needs of the audit and log community. It is believed that CEE will revolutionize how event records are created and utilized. CEE's approach for creating a system-neutral and vendor-neutral event standard will facilitate and enable true interoperability. This document highlights the overall CEE Architecture and CDET, CLS, CLT, and CELR components that combine to solve the ongoing event management problems.

As the CEE Architecture and supporting standards evolve, it is important to be cognizant of the CEE design considerations and goals; they are important criteria specified by the CEE community. In addition, the change management and conformance processes will need to be defined in further detail and implemented to ensure CEE continues to evolve and remain useful to the community.

The CEE Community is vital to the success and adoption of CEE. Given this, inputs, feedback, and discussion are crucial to the production of an open, practical, and industry-accepted standard. Comments concerning the CEE Architecture, or CEE in general, should be submitted to the CEE Discussion List (cee-discussion-list@lists.mitre.org) or to the MITRE CEE Team (cee@mitre.org).

Appendix A References

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

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

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

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

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

[6] 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

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

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***
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***
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.