eCSIRT.net Deliverable[1] Common Language Specification & Guideline to Application of the Common Language part (i)

Deliverable D2.1 & Deliverable D2.3.2/D3.2.1/D3.2.3 part (i)[2]

Arne Helme
Stelvio
the Netherlands

Version 2.3, December 2003

[1] This work is funded by the European Commision's Fifth Framework Program, within the IST (Information Society Technology) branch, contract no. IST-2001-37558.

[2] Part (ii) of this deliverable is contained in a separate document, named "Guideline to Application of the Common Language part (ii) — an outline of the eCSIRT.net toolkit"

Abstract

This document is the main deliverable for the WP2 Preparation Phase. WP2 is concerned with the definition of a common language for the format for messages being exchanged between computer security incident response teams. Both requirements to the common language and a specification of its syntax and semantics are contained in this document. The document is so specific, that in fact it has also been adopted during the program as part (i) of the Guideline to Application (part (ii) being the description of the toolkit resulting from eCSIRT.net: IHSH and the Perl Library). Hence, it has been adapted during the whole course of the project to include new developments and insight. This included small changes in the common language itself, since the underlying standard (IODEF datamodel) was not entirely stable yet (in details), and since the eCSIRT.net testing also made changes to the defined syntax necessary.

Contents

1 Overview

In order to exchange incident handling information unambiguously between CSIRTs a common language is required. For the common language both a notational syntax and semantics of its content must be agreed upon. This document describes eCSIRT.net's Work Package 2 (WP2): Common language.

2 Common Language Requirements

For the eCSIRT.net consortium, efficient coordination of CSIRTs is the main focus of the efforts. By this we mean coordination of CSIRT activities through the electronic exchange of incident handling information between TI-accredited CSIRTs.

Messages being exchanged between the teams are human readable, but defined as a wire format. It is the responsibility of each team to translate the message to a local database format that is suitable for the incident tracking system used by each team.

Furthermore, the messages are machine readable, and thereby departing from the current best practice of exchanging advisories in ASCII between the teams by the means of electronic mail. Due to the nature of the problem being investigated, XML is viewed as fundamental building stone both to provide the document structure and multi-lingual capabilities.

3 Operational Framework

In order to contain and prevent security incidents, CSIRTs need to have a common understanding of such incidents and agreed-on operational procedures of exchanging and handling information.

Operational frameworks for CSIRTs are well documented in literature (see, e.g., the CSIRT handbook [West-Brown et al., 1998]). The definition of a computer security incident response team used in this paper is the one devised by [Koek et al., 2001]:

> If an entity A entertains a function B where customers/constituencies can report computer/network security incidents, and B then handles these reports in a constructive and secure way (consulting, coordination, feedback, ...), then function B essentially is the CSIRT of entity A.

For eCSIRT.net purposes a certain similarity in purpose and operation of the participating CSIRTs is necessary, for the exchange of incident data to be successful and meaningful. This necessary similarity is ensured by only allowing teams in that are TI accredited. TI accreditation is seen as sufficient condition - a necessary condition has not been defined yet.

Additionally eCSIRT.net partners will have to sign a “Code of Conduct” before embarking on standardized data exchange — which together with TI accreditation further defines the eCSIRT.net operational framework.

4 Common Language Syntax

Syntax for the messages to be exchanged between CSIRTs participating in eCSIRT.net is derived from work already in progress by the IETF Working Group on Extended Incident Handling (INCH), in particular, the Incident Object Description and Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition (IODEF) [Meijer et al., 2002], Incident Object Description and Exchange Format Requirements [Demchenko, 2002] and Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition (IDMEF) [Curry and Debar, 2003].

At the time of writing, the IETF INCH Working Group is addressing the need for a message format for exchanging incident related information between CSIRTs and is still work in progress. The eCSIRT.net consortium is actively participating in INCH and the standardization efforts to develop incident handling message exchange formats.

Having defined a syntax for the exchange of incident handling information between teams is clearly not sufficient in order for this information to be taken up and used successfully. The syntax defined in IODEF contains too much flexibility to be used as is without putting more restrictions on the use of it. And of course semantics are needed as well: a language with only syntax is a language nobody will understand.

5 Common Language Semantics

Semantics of the messages is partially derived from, and limited to the semantics already implicitly defined in IODEF and the requirements of applying the IODEF exchange format. In addition, the eCSIRT.net project has further developed semantics to address the problems studied by the project within the context of efficient cooperation of CSIRTs. More specifically, in the semantics enforced by the project, there is a clear separation between information used to handle the incident in question and the event descriptors used to describe organizational and/or technical details of the events associated with the incident itself.

Messages to be exchanged by the teams are encoded as XML documents while in transit, using the IODEF syntax notation. The semantics and format of each report are layered into the following three levels:

Top level data: At the top level, each message contains information about the purpose (application) of the message, the sender of the message, identifying information for the message, and information about past actions made with regard to the incident in question. The purpose of the message is currently either:

- incident handling,
- incident alerting, or
- statistics.

Each CSIRT team is identified by a global unique identifier[1] and each incident, translated to an IODEF message, is identified by the team's own tracking ID of that incident, which preferably is a concatenation of the team's identifier with some structured or un-structured string (e.g., a number). In addition, each IODEF message can be tagged with the tracking IDs of other teams' registration of the same incident, if known. Past actions are recorded in a history section so other teams can see what actions that have already been taken with regard to the incident at hand.

Incident level data: At the incident level, each message contains information relevant to incident handling by the receiving CSIRT. Mainly this information consist of impact estimators and expectations that the sending CSIRT has of actions the receiving CSIRT should take with regard to the incident at hand. In addition, information is conveyed to describe past actions already taken by other CSIRTs.

Important is this notion of expectation: conveying information to other teams as to how they are expected to react. Recognizing the fact that each team operates autonomously and make their own decisions on how to react to received messages. "Expectation" on other teams clearly is not part of a team's incident database — it used to be typed in by hand when sending an e-mail to another team describing an incident. This typing-in-by-hand has now been replaced by the the "expectation" data-structure.

[1] The TI database solves the problem of "global unique identifiers" for CSIRTs in Europe, but this framework would need extension to scale globally.

Event level data: At the event level, each message contains information about "events" of relevance to the incident at hand. "Event" has to be understood in the broadest meaning of the word, not necessarily as something that happened in the course of the incident. E.g. the victim system is described as an "event" in the model, but so can contact data. A flexible and nested information syntax allows for complex structures of event information that is used to describe both informational and technical data of the incident.

Event data can be nested to allow for complex combinations of event data. In particular, it is possible to use the flexibility of IODEF to zoom in on levels of detail and group information together to describe different kinds of combinations of event data.

To achieve unambiguity in event data structures, within eCSIRT.net profiles are developed for classes of incidents according to existing taxonomies for computer security incidents. In this context, a profile must be seen as a further restriction on how IODEF is being applied when an incident is actually described on the semantic level.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in the remainder of this document are to be interpreted as described in RFC 2119 [Bradner, 1997].

5.1 eCSIRT.net IODEF Profile

For communication between CSIRTs in eCSIRT.net not all of the classes defined in the IODEF specification will be used. However, some of the classes are MANDATORY,[2] and specified in the eCSIRT.net IODEF profile below:

```
<IODEF-Document version="">
  <Incident purpose="" restriction="">
    <IncidentID name=""> </IncidentID>

    <AlternativeIDs> </AlternativeIDs>

    <IncidentData>
      <Description> </Description>

      <Assessment>
```

[2] eCSIRT.net compliant implementations should support all classes of the profile enev though some optional classes may be omitted in certain message exchanges.

```
  <Impact completion="" type=""> </Impact>
</Assessment>

<Method>
  <Description> </Description>
</Method>

<Expectation priority="">
  <Description> </Description>
</Expectation>

<ReportTime> </ReportTime>

<Contact role="" type="">
  <name> </name>
  <Email> </Email>
  <Telephone> </Telephone>
  <Timezone> </Timezone>
</Contact>

<History>
  <HistoryItem type="">
    <Description> </Description>
    <DateTime> </DateTime>
  </HistoryItem>
</History>

<EventData>
  <Description> </Description>
  <System category="">
    <Node>
      <Address category=""> </Address>
    </Node>
  </System>
</EventData>

<EventData>
  <Description> </Description>
  <Record>
    <RecordData>
      <DateTime> </DateTime>
      <RecordItem type=""> </RecordItem>
    </RecordData>
  </Record>
```

```
        </EventData>
      </IncidentData>

      <AdditionalData type="" meaning=""> </AdditionalData>
    </Incident>
  </IODEF-Document>
```

5.2 IODEF-Document

The IODEF-Document class is the outer container class for IODEF messages:

```
<IODEF-Document version="">
.
.
.
</IODEF-Document>
```

The `version` attribute SHALL denote the version of the IODEF specification to which the IODEF document conforms. The value of this attribute SHALL be `1.0`.

5.3 Incident

The Incident class provides a standardized representation for commonly exchanged incident data and associates a unique identifier with the described activity:

```
<Incident purpose="" restriction="">
.
.
.
</Incident>
```

The `purpose` attribute is MANDATORY and SHALL be set to either `handling`, `alerting`, or `statistics` depending on whether the message contains incident handling information, contains an alert, or contains statistical information. The `restriction` attribute is also MANDATORY and SHALL be set to the default restriction level placed on the entire document. Supported values are `public`, `need-to-know`, and `private`.

5.4 IncidentID

The IncidentID class represents the incident tracking number or identifier used by a CSIRT or reporter to uniquely identify (in their organization) the activity characterized in this IODEF document:

```
<IncidentID name=""> </IncidentID>
```

The IncidentID class is MANDATORY and SHALL contain a global unique identifier for the incident that has been assigned to the incident by the sending CSIRT. The `name` attribute SHALL contain a unique CSIRT team identity.

Each CSIRT team is identified by a global unique identifier[3] and each incident, translated to an IODEF message, is identified by the team's own tracking ID of that incident, which preferably is a concatenation of the team's identifier with some structured or un-structured string (e.g., a number). IncidentIDs are encoded as the CSIRT unique identifier and the incident identifier joined by by the sign '-'.

5.5 AlternativeIDs

The AlternativeIDs class contain references to other incidents related to the current one. AlternativeIDs elements are encoded in the same way as IncidentID mentioned above.

```
<AlternativeIDs>
.
.
.
</AlternativeIDs>
```

The AlternativeIDs class is MANDATORY. Each IODEF message MAY be tagged with the tracking IDs of other teams' registration of the same incident, if known. Semantics of AlternativeIDs is the same as for IncidentID.

5.6 IncidentData

At the incident level, each message contains information relevant to incident handling by the receiving CSIRT. Mainly this information consist of impact estimators and expectations that the sending CSIRT has to actions

[3] The TI database solves the problem of "global unique identifiers" for CSIRTs in Europe, but this framework would need extension to scale globally.

the receiving CSIRT should take with regard to the incident at hand. In addition, information is conveyed to describe past actions already taken by other CSIRTs.

The IncidentData class summarizes the details of the incident activity and a CSIRT's handling of the information, as well as, groups the security events that constitute the incident:

```
<IncidentData>
  <Description> </Description>
.
.
.
</IncidentData>
```

The IncidentData class is MANDATORY and SHALL contain incident related information. The semantics of the IncidentData class is described in detail below. The Description class is MANDATORY and SHALL contain a textual description of the type of incident.

5.6.1 Assessment

The Assessment class describes the technical and non-technical repercussions of the incident activity:

```
<Assessment>
  <Impact completion="" type=""> </Impact>
</Assessment>
```

The Assessment class is used to provide the CSIRT's assessment of an event. The Assessment class is MANDATORY and SHALL contain a an assessment of the incident.

The Assessment SHALL contain an Impact class. The Impact class allows for classifying as well as providing a description of the technical impact due to the incident activity on the computers and networks of an organization. MANDATORY is the use of the `completion` and `type` attributes. The completion attribute SHALL be set to either `failed` or `succeeded`. The type attribute SHALL be set according the type specification in IODEF.

5.6.2 Method

The Method class provides information about the methodology used by the intruder to perpetrate the events of the incident:

```
<Method>
  <Description> </Description>
</Method>
```

The Method class is MANDATORY and SHALL contain a Description class with a textual description of the actions taken by the attacker in the incident.

5.6.3 Expectation

The Expectation class conveys to the recipient of the message the actions the sender is requesting:

```
<Expectation priority="">
  <Description> </Description>
</Expectation>
```

The Expectation class is MANDATORY and REQUIRED is to use an priority attribute to indicates the desired priority of the action. The priority attribute SHALL be set to either `low`, `medium` or `high`. The Expectation class SHALL contain a Description class with the textual description of the expectation.

Important is this notion of expectation: conveying information to other teams as to how they are expected to react. Recognizing the fact that each team operates autonomously and make their own decisions on how to react to received messages. Expectation on other teams clearly is not part of a team's incident database — it used to be typed in by hand when sending an e-mail to another team describing an incident. This typing-in-by-hand has now been replaced by the the Expectation data-structure.

5.6.4 ReportTime

The ReportTime class represents the time stamp of when a detected activity was reported:

```
<ReportTime> </ReportTime>
```

The ReportTime class is MANDATORY.

5.6.5 Contact

The Contact class describes contact information for organizations and personnel involved in the incident. This class encapsulates naming the involved party, specifying contact information to reach them, and identifying their role in the incident.

```
<Contact role="" type="">
  <name> </name>
  <Email> </Email>
  <Telephone> </Telephone>
  <Timezone> </Timezone>
</Contact>
```

The Contact class is MANDATORY and SHALL contain contact information for the organization handling the incident. REQUIRED is to use the `role` and `type` attributes. The role attribute SHALL be set to `irt` and the type attribute SHALL be set to `organization`.

The Contact class SHALL contain at least the following classes to describe the contact: `name`, `Email`, `Telephone`, and `Timezone`.

5.6.6 History

Past actions are recorded in a history section so other teams can see what actions that have already been taken with regard to the incident at hand. The History class is a log or diary of the significant events that occurred or actions performed by the involved parties (e.g., initial reporter, investigating CSIRT, or involved system administrators) during the course of handling the incident:

```
<History>
  <HistoryItem type="">
     <Description> </Description>
    <DateTime> </DateTime>
  </HistoryItem>
</History>
```

The History class is MANDATORY and MAY contain a HistoryItem class. REQUIRED is to use the `type` attribute to classify the type of activity or event being document in this history log entry. Currently supported types are those defined in IODEF. In Addition, each HistoryItem entry SHALL contain a Description and a DateTime class item to show the timestamp of the this entry in the history log (e.g., when the action described in the Description was taken).

5.6.7 EventData

At the event level, each message contains information about "events" of relevance to the incident at hand. "Event" has to be understood in the broadest meaning of the word, not necessarily as something that happened in the course of the incident. E.g. the victim system is described as an “event” in the model, but so can contact data. A flexible and nested information syntax allows for complex structures of event information that is used to describe both informational and technical data of the incident.

Event data can be nested to allow for complex combinations of event data. In particular, it is possible to use the flexibility of IODEF to zoom in on levels of detail and group information together to describe different kinds of combinations of event data. To achieve unambiguity in event data structures, within eCSIRT.net a profile has been developed. In this context, a profile must be seen as a further restriction on how IODEF is being applied when an incident is actually described on the semantic level.

The EventData class is MANDATORY and SHALL contain a Description class with a textual description of the event or report of the incident events. In addition, the EventData class SHALL contain either a System class or a Record class to describe events of the incident. The semantics of the EventData class is described in detail below.

To exchange network addresses the System class is REQUIRED:

```
<EventData>
  <Description> </Description>
  <System category="">
    <Node>
      <Address category=""> </Address>
    </Node>
  </System>
</EventData>
```

REQUIRED is the use of the `category` attribute. The category attribute SHALL be set to either `source`, `target`, or `intermediate`. The System class SHALL contain a Node class. The Node Class SHALL contain an Address class to represents a network, hardware, or application address. REQUIRED is the use of a `category` attribute to denote the address category. Supported values for the attribute are those provided by IODEF.

To exchange forensic information the Record class is REQUIRED:

```
<EventData>
  <Description> </Description>
  <Record>
    <RecordData>
      <DateTime> </DateTime>
      <RecordItem type=""> </RecordItem>
    </RecordData>
  </Record>
</EventData>
```

The Record class SHALL contain a RecordData class item. REQUIRED is the use of the `type` attribute. To facilitate arbitrary data types, the type attribute SHALL be set to `string` to denote a text string, or `file` to denote a base64 encoded binary file. REQUIRED is also to use a DateTime class item to hold timestamp information for the RecordItem data.

5.7 AdditionalData

The AdditionalData class serves as an extension mechanism for information not otherwise represented in the IODEF data model:

```
<AdditionalData type="" meaning="">
.
.
.
</AdditionalData>
```

MANDATORY is to include one AdditionalData class item to specify the eCSIRT.net IODEF profile version. The `type` attribute SHALL be set to `string`. The `meaning` attribute SHALL contain the string "eCSIRT.net IODEF Profile Version 1.0". The AdditionalData shall be set to *eCSIRT-net-IODEF-profile-v1.0.*

5.8 Transport Mechanisms

MANDATORY transport mechanism is electronic email. Incident handling information formatted according to the specification in this document SHALL be sent as MIME style attachments to emails exchanged between the CSIRTs. More specifically, each team SHALL support the following email addresses:

- Email-Account for IODEF-statistics: *iodef-stats@ecsirt.net.*
- Email-Account for Alerting: *iodef-alerts@ecsirt.net.*
- Email-Accounts for each team to send and receive incident handling information: *iodef-in@team.domain* and *iodef-return@team.domain.*

6 Implementation Issues

The ease implementation and facilitate a smooth integration of the Common Language into the workflow handling systems of the participating CSIRTs, a simple approach has been chosen. The essence of this approach is to create profiles that models the minimum amount of information that is required in order to enable inter-team communication using IODEF. In practice this means that the information used today is modeled as simple as possible in order to ensure a tight mapping between the existing formats for email exchange, and information exchanged using the Common Language. As a consequence, exotic features of IODEF are not utilized. However, teams should endorse conforming applications that contains other classes than those specified in the eCSIRT.net IODEF profile.

7 Security Considerations

Though swift and efficient information exchange is the focus of the project, security requirements can obviously not be neglected. More specifically, establishing the origin and authenticity of received incident handling messages, verifying the integrity of their contents, and in some cases ensuring the confidentiality of the information while being in transit between the teams can be important for CSIRTs to authorize taking due actions. Service denial is, of course, also relevant, but outside the scope of the project and this paper.

Some of the issues just mentioned can be solved by tweaking the underlying communication mechanisms (e.g. building a secure web of mail servers). By nature however most of the solution to this security problem will have to be found in establishing end-to-end security. Helpful factors are the relative trust found between TI-accredited teams, and the existence of a public-key infrastructure based on PGP, and supported by the TI. For the future, the project also expects that XML signatures will be deployed to guarantee the authenticity of origin and integrity of content.

References

[Bradner, 1997] Bradner, S. (1997). Key words for use in RFCs to Indicate Requirement Levels. RFC 2119.

[Curry and Debar, 2003] Curry, D. and Debar, H. (2003). Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition (IDMEF). IETF Internet-Draft draft-ietf-idwg-idmef-xml-10.txt.

[Demchenko, 2002] Demchenko, Y. (2002). Incident Object Description and Exchange Format Requirements. IETF Internet-Draft draft-ietf-inch-iodef-rfc3067bis-requirements-00.txt.

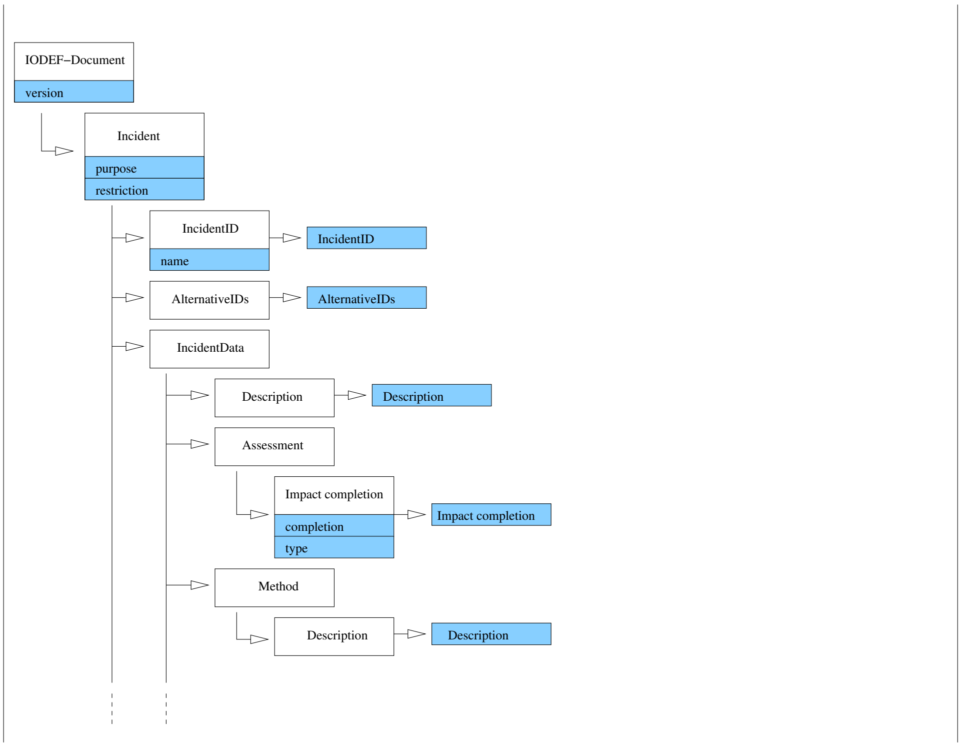
[Koek et al., 2001] Koek, M., Smits, E., Stikvoort, D., and Kossakowski, K.-P. (2001). The Trusted Introducer Service. In *FIRST Conference on Computer Security Incident Handling & Response 2001*, Toulouse, France.

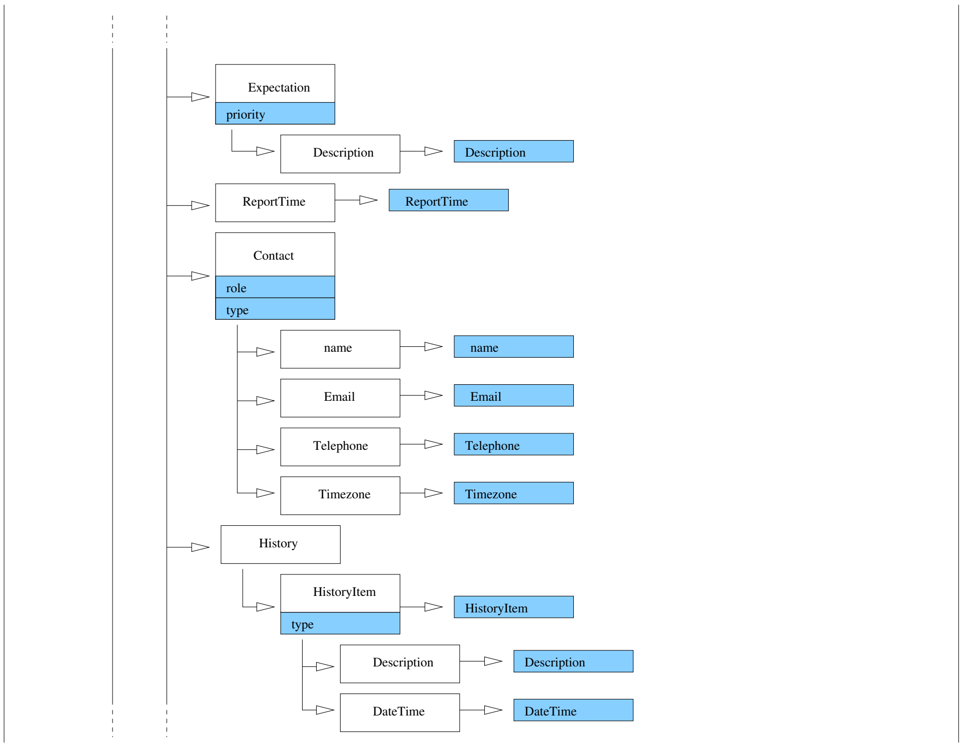
[Meijer et al., 2002] Meijer, J. J., Danyliw, R., and Demchenko, Y. (2002). Incident Object Description and Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition (IODEF). IETF Internet-Draft draft-ietf-inch-iodef-00.txt.

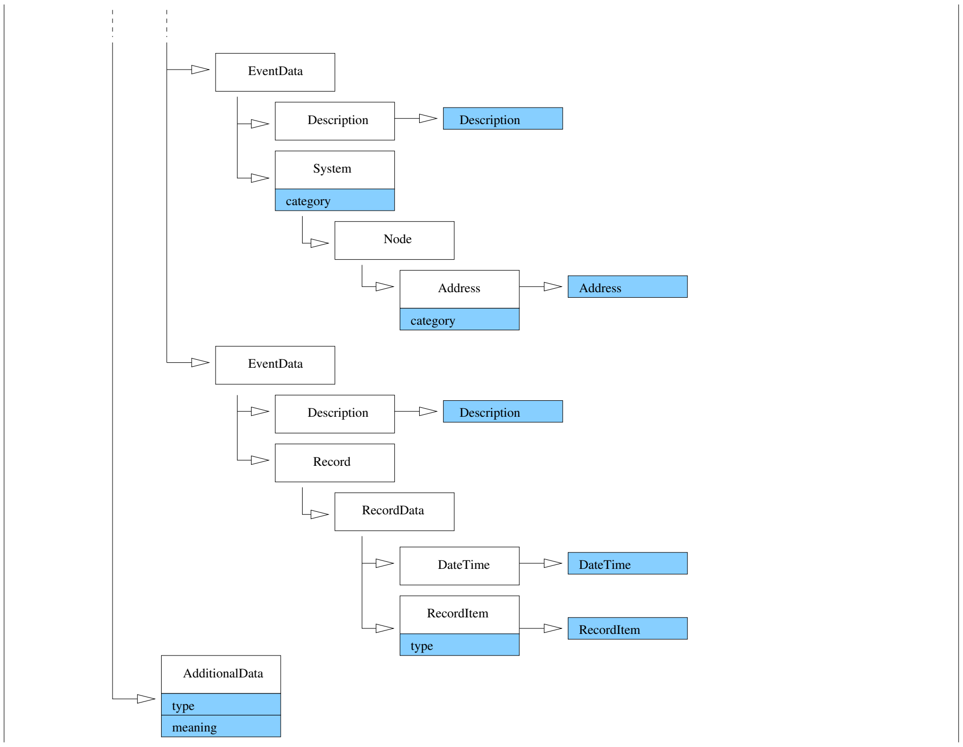
[West-Brown et al., 1998] West-Brown, M. J., Stikvoort, D., and Kossakowski, K.-P. (1998). Handbook for Computer Security Incident Response Teams (CSIRTs). Technical Report CMU/SEI-98-HB-001, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

A Illustrations: Common Language

The following illustration show the eCSIRT.net Common Language IODEF profile in detail:




Expectation
priority
Description
Description
ReportTime
ReportTime
Contact
role
type
name
name
Email
Email
Telephone
Telephone
Timezone
Timezone
History
HistoryItem
type
HistoryItem
Description
Description
DateTime
DateTime


EventData
Description
Description
System
category
Node
Address
category
Address
EventData
Description
Description
Record
RecordData
DateTime
DateTime
RecordItem
type
RecordItem
AdditionalData
type
meaning

B Example: eCSIRT.net IODEF document

The following is an example document conforming to the eCSIRT.net IODEF profile:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE IODEF-Document>

<IODEF-Document version="1.0">
  <Incident restriction="need-to-know" purpose="handling">
    <IncidentID name="CERT-NL">CERT-NL-42353465345</IncidentID>
    <AlternativeIDs>
      <IncidentID name="CERT-CC">CERT-CC-42353465345</IncidentID>
    </AlternativeIDs>
    <IncidentData>
      <Description>Portscan report</Description>
      <Contact role="irt" type="organization">
        <name>CERT-NL</name>
        <Email>cert-nl@surfnet.nl</Email>
        <Telephone>+31622923564</Telephone>
      </Contact>
      <ReportTime>200305201453</ReportTime>
      <Expectation priority="low">
        <Description>Take action and report back</Description>
      </Expectation>
      <Method>
        <Description>Unknown</Description>
      </Method>
      <Assessment>
        <Impact completion="failed" type="recon">
          Low impact, not completed
        </Impact>
      </Assessment>
      <EventData>
        <Description>Source IP address</Description>
        <System category="source">
          <Node>
            <Address category="ipv4-addr">
              <address>129.242.16.25</address>
            </Address>
          </Node>
        </System>
      </EventData>
    </IncidentData>
    <AdditionalData
      type="string"
      meaning="eCSIRT.net IODEF Profile Version 1.0">
      eCSIRT-net-IODEF-profile-v1.0
    </AdditionalData>
  </Incident>
</IODEF-Document>
```