



Common Event Record Concept

Concept

- Common Event Record (CER)– normalized output from diverse log files and network flow data that enables
 - Basic hooks for correlating diverse event types
 - Coupling sets of events (threads) among peer-layer processes
 - Example: Shibboleth transitions
 - Profiling a complete end-end model of a proper working transaction; composed of multiple, sequenced threads
 - Drop downs (calldowns) in layers to connect threads across multiple subsystems
 - a 323 call denied authz due to improper attribute release policy

Common Event Record

Meta Field Header – what/how/when was the event produced



- **CER Version Number**
- **Observation Description Pointer**
- **ID** – unique event identifier
- **Time** - start/stop
- **IP Address(es)/Type** – source/(destination), IPV4, IPV6, MAC, etc.
- **Source Class** – application, network, system, security, \$USERDEFINED
- **Event Name Tag** – Native language ID (English, French, etc.), user defined
- **Status** – emergency, alert, critical, error, warning, notice, informational, debug
- **Major Source Name** – filename, Netflow, Syslogd, SNMP, shell program, etc.
- **Minor Source Name** – logging process name (named), SNMP variable name, etc.
- **Raw Data Encoding Mechanism** – Binary, ASN1, ASCII, XML, etc.
- **Raw Event Data Description Pointer**

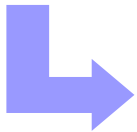
Common Event Record

Observation Description – where was the event seen



- **Observation Description Pointer**
 - Address type of observer (IPV4, IPV6, MAC, etc.)
 - Address of observer
 - Address type of collection agent (IPV4, IPV6, MAC, etc.)
 - Address of collection agent
 - Source Type (file, stream, polled, interrupt)
 - Collection agent name (Netflow.1.0, named.2.3, etc.)

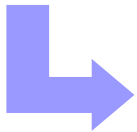
Raw Event Descriptor – how to dissect the event



- **Raw Event Data Description Pointer**
 - Schema of raw event data
 - Pointer to resource to get parser code

Common Event Record

Event Tagging – giving meaning to events



- **Event Name Tag** – (null), user defined (can be multiple tags)
 - Examples:
 - “astronomy-app”
 - “ShibUserHandle=foo”
 - “WebFrontEnd”

To Do

- Use findings of the Pilot (ccBay) to mature concept and design
- Define classification guidelines for event status types
- Examine time reporting method
- Design a processes and structures to decode the raw event data on-the-fly
 - Clearinghouse for structure new event types
 - Creating an event parser repository
- Review requirements for efficiency and scale
- Draft formal specification
- Solicit input from community