

Common Event Record Specification for E2ED Backplane Pilot Project

Version 0.5

Introduction

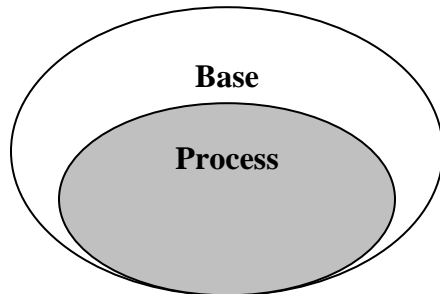
The following is the description of the Common Event Record (CER) designed for the Internet2 Middleware End-To-End Diagnostic Advisory Group Pilot effort to provide a comprehensive format for representing events from the application, network, security and system domains. The goal was to facilitate the correlation of events from these for domains to enable the diagnosis of related faults.

CER Event Type Definitions

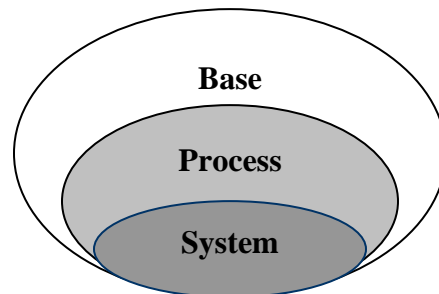
- **Application** – events produced by application processes running that do not pertain to the operation of the host or the OS. Examples are Shibboleth, BIND, SNMPD, HTTPD, etc.
- **System** – events produced by the OS that pertain to itself or the underlying hardware
- **Network** – events that happen on the network. Examples are Netflow data, SNMP traps, etc.
- **Security** – events that are generated by a system/application who's focus is security. Examples are IDS, firewalls, Snort, etc.

Elements of the CER

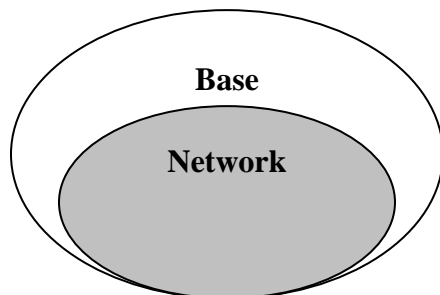
The CER is made up four distinct elements, base, process, system, network, and security. By combining the elements in the following way you can define the four major types of events.



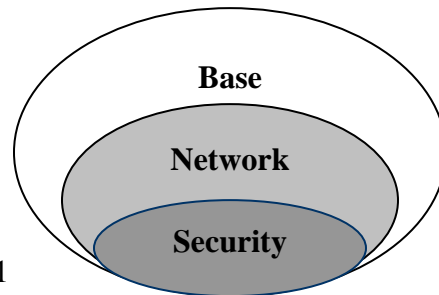
Application Event



System Event



Network Event



Security Event

Hence to further illustrate the event structure,

Application Event = Base Element + Process Element
System Event = Base Element + Process Element + System Element
Network Event = Base Element + Network Element
Security Event = Base Element + Network Element + Security Element

Element Data Structure Definitions

- **Base** - includes data that describes the fundamental relationship between events. The base component encapsulates all other components which make up a CER. The internal data representation of a base component includes,
 - **Version (FLOAT/required)** – version number of the base component data structure
 - **Tag (STRING/optional)** – arbitrary string defined by the collection infrastructure to aid in correlation
 - **TimeStart (TIMESTAMP/required)** – time that the event was observed
 - **TimeStop (TIMESTAMP/optional)** – time that the event has finished. Note most events are recorded from log files and no stop time is recorded.
 - **ServerHostName (FQDN/optional)** – node name where the events are being created and observed.
 - **ServerIP (IP_ADDRESS/required)** - IPv4 address where the events are being created and observed
 - **CollectorHostname (FQDN/optional)** – node name where the events are being collected
 - **CollectorIP (IP_ADDRESS/required)** – Ipv4 address where the events are being collected
 - **CollectorName (TEXT_STRING/required)** – name of the normalization agent that collects the events
 - **CollectorVersion (FLOAT/required)** – version number of the normalization agent that collects the events
 - **WarnLevel (STRING/required)** – metric that describes the severity and type of event.
 - **EMERG** – system is in an unusable state
 - **ALERT** – action must be taken immediately to prevent the system to be unusable within a short period of time
 - **CRIT** – critical conditions, system is operating but is in danger of eventually being unusable
 - **ERR** – error conditions, system is operating but is reporting errors
 - **WARNING** – warning conditions
 - **NOTICE** – normal but significant condition, major changes in state

- **INFO** – informational, notices of changes in configurations or state
- **DEBUG** – debug level events
- **EventMessage (TEXT_STRING/required)** – raw event message collected
- **Sub EventType(s) (ELEMENT/required)** – nested elements that make up a base event that can be one of the following.
 - **ProcessElement**
 - **SystemElement**
 - **NetworkElement**
 - **SecurityElement**

CER Element Data Structure Definitions (ELEMENT structure)

- **ProcessElement** – defines the process that determined and generated the event. This is a physical running process on a host or a network device such as a switch, router or security device such as a firewall or IDS. The internal data representation of a process component includes,
 - **ProcessID (INT/required)** – physical process ID defined by the OS on the device that generated the event
 - **ProcessName (STRING/required)** – physical name of the process defined by the OS on the device that generated the event. Examples of names are processes such as Syslog(8) on Unix based OSs.
 - **ProcessOwner (STRING/required)** – physical owner name defined by the OS on the device that generated
 - **SystemComponent (ELEMENT/optional)** – system component is only included if the event is of type **System**.
- **SystemElement** – defines the low level facility that determines the fundamental events of an OS and the marriage with the underlying hardware
 - Facility (**STRING/Required**) – component of the OS that generates the event. Example: kernel.
 - Subsystem (STRING/Optional) - lower level sub-system that recated the event. Example: raid controller.
- **NetworkElement (NetFlow)** – defines the event that occurs on a network. This could either be generated from a NetFlow ¹, SNMP or RMON type of event. Thus far, only a NetFlow version 5 data structure has been defined.
 - **SourceIP (IP_ADDRESS_STRING/Required)** – IPV4 address of the source of the flow

¹ Cisco defined unidirectional flow measurement and accounting data structure.

- **DestinationIP (IP_ADDRESS/Required)** – IPV4 address of the destination of the flow
 - **NextHopIP (IP_ADDRESS/Optional)** – IPV4 address of the next hop in the flow if known
 - **InterfaceIn (STRING/Optional)** – flow input interface if known
 - **InterfaceOut (STRING/Optional)** – flow output interface if known
 - **Packets (INT/Required)** – total packets in the flow
 - **Octets (INT/Required)** – total octets in the flow
 - **SourcePort (INT/Optional)** – source port of a TCP or UDP flow
 - **DestinationPort (INT/Optional)** – destination port of a TCP or UDP flow
 - **IPTypeOfService (STRING/Optional)** – type of IP service
 - **Protocol (INT/Required)** - IP Protocol number (UDP, TCP, etc.)
 - **TypeOfService (INT/Required)** – type of service
 - **SourceAutonomousSystem (INT/Optional)** – source autonomous system number
 - **DestinationAutonomousSystem (INT/Optional)** - destination autonomous system number
 - **SourceMask (IP_ADDRESS/Optional)** – IPv4 netmask of source flow address
 - **DestinationMask (IP_ADDRESS/Optional)** – IPv4 netmask of destination flow address
- **SecurityElement (Snort)** – defines the event that occurs within the security domain. This could either be generated from Snort², Tripwire or some other type of security baseline or intrusion detection system. Thus far, only a Snort version 5 data structure has been defined.
 - **ReferenceSystem (STRING/Required)** – system that produced the security event (Example: Snort)
 - **TCPHeader (STRING/Optional)** – TCP header of IP packet generating security event
 - **UDPHeader (STRING/Optional)** – UDP header of IP packet generating security event
 - **ICMPHeader (STRING/Optional)** – ICMP header of IP packet generating security event
 - **RawData (STRING//Optional)** - event data

² For more information on Snort, see www.snort.org.

Appendix A

XML representation of the four CER Elements

Base Element

```
<org.Internet2.Middleware.ccBay.Event Version = "0.1">
  <Record>
    <Tag STRING/> <!--Optional -->
    <TimeStart TIMESTAMP/> <!-- Required -->
    <TimeEnd TIMESTAMP/> <!-- Optional -->
    <ServerHostname FQDN/> <!-- Optional -->
    <ServerIP IP_ADDRESS/> <!-- Required -->
    <CollectorHostname FQDN/> <!-- Optional -->
    <CollectorIP IP_ADDRESS/> <!-- Required -->
    <CollectorName STRING/> <!-- Required -->
    <CollectorVersion FLOAT/> <!-- Required -->
    <WarnLevel STRING/> <!-- Required -->
    <EventMessage STRING/> <!-- Required -->
    <org.Internet2.Middleware.ccBay.ProcessEvent/> <!-- Optional -->
    <org.Internet2.Middleware.ccBay.NetworkEvent/> <!-- Optional -->
  </Record>
</org.Internet2.Middleware.ccBay.Event>
```

ProcessElement

```
<org.Internet2.Middleware.ccBay.ProcessElement Version = "0.1">
  <Record>
    <ProcessID INT/> <!-- Required -->
    <ProcessName STRING/> <!-- Required -->
    <ProcessOwner STRING/> <!-- Required -->
    <org.Internet2.Middleware.ccBay.SystemEvent/> <!-- Optional -->
  </Record>
</org.Internet2.Middleware.ccBay.ProcessElement>
```

System Element

```
<org.Internet2.Middleware.ccBay.SystemEvent Version = "0.1">
  <Record>
    <Facility STRING/> <!-- Optional -->
    <Subsystem STRING/> <!-- Optional -->
  </Record>
</org.Internet2.Middleware.ccBay.SystemEvent>
```

Security Element

```
<org.Internet2.Middleware.ccBay.SecurityElement Version = "0.1">
  <Record>
    <ReferenceSystem STRING/> <!-- Required -->
    <TCPHeader INT/> <!-- Optional -->
    <UDPHeader INT/> <!-- Optional -->
    <ICMPHeader INT/> <!-- Optional -->
    <RawData STRING/> <!-- Optional -->
  </Record>
</org.Internet2.Middleware.ccBay.SecurityElement>
```

Network Element

```
<org.Internet2.Middleware.ccBay.NetworkElement Version = "0.1">
  <Record>
    <SourceIP IP_ADDRESS/> <!-- Required -->
    <DestinationIP IP_ADDRESS/> <!-- Required -->
    <NextHopIP IP_ADDRESS/> <!-- Required -->
    <InterfaceIn STRING/> <!-- Required -->
    <InterfaceOut STRING/> <!-- Required -->
    <Packets INT/> <!-- Required -->
    <Octets INT/> <!-- Required -->
    <SourcePort INT/> <!-- Required -->
    <DestinationPort INT/> <!-- Required -->
    <IPTypeOfService STRING/> <!-- Required -->
    <Protocol STRING/> <!-- Required -->
    <TypeOfService STRING/> <!-- Required -->
    <SourceAutonomousSystem STRING/> <!-- Required -->
    <DestinationAutonomousSystem STRING/> <!-- Required -->
    <SourceMask OCTAL/> <!-- Required -->
    <DestinationMask OCTAL/> <!-- Required -->
    <org.Internet2.Middleware.ccBay.SecurityEvent/> <!-- Optional -->
  </Record>
</org.Internet2.Middleware.ccBay.NetworkElement>
```

Appendix B

Event Element Content Types

IP_ADDRESS:

- IPADDRESS.IPADDRESS.IPADDRESS.IPADDRESS

STRING:

- Any printable ASCII character

INT:

- Up to 16 characters in the range of 0-9

FLOAT:

- INT.INT

IPADDRESSOCTET:

- The range of a number string from 0 to 255

TIMESTAMP:

- The format of

FQDN:

- Fully qualified domain name
- HOSTNAME.DOMAIN

DOMAIN

- STRING

HOSTNAME

- [STRING.[STRING[.STRING.[STRING.[STRING.]]]]STRING.