MITRE

Common Event Expression

Architecture Overview

Version 0.5

The CEE Editorial Board
May 2010

Approved for Public Release;
Distribution Unlimited.
Case 10-2296

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 Data Dictionary and Taxonomy, and Eric Fitzgerald for the idea of event profiles and recommendation 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, Log Analysis, Log Management, SIEM

This page intentionally left blank.

Table of Contents

List of Figures

List of Tables

1 Introduction

The Common Event Expression (CEE) Architecture defines 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 generating electronic trails. CEE is developed on top of the previous log standardization attempts and makes possible the "blessing" of other log-related standards under CEE umbrella.

The overall purpose of CEE is to improve the audit process and users' ability to effectively interpret and analyze event log and audit data as well as to enable creation of useful and efficient log records within applications. 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 the electronic event trails from electronic or physical 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 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).

Event
CEE
Log

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, Japanese). 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: Common Event Expression Dictionary and Taxonomy (CDET), Common Log Syntax (CLS), Common Log Transport (CLT), and the Common Event Log Recommendations (CELR). CEE provides a dictionary and taxonomy for consistent description of event details whereas CLS provides an encoding scheme for processing event data. The CLT standardizes mechanisms for secure, reliable recording and transmission of events. The CELR proposes recommended log events and attributes for IT devices.

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.
>
> **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.

An **event field** describes one characteristic of an **event**. Examples of an **event field** include date, time, source IP, user identification, and host identification.

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".

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 creating an event record comprised of the **event fields** associated with a single **event**.

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 is depicted below (Figure 2): an **event** is described via its **fields**, which are chronicled in one or more **records** that are collected in a **log** and evaluated during an **audit**.

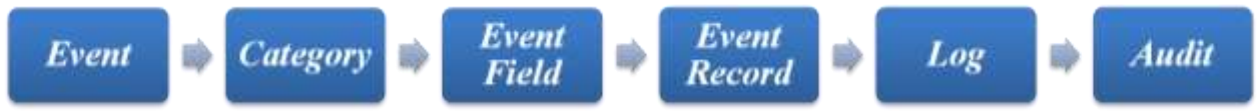

Figure 2: Relationship between Terms

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[1], SDEE[2], CIEL[3], CEF[4]) 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. Additional opportunities for feedback included the birds-of-the-feather (BOF) sessions held at BlackHat USA 2008 and 2009, and the RSA 2009 Conference. The CEE Board reviewed all feedback; applicable feedback and recommendations were incorporated into the CEE Architecture.

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. **Encoding Neutral**: CEE Architecture shall support multiple ways of encoding event records in order to support the variety of event log environments. Attempts to create XML log specifications (e.g., SDEE, IDMEF) had limited success, and investigation to date has shown the CEE Community does not want to be tied to a single specific syntax or encoding. 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 supplying multiple encoding choices.

[1] The Intrusion Detection Message Exchange Format (IDMEF). IETF RFC4765. http://www.ietf.org/rfc/rfc4765.txt
[2] Security Device Event Exchange (SDEE). Developed by ICSA Labs.
[3] Common Intrusion Event List (CIEL). The MITRE Corporation. http://cve.mitre.org/data/board/archives/2001-03/msg00013.html
[4] Common Event Format (CEF). ArcSight. http://www.arcsight.com/solutions/solutions-cef/

2. **Flexible**: CEE Architecture shall provide options in the choice of event fields and syntax encodings to provide flexibility to the end user within constraints defined by the CEE standard to preserve interoperability. Providing log producers and consumers some flexibility options allows for CEE in a manner that best suits 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, but still ensuring interoperability.
4. **Compatible:** CEE Architecture shall strive to utilize or provide compatibility with widely used components, such as XML, JSON, and 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 Taxonomy (CDET), Common Log Syntax (CLS), Common Log Transport (CLT), and the Common Event Log Recommendations (CELR). The CEE Dictionary defines the event terminology (i.e., field names and value types), the Taxonomy provides entries with which to categorize events, and the CLS defines the event representation. CLT allows for the transport of event information between producers and consumers. 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 CEE Dictionary and Taxonomy, specifies which events and event fields are recorded. These fields are recorded into a record according to the CLS language. Finally, a CLT-defined standard protocol can share these records or transmit them to a log repository or log consumer.

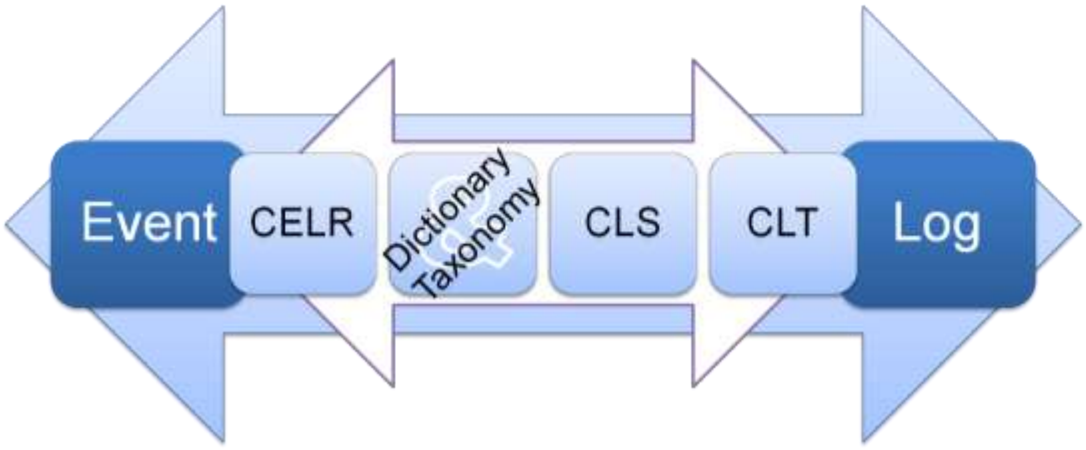

Figure 3: CEE Architecture and Components

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 the CLS language. 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 CEE 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: Common Log Syntax

The CEE Common Log Syntax (CLS) is how the event and event data is represented. The event syntax is what an event producer writes and what an event consumer processes.

In general, each event record describes how an event is categorized and a collection of relevant event data. Each "piece" of event data is represented in an **event field**. A field is a combination of a field name, such as those defined in the CEE Dictionary, and an **entry** (Figure 4). An entry can represent a field value, tag, or another field. To handle collections, CEE provides a special type of field called a **set**, which contains an unordered group of entries.

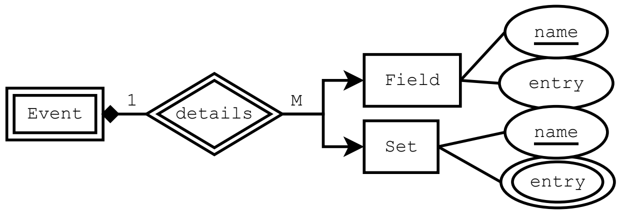

Figure 4: Event Fields

In addition to defining the general CEE event representation format, the CLS component defines a number of different ways to encode the event and event fields. 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], JavaScript Object Notation (JSON) [3], and a Syslog v3 [4] compatible structured text. Consideration will be given to providing compatibility with other encodings, such as binary syntaxes or the W3C Extended Log Format (ELF) [5].

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

3.5 CEE Dictionary

The CEE Dictionary defines a collection of event fields and value types that can be used within event records to specify the values of an event property associated with a specific event instance.

Each **event field** represents 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 value type (e.g., integer, string, timestamp, IPv4 address).

To support both expressive and terse event encodings, each event field entry may define two names: one long and one short (e.g., *InterfaceIPv4Address*, *ipv4*). Using a name in the CEE 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 CEE Dictionary entries would inform that the correct event field to use could be *AccountName* or *aName*. Similarly, if a product records an event field entitled *procId*, the Dictionary would explain that this field describes the process identifier of an executing process.

A **value type** defines the valid values for an event field. Each value type entry has a name, definition, and description. The type definition specifies range of acceptable values supported by the type. This is done by defining restrictions on an existing value type[5]. For example, a pattern restriction uses a regular expression pattern that all values of the value type must match. A value type definition for an IPv4 address might state that all valid values must match the pattern: `\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}`. Other restrictions can specify the minimum or maximum value for number, such as a network port is a number between 0 and 65535, or an enumeration of acceptable values.

In addition to defining restrictions, other ways to define the value type are through unions and sets. A union value type defines entry types as combination of other types (e.g., an IP address type is a union of an IPv4 and IPv6 address type). A set value type means that the type supports a collection of certain types (e.g., a set of integers).

3.6 CEE Taxonomy

The CEE Taxonomy defines a collection of "tags" that can be used to categorize events. Its goal is to provide a common vocabulary, through sets of tags, to help classify and relate records that pertain to similar types of events. Using Taxonomy tags, event producers can provide obvious and consistent event categorization identifiers. For example, users and event consumers can leverage these categories to improve event correlation or easily locate certain classes of events.

The CEE Taxonomy defines a **tag set** as way to categorize events. Each tag set consists of one or more **tags**. Similar to an event field, each tag entry has an identifying long and short name. These tag sets 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.

[5] The CEE specification defines several base types, such as string, integer, and Boolean types.

Common tag sets 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 categories and entry names is in Table 1. The combination of the entry name and type provides a unique Taxonomy identifier.

Table 1: Example Event Categories and Entries

Tag Set	Tag
action	start, stop, execute, read, delete, logon
object	file, acct, app, service, system, malware
status	success, failure, error
attack	dos, exploit, xss, buffer-overflow
device-type	host, ids, fw, router, web, db

The CEE Dictionary can declare tag set value types that reference a specific tag set defined in the Taxonomy. These tag set types can be used by fields and within events to indicate the event's categorization. For example, an event's *action* field could indicate it is a `logon` event, or a *taxonomy* set could state the event was about a `remove` action on a `file` object with a `success` status.

3.7 CELR: Log Recommendations

The purpose of the Common Event Log Recommendations (CELR) component of the CEE Architecture is to provide recommendations to developers and implementers of applications or systems as to which events and fields should be recorded in certain situations and what log messages should be recorded for various circumstances. CELR provides this guidance in the form of a machine-readable **profile**.

The CEE community provides all CEE identifiers in a default, "base" profile. All other profiles are required to extend from the CEE base profile.

Each profile can define and extend the CEE Taxonomy and Dictionary terminology. In addition to these two components, the profile can define functions and event structures. An **event structure** defines the optional and mandatory fields for all events or a certain type of event (using CEE Taxonomy identifiers). All event records that conform to the profile specification must contain the event fields defined by all applicable taxonomic event structures.

A **function** is a group of event structures that comprise a certain capability. For example, a "firewall" function can be defined consisting of "connection allow" and "connection block" event structures. Similarly, an "authentication management" function can be composed of "account logon," "account logoff," "session started," and "session stopped."

Tools can be used to check whether an event record is compliant with that event's profile. If the record contains all of the fields required by the appropriate event structures, 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.

Each CEE-defined 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. These profiles provide guidance to help product implementers produce the right event records for the environment. Vendors and third parties are encouraged to extend the published CEE profiles to better reflect the events produced by a certain product or environment.

3.8 CLT: Common Log Transport

The Common Log Transport (CLT) provides the technical support necessary for an improved log transport framework. A good framework requires more than just standardized event records, support is needed for international string encodings, standardized event record interfaces, and reliable, verifiable log trails.

Functionally, the CLT meets these requirements by defining **event streams**. Whenever a new application instance is created, that instance initializes a new event stream. Each event stream provides a strong binding with the application instance and allows for additional metadata (e.g., application instance ID, event ID, event sequencing hints) to be associated with each event record. An event stream may be implemented in the form of an application programming interface (API) specification or event recording service.

In addition to the application support, the CLT event streams supplement the CLS event record encodings to allow systems to share event records securely and reliably. The event stream can specify the CELR profile, string (e.g., byte order mark), and event encoding formats. The stream provides the additional 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 CEE Dictionary and Taxonomy, schemas, 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 5 outlines the change management process that will be used for all CEE products.

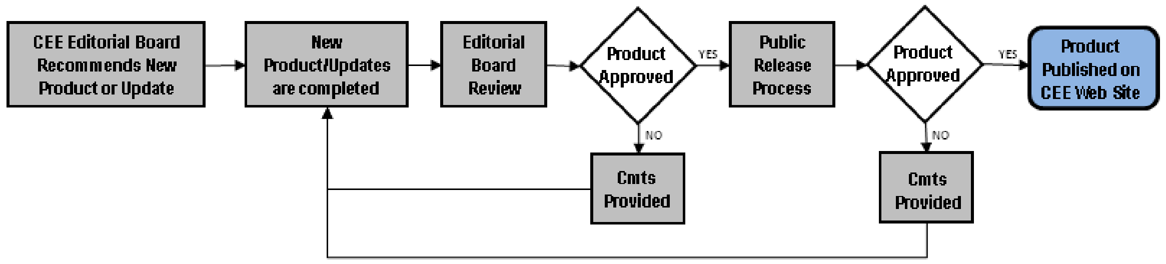
CEE Editorial Board Recommends New Product or Update
New Product/Updates are completed
Editorial Board Review
Product Approved
YES
NO
Cmts Provided
Public Release Process
Product Approved
YES
NO
Cmts Provided
Product Published on CEE Web Site

Figure 5: 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 Conformance

CEE conformance is defined as supporting the CEE effort by having the capability to encode or decode event information into CEE-compatible event records. In order to ease transition and encourage vendor adoption, CEE recognizes four (4) levels of conformance.

- Level 1 [Well-formatted Events]: Be able to record events in agreement with at least one CLS encoding syntax. An example of Level 1 conformance would be to record events using the CLS XML encoding.
- Level 2 [Search & Correlation Support]: Achieve Level 1 conformance, record event fields and values as defined by the CEE Dictionary, and specify the event categorization using CEE Taxonomy entries.
- Level 3 [Published Profiles]: Achieve Level 2 conformance as well as publish a valid CELR profile describing the events the product can generate. The profile must comply with the CEE Profile XML Schema and be made available at a publicly accessible Internet address.
- Level 4 [Adherence to Profiles]: Achieve Level 3 and have validation that the product's event records conform to the applicable official CEE-published event profiles. This entails ensuring all of the mandatory events and event fields are recorded according to the associated profiles.

Regarding conformance levels, "support" means that log producers must be able to generate the appropriate CEE event records (as defined by the applicable CELR profiles) whereas log consumers must be able to receive the appropriate event records. At this time, the transport mechanism is not currently part of the CEE conformance determination.

Figure 6 provides a notional diagram of how each conformance level builds upon the previous conformance level. As shown in the diagram, although it takes additional cost and effort to become fully compliant, the reward is significant in the sense that end users will be able to understand and leverage the event records generated by the compliant device.

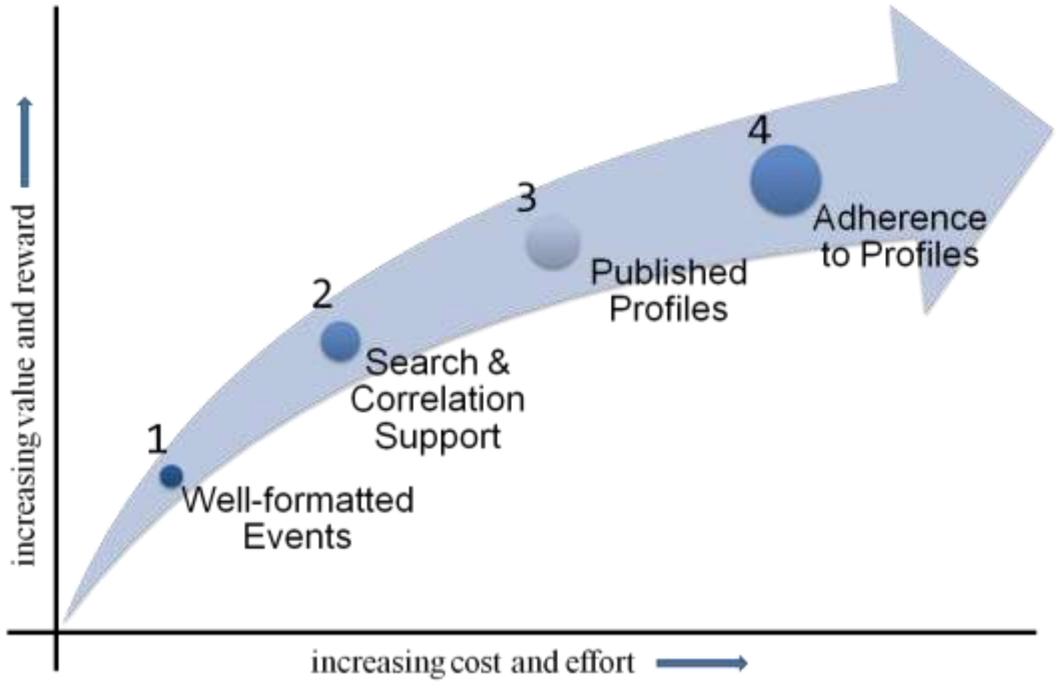
increasing value and reward
1
Well-formatted Events
2
Search & Correlation Support
3
Published Profiles
4
Adherence to Profiles
increasing cost and effort

Figure 6: Conformance Levels

5 Summary

The successful adoption of CEE depends on its ability to meet the needs of the audit and log community. 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 CEE Dictionary and Taxonomy, 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`).

This page intentionally left blank.

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] 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	groups events based upon one or more event categorization methodologies. The Taxonomy uses **tag sets** to categorize events
entry	an individual value associated with an event field
entry set	an event field that contains an unordered collection of entries
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 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 entry.
event record	a collection of event fields that, together, describe a single event. Terms synonymous to event record include "audit record" and "log entry".
event stream	a sequence of events produced by a single instance of an application or a single device.
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.
log entry	*see **event record***
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.
set	an unordered collection. *see also **tag set**, **field set**, **entry set***
tag set	set of related tags used to 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.

This page intentionally left blank.