Introduction to CEE v0.6

William Heinbockel
Tom Graves
{heinbockel,tgraves}@mitre.org
First things first

- CEE = Common Event Expression
- CEE Specifications released (v0.6)
- Initial CEE Repository available
- Latest CEE Information available at: http://cee.mitre.org
Organization

- 6 Sections

- Each section ends with a discussion
CEE OVERVIEW

CEE Architecture
Background

- Event
  - a single occurrence within an environment, usually involving an attempted state change

- Event Record
  - a collection of event fields that, together, describe a single event

- Log
  - a collection of event records

** From this point, "event" is used as shorthand for "event record" **
(Some) Other Event Standards

- XDAS
- CEF
- SDEE
- IDMEF
- CBE
- Syslog
- SNMP

http://xkcd.com/927
Design Goals

- Open, Neutral Standard
- Efficiency
- Simplicity
- Compatibility
  - Work in current event environments
  - Work with existing products
Event Management Environment

- Event Producer
- Event Consumer
- Intermediate System
  - Event Relay
  - Guard
Consuming Events

Receive  Parse  Comprehend  Analyze

Map  Normalize  Augment  Filter
Problem

- Effective analysis requires parsing and comprehension
- Parsing events is hard
- Comprehending events is harder
  - What "type" of event is it?
  - What does the event mean?
- Limited secure, resilient log protocols
Solution

Common Event Expression

CEE

CEE Log Transport

CEE Log Syntax

CEE Dictionary and Event Taxonomy

CEE Event Log Recommendations

CLT

CLS

CDET

CELR

Analyze
New Approach

Problem

- Receive
- Parse
- Comprehend
- Analyze

CEE

- CLT
- CLS
- CEE Profile
Discussion

1. What to do with non-events? I.e., status, debug, alert messages

2. Any missing event management pieces? Are they better suited for inclusion in EMAP?
EVENT MODELING

How CEE views events
Field & Tags

- Events are just a series of fields and tags
- Field :: a name and value associated with an object or property of an event
- Tag :: the event "type"
  - action tags = login, remove, read, block, search
  - status tags = success, fail, error
  - others? = hipaa, audit, critical, warning, info
Event Conceptual Model

Record := (Producer, Event)
Event := (id, time, Type, Subject?, Object+, Field*)
Type := (action, status, tag*)
Producer := (p_sys_id, p_prod_id, Field*)
Subject := (Field*)
Object := (Field*)
Field := (name, value*)
Structured Field Names

- **Format:** \[A-Za-z0-9_]\{1,32\}

- **Structure:**
  Role? Object? Semantic* Syntax Temporal?

- **Role:** Field Object's Event Role
  - \( p_ \) \( \rightarrow \) Event Record Producer
  - \( s_ \) \( \rightarrow \) Subject (Event Action Initiator)
  - otherwise, role is Event Object (Action Target)

- **Temporal:**
  - \( _{\text{old}} \) \( \rightarrow \) Old / Previous value
  - otherwise, current value
Field Name Examples

1. file_name
2. file_path
3. acct_id
4. prod_cpe
5. file_name_old
6. p_proc_name
7. p_sys_ipv4
8. s_sess_id
9. s_proc_id
10. fname_a_time
11. file_sha1_hash
12. src_ipv4
13. dst_ipv6
14. src_port
15. dst_mac
16. email_to_email
Discussion

1. Should field names have (some) structure?

2. Are there better ways to do field naming?
CEE EVENT LANGUAGE

Common Log Syntax (CLS)
CLS Overview

- **CLS Specification**
  - Defines a set of base field value types
  - Defines a Generic CEE Event Record Structure
  - CLS Encoding Requirements

- **CLS Encoding Specification**
  - Defines encodings to/from various syntaxes
  - XML
  - JSON
CLS Event Record

- Events are a sequence of fields
- Fields have a name and a sequence of values
- Every event must have 6 required core fields
  - $id$ :: Event ID
  - $time$ :: Event start time
  - $action$ :: Primary action of the event (login, read)
  - $status$ :: Result of the event action (success, fail)
  - $p_sys_id$ :: ID of the producing system
  - $p_prod_id$ :: ID of the producing product
CLS Field Value Types

1. string
2. binary
3. integer
4. float
5. timestamp
6. duration
7. ipv4Address
8. ipv6Address
9. macAddress
10. boolean
11. tag
Limitations

- Field values should be processed sequentially.
- Ordering of fields and field values must not be changed by intermediary systems.

<table>
<thead>
<tr>
<th>Area</th>
<th>Maximum Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoded Event Size</td>
<td>64 KB</td>
</tr>
<tr>
<td>Field Value Size</td>
<td>2 KB</td>
</tr>
<tr>
<td>Number of Fields</td>
<td>255</td>
</tr>
<tr>
<td>Number of Values per Field</td>
<td>255</td>
</tr>
</tbody>
</table>
CLS Event Record Structure

Event
- id
- time
- action
- status
- p_sys_id
- p_prod_id

Extensions
- Ext 0

Fields
- Field 0
- Name
- Value
- Value Type
- Data

Ext. Type
- Data
Extensions

- Augmentation
  - Non-destructive modification of events
  - Ordered

- Digital Signatures (planned; 2012Q1)
Example (XML)

```xml
<Cee xmlns="http://cee.mitre.org">
  <Event>
    <id>example-event-2</id>
    <time>2011-04-01T12:01:00-05:00</time>
    <action>download</action>
    <status>-</status>
    <p_sys_id>host.example.com</p_sys_id>
    <p_prod_id>product</p_prod_id>
    <Field name="tags"><tag>web</tag></Field>
    <Field name="file_name"><str>example.txt</str></Field>
    <Field name="file_data">
      <binary>RmlsZSBDb250ZW50Li4uAAo=</binary>
    </Field>
  </Event>

  <Augmentation order="1">
    <time>2011-04-01T14:11:53-04:00</time>
    <status>success</status>
    <p_sys_id>relay.example.com</p_sys_id>
    <p_prod_id>cee-relay</p_prod_id>
    <Field name="tags"><tag>hipaa</tag></Field>
  </Augmentation>
</Cee>
```
Example (JSON)

{"Event":{"id":"example-event-2","time":"2011-04-01T12:01:00-05:00","action":"download","status":[],"p_sys_id":"10.10.0.1","p_prod_id":"process","file_name":"example.txt","tags":"web","file_data":"b|RmlsZSBDb250ZW50Li4uAAo="},"Augmentation": [{"time":"2011-04-01T14:11:53-04:00","status":"success","p_sys_id":"relay.example.com","p_prod_id":"cee-relay","tags":"g|hipaa"}]}
Discussion

1. Do we need more/less required fields?
2. Do we need more/less field value types?
3. Ideas for addition event extensions
EVENT COMPREHENSION & ANALYSIS

CEE Profiles
CEE Profile Overview

- CEE Profile Specification
  - Documents the features and usage of a CEE Profile document

- CEE Profile XML Schema (XSD)

- CEE Profile Repository
  - Collection of CEE Profile XML Documents
CEE Profile Purpose

- Comprehension & Analysis of CEE Events
  - CEE Dictionary and Event Taxonomy (CDET) provides event vocabulary
  - CEE Event Log Recommendations (CELR) provides event profiles for common events
CEE Profile Structure

- Publicly available
- 3 Profile Types
- Definitions for
  - Field Types
  - Fields
  - Tag Types
  - Tags
  - Event Profiles
Field Type Definition

```xml
<FieldType name="actionTagType">
  <TagRestriction>
    <TagType>actionTag</TagType>
  </TagRestriction>
</FieldType>

<FieldType name="emailAddress">
  <Description>
    <Text_Title>E-mail Address</Text_Title>
  </Description>
  <StringRestriction>
    <Pattern>^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]+</Pattern>
  </StringRestriction>
</FieldType>
```
Field Definition

```xml
<Field name="file_name" type="string"/>

<Field name="time" role="object" type="timestamp">
  <Description>
    <Text_Title>Event Start Time</Text_Title>
    <Text>An ISO8601 compliant timestamp designating the date, time, and timezone offset when the event began</Text>
  </Description>
</Field>
```
Tag Type Definition

<TagType name="actionTag">
  <Description>
    <Text_Title>
      Action Tags
    </Text_Title>
  </Description>
</TagType>

<TagType name="statusTag"/>
Tag Definition

<Tag name="access" type="actionTag">
  <Description>
    <Text_Title>Access Event</Text_Title>
    <Text>…Text</Text>
  </Description>
</Tag>

<Tag name="read" type="actionTag">
  <Metadata>
    <subclassOf value="access"/>
  </Metadata>
</Tag>
CEE Profile: Event Profile

- Defines "event templates"
  - Required & Optional Fields
  - Required Field Values
  - Extensible
Event Profile Example

<EventProfile id="cee_base_event" xml:id="cee_base_event">
  <Description>
    <Text_Title>CEE Base Event Profile</Text_Title>
  </Description>
  <Field ref="time" required="true"/>
  <Field ref="id" required="true"/>
  <Field ref="p_sys_id" required="true"/>
  <Field ref="p_prod_id" required="true"/>
  <Field ref="action" required="true"/>
  <Field ref="status" required="true"/>
  <Field ref="rec_id" required="false"/>
  <Field ref="crit" required="false"/>
  <Field ref="end_time" required="false"/>
  <Field ref="dur" required="false"/>
  <Field ref="tags" required="false"/>
</EventProfile>
CEE Profile Types

- **Base Profile**
  - Defines the base event profile and commonly used fields

- **Function Profile**
  - Defines the event profiles for events associated with a specific function
  - Example: Firewall, Session Management Profile

- **Product Profile**
  - Defines event profiles for events that a specific product may generate
Discussion

1. Do we need more granularity or optional structures in an event profile?
   – Match [FieldSet1] or [FieldSet2]

2. Should event field values be able to be inferred via an event profile?
   – If an event profile specifies a static value in a required field and that field is not present, what does it mean? Non-compliance?
SHARING CEE EVENTS

Common Log Transport (CLT)
CLT Overview

■ CLT Goal
  ■ Provide Technical support necessary for a secure, interoperable, and reliable log infrastructure

■ CLT Requirements Specification
  – Mandatory and optional requirements for log transport protocols

■ CLT Protocol Mappings
  – How to send CLS Encoded CEE Events over certain protocols
  – E.g., Syslog (RFC3164, RFC5424)
CLT Transmission Models
CLT Session Model
CLT Packet Model

- Channel:
  - Packet 1
  - Packet 2
  - Packet 3
  - ...

- Header:
  - Packet ID
  - Timestamp
  - Route Info
  - <metadata>

- Body:
  - <data>
    - CEE Flag
    - CEE Event 1
    - CEE Event 2

- Origination:
  - Hop 1
  - Hop 2
CLT Protocol Requirements

Conformance Level 0 – Core Requirements

– Publish
  ■ published specification with no licensing barriers to interoperability, no royalties, and no approval process

– Transport
  ■ shall be able to transport at least one form of CEE encoded event records within the body of the protocol packet

– Self-Identification
  ■ Identification of CEE Events
  ■ Encoding Identifier

– Time Stamp
CLT Protocol Requirements (2)

- Conformance Level 1 – Basic Capabilities
  - Event Record Delivery
    - preserve integrity of logical order of channel's packets
  - Compression of Records
  - Missing Record detection
  - Transmission Encryption
  - Confidentiality
  - Message Identification
    - Packet Integrity
    - Packet Acknowledgement
Conformance Level 1 – Basic capabilities

- Packet Traversal Traceability
  - capability of tracing and recording the path the packet traverses

- Tamper Detection
  - capability of accurately and reliably detecting evidence of tampering through digital signatures

- Authenticity
  - Use of SASL, GSS-API, and Kerberos
CLT Protocol Requirements (4)

- Conformance Level 2 – Log in Presence of Attackers
  - Full Integrity Acknowledgements
  - Negotiation of Encryption System
  - Message Replay Protection
  - Event Integrity
    - Chain of Modification
    - Reproduction of Original Event
CLT Protocol Requirements (5)

- Conformance Level 3 – Secure Against Local Administration Attacks
  - Tamper Resistant
  - Record Channels
  - Profile Channels
CLT Implementation Requirements

- Conformance Level 0 – Core Requirements
  - Support CLT Protocol Level 0

- Conformance level 1 – Basic Requirements
  - Support CLT Protocol Level 1
  - Sender Side Buffering
    - Single Log Record Buffering
    - Batch log Record Buffering
    - Enable/Disable Switch
CLT Implementation Requirements (2)

- Conformance Level 1 – Basic Requirements
  - Log in Limited Network Environments
    - Retransmission Priority
    - Network Address Translation (NAT)

- Conformance Level 2 – Log in Presence of Attackers
  - Must support at least Conformance Level 2 CTL Protocol
CLT Implementation Requirements (3)

- Conformance Level 3 – Secure Against Local Administrative Attacks
  - Support CLT Protocol level 3
  - Event Source Channel Binding
  - Event Destination Channel Binding
  - Channel Profiles
  - Continuous Operation
CLT Protocol Mapping

- Specification defines how to encode a CEE Event and transmit over a protocol

- CLT Mapping: Syslog
  1. Encode CEE Event using CLS JSON Spec
  2. Add cee: flag
  3. Place in the end of the Syslog message area
CEE-over-Syslog Example

<165>1 2011-04-01T17:01:20Z 10.10.0.1 process -
example-event-1 cee:{"Event":{"id":"example-event-1","time":"t|2011-04-01T17:00:00.123456789Z","action":"g|remove","status":"g|failed","p_sys_id":"host.example.com","p_prod_id":"cpe:2.3:Vendor:Product:Version:*:*:*:*:*:*","file_name":"example.txt","proc_dur":"d|PT.0014S","sess_id":"user1"}}

<0>Apr 4 17:01:20 10.10.0.1 process[35]: cee:{"Event":{"id":"example-event-2","time":"2011-04-01T17:00:00.123456789Z","action":"download","status":"success","p_sys_id":"host.example.com","p_prod_id":"cpe:2.3:Vendor:Product:Version:*:*:*:*:*:*","example_internal_id":10000,"proc_dur":"PT.0014S","sess_id":12345,"file_name":"example.txt","proc_dur":"b|RmlsZSBDb250ZW50Li4uAAo="}}
Discussion

1. Authenticity, Confidentiality, and Packet Integrity are requirements. How would conformance testing be conducted?

2. There should probably be backward compatibility requirements for Sender and Receiver versioning.
WHAT NOW

Where do we go from here
Development

- Software implementations & libraries
- Expand repository
  - More field and tag definitions
  - Validation
  - Add i10n support
- Build more CEE Profiles
  - Common functionalities
  - Profiles for audit requirements: HIPAA, Common Criteria, PCI-DSS
Conformance

- Need vendor/product support
- Compliance program
  - Who supports CEE? Which parts?
  - How can we validate?
  - Can we provide test cases and software libraries to support this?
Discussion

1. Any vendor volunteers to build CEE into their product(s)?
2. Any end user volunteers to begin to integrate CEE into their IT environment?
3. Is anything missing? Is it best suited for inclusion in EMAP or CEE?
BACKUP SLIDES

Additional Content