The purpose of the Incident Data Exchange Format (IODEF) is to define data formats for information related to computer security incidents typically exchanged between collaborating Computer Security Incident Response Teams (CSIRTs). The IODEF satisfies the requirements specified in RFCXXX [1]

This Internet-Draft describes a data model for representing commonly exchanged incident information exported from incident handling systems managed by CSIRTs. An implementation of the data model in
the Extensible Markup Language (XML) is presented, an XML Document Type Definition is developed, and examples are provided.

Table of Contents

1. Introduction ........................................... 4
   1.1 Terminology ........................................ 4
   1.2 Overview ............................................ 4
   1.3 About the IODEF Data Model ......................... 4
   1.3.1 Issues with Representing Incident Data .......... 5
   1.4 About the IODEF Implementation ..................... 6
   1.5 Related Work ........................................ 6
2. Notational conventions and formatting issues .......... 8
   2.1 IODEF XML Documents ............................... 8
   2.1.1 The Document Prolog ............................. 8
   2.1.2 Character Data Processing in the IODEF .......... 9
   2.1.3 Languages in the IODEF .......................... 10
   2.2 IODEF Data Types .................................. 11
   2.2.1 Integers .......................................... 11
   2.2.2 Real Numbers ...................................... 11
   2.2.3 Characters and Strings ........................... 11
   2.2.4 Bytes ............................................. 12
   2.2.5Enumerated Types ................................ 12
   2.2.6 Date-Time Strings ................................ 12
   2.2.7 NTP Timestamps .................................. 12
   2.2.8 Port Lists ........................................ 12
   2.2.9 Postal Address .................................... 12
   2.2.10 Person or Organization ........................... 13
   2.2.11 Telephone and Fax Numbers ....................... 13
   2.2.12Email string ...................................... 13
   2.2.13 Uniform Resource Identifier strings ............ 13
   2.2.14 Unique Identifiers ............................... 13
3. The IODEF Data Model .................................. 14
   3.1 IODEF-Document class ................................ 14
   3.2 Incident class ....................................... 14
   3.3 IncidentID class .................................... 17
   3.4 AlternativeID class ................................ 17
   3.5 RelatedActivity class ................................ 18
   3.6 AdditionalData ....................................... 19
   3.7 IncidentData ......................................... 20
   3.8 Contact class ........................................ 22
   3.8.1 RegistryHandle class .............................. 25
1. Introduction

1.1 Terminology

The key words "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [5].

Definitions for some of the common computer security-related terminology used in this document can be found in Section 2 of [1].

1.2 Overview

The Incident Data Exchange Format (IODEF) is intended to be a standard format for computer security information exchanged by Computer Security Incident Response Teams (CSIRTs). The development and subsequent deployment of an incident data format that extends beyond a closed communities would improve the operational capabilities of the CSIRTs.

Assuming widespread adoption of the IODEF by the community, an organization can potentially benefit from:

- the increased ease to collaborate with other CSIRTs, on behalf it its constituency, to resolve incidents;
- increased automation in the processing of incident data, since the commitment of security analysts to parse free-form textual
document will be reduced;

- decreased effort in normalizing similar data (even when highly structured) from different sources; and

- a common format on which to build inter-operable tools for incident handling, such as correlation systems that process data from different sites.

Terminology, notation, and conventions of the data model and XML DTD are presented in Sections 2. The data model is described in Section 3, and the implementation considerations are covered in Sections 4 through 6, and 9. Section 7 provides several examples of IODEF documents for representative incidents. Section 8 formally specifies the XML DTD implementation of the data model.

1.3 About the IODEF Data Model

The IODEF data model is an object-oriented representation of information reported, maintained, and exchanged by a CSIRT about a computer security incident.

1.3.1 Issues with Representing Incident Data

The IODEF data model addresses several problems in representing incident data:

- There is no precise, widely agreed upon definition for an incident. Therefore, the data model does not attempt to imply a definition through its implementation. Rather, a broad understanding is assumed that is flexible enough to encompass most of the CSIRT community.

- Incident data is inherently heterogeneous. It may encompass many functional purposes such as a description of intruder behavior or an analysis process correlating related incidents. An object-oriented model provides extensibility via aggregation and sub-classing while preserving the consistency of the model. If the data model required modification, it is extended with new classes. In implementations that do not recognize these extensions, the basic subset of the data model will still be
understood.

- Incidents have a life-cycle, which causes potentially different information or levels of detail to be present depending on their stage in the cycle. For example, newly reported incidents may only contain a short description of the involved parties. On the other hand, closed incidents can contain a full description complete with the associated evidence and annotation of actions taken by the CSIRT. The data model that represents this information must be flexible to accommodate different needs.

- Communication and coordination are central to the role of a CSIRT. As a result of this activity, incident information can originate from a number of sources. Tracking all the sources of data is key to managing this information. The data model defines support classes that accommodate the differences between incident reporters. This support includes various meta information to represent the reporter's identity as well as prescribe a confidence level to the submitted information.

- Incident data may contain sensitive information. Such information should not be exposed to unauthorized parties during collaboration. The data model allows for a granular tagging in the individual classes to indication restrictions on the usage of the data. However, it is the role of the incident handling system implementing the data model to honor these labels.

---


1.4 About the IODEF Implementation

The IODEF implementation uses the Extensible Markup Language (XML) [2]. XML-based specifications define an XML DTD or Schema and register a specific XML namespace [3]. The IODEF conforms to the IETF-defined procedure for registering an application-specific XML namespace [9].

For clarity in this document, we will use the terms "XML" and "XML documents" when speaking in the general case about the Extensible Markup Language (XML). The terms "IODEF description", "IODEF markup" and "IODEF document" will be used to refer to specific elements (tags) and attributes of the IODEF DTD. Furthermore, the terms "class" and "subclass" are synonymous to an element in the
The implementation of the IODEF in XML has many benefits:

- XML provides all the necessary features to define a specific markup language for describing security incidents. It also defines a standard way to extend this language, either for later revisions ("standard" extensions), or for organizational-specific use ("non-standard" extensions).
- Software tools for processing XML documents are widely available in commercial and open source forms.
- XML can aid in implementing internationalization and localization since it is required (and therefore IODEF documents are required) to support both the UTF-8 and UTF-16 encodings of ISO/IEC 10646 (Universal Multiple-Octet Coded Character Set, "UCS") and Unicode. XML also provides support for specifying, on a per-element basis, the language in which the element's content is written, making the IODEF easy to adapt to the local languages in which a CSIRT operates.
- XML coupled with XSL [4], a style language, allows IODEF documents to be aggregated, filtered, discarded, and rearranged.
- XML is free (no license, license fees or royalties).

1.5 Related Work

The IODEF and the IDMEF [7] are complementary formats. The latter represents data generated by an intrusion detection system. Such event data is commonly used by a CSIRT as the basis for an incident report or investigation which is represented by the IODEF.

The IODEF data model makes use of certain classes defined in the IDMEF, although the semantics of some of these classes has changed. Due to their related nature, the data in an IDMEF message can be easily represented in an IODEF document. Through various extension mechanisms, it is possible to include IDMEF messages outright in an IODEF document. Alternatively, the similarity in structure of the data model makes it possible to decompose the key IDMEF data and
include it in the corresponding IODEF classes. However, this transformation may not preserve the original semantics of the data.
2. Notational conventions and formatting issues

2.1 IODEF XML Documents

This document uses three notations: the Unified Modeling Language (UML) to describe the data model, an Extensible Markup Language (XML) Document Type Definition (DTD) to define the IODEF syntax, and IODEF XML markup conforming to the specified DTD to represent the incident data.

This section describes the XML notations and conventions used in this memo and explains particular issues related to using them to describe the IODEF data model and syntax. For readers unfamiliar with these notations [19] and [7] will provide a comprehensive reference.

2.1.1 The Document Prolog

The "prolog" of an XML document, that part that precedes anything else, consists of the XML declaration and the document type declaration.

2.1.1.1 XML Declaration

Every IODEF document starts with an XML declaration. The XML declaration specifies the version of XML being used, and optionally the character encoding being used (see Section 2.1.2).

The XML declaration looks like:

```xml
<?xml version="1.0" ?>
```

IODEF documents exchanged between applications MUST begin with an XML declaration and MUST specify the XML version in use. Specification of the encoding in use is REQUIRED if UTF-8 encoding is not used.

2.1.1.2 IODEF DTD Formal Public Identifier

The formal public identifier (FPI) for the IODEF Document Type Definition described in this document is:

```
"-//IETF//DTD RFCxxxx IODEF v0.0//EN"
```

NOTE: The "RFCxxxx" text in the FPI value will be replaced with the actual RFC number when this document is published as an RFC.

This FPI MUST be used in the document type declaration within an XML document referencing the IODEF DTD defined by this document, as shown in Section 2.1.1.3.
2.1.3 IODEF DTD Document Type Declaration

The document type declaration for an XML document referencing the IODEF DTD will be specified in the following ways:

```xml
<!DOCTYPE IODEF-Document PUBLIC "-//IETF//DTD RFCxxxx IODEF v0.0//EN">
```

The last component of the document type declaration is the FPI specified in Section 2.1.1.2.

```xml
<!DOCTYPE IODEF-Document SYSTEM "/path/to/IODEF-Document.dtd">
```

The last component of the document type declaration is a URI that points to a copy of the Document Type Definition.

2.1.2 Character Data Processing in the IODEF

A document's XML declaration specifies the character encoding to be used in the document, as follows:

```xml
<?xml version="1.0" encoding="charset" ?>
```

where "charset" is the name of the character encoding, as registered with the Internet Assigned Numbers Authority (IANA), see [9].

The XML standard requires that XML processors support the UTF-8 and UTF-16 encodings of ISO/IEC 10646 (UCS) and Unicode, making all XML applications (and therefore, all IODEF-compliant applications) compatible with these common character encodings.

While XML supports other character encodings (e.g., UTF-7, UTF-32), implementers should carefully consider the portability implications of using character encodings other than UTF-8 and UTF-16.

Consistent with the XML standard, if no encoding is specified for an IODEF document, UTF-8 is assumed. IODEF documents encoded in UTF-16 MUST begin with the Byte Order Mark described by ISO/IEC 10646 Annex E and Unicode Appendix B (the "ZERO WIDTH NO-BREAK SPACE" character, #xFEFF).

2.1.2.1 Character Entity References

Within XML documents, certain characters have special meanings in
some contexts. To include the actual character itself in one of these contexts, a special escape sequence, called an entity reference, must be used.

The characters that sometimes need to be escaped, and their entity references, are:

<table>
<thead>
<tr>
<th>Character</th>
<th>Entity Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;</td>
<td>&amp;</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
</tr>
<tr>
<td>&gt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>'</td>
<td>'</td>
</tr>
</tbody>
</table>

Figure 1

It is RECOMMENDED that IODEF-compliant applications use the entity reference form whenever writing these characters in data, to avoid any possibility of misinterpretation.

2.1.2.2 Character Code References

Any character defined by the ISO/IEC 10646 and Unicode standards may be included in an XML document by the use of a character reference. A character reference is started with the characters '&', and '#' and ended with the character ';'. Between these characters, the character code for the character inserted.

If the character code is preceded by an 'x' it is interpreted in hexadecimal (base 16), otherwise, it is interpreted in decimal (base 10). For instance, the ampersand (&) is encoded as '&#38;' or '&#x0038;'; and the less-than sign (<) is encoded as '&#60;' or '&#x003C;'.

Any one-, two-, or four-byte character specified in the ISO/IEC 10646 and Unicode standards can be included in a document using this technique.

2.1.2.3 White Space Processing

All IODEF elements support the "xml:space" attribute. If "xml:space"
is set to "preserve," the IODEF processing application MUST treat all white space in the element's content as significant. If "xml:space" is "default," the application is free to do whatever it normally would with white space in the element's content.

2.1.3 Languages in the IODEF

All IODEF tags support the "xml:lang" attribute thereby allowing each element to identify the language in which its content is. The valid language code for the "xml:lang" attribute are described in RFC 3066 [6].

2.2 IODEF Data Types

2.2.1 Integers

Integer attributes are represented by the INTEGER data type. Integer data MUST be encoded in Base 10 or Base 16.

Base 10 integer encoding uses the digits '0' through '9' and an optional sign ('+' or '-'). For example, "123", "-456".

Base 16 integer encoding uses the digits '0' through '9' and 'a' through 'f' (or their upper case equivalents), and is preceded by the characters "0x". For example, "0x1a2b".

2.2.2 Real Numbers

Real (floating-point) attributes are represented by the REAL data type. Real data MUST be encoded in Base 10.

Real encoding is that of the POSIX "strtod" library function: an optional sign ('+' or '-') followed by a non-empty string of decimal
digits, optionally containing a radix character, then an optional exponent part. An exponent part consists of an 'e' or 'E', followed by an optional sign, followed by one or more decimal digits. For example, "123.45e02", "-567,89e-03".

IODEF-compliant applications MUST support both the '.' and ',' radix characters.

2.2.3 Characters and Strings

Single-character attributes are represented by the CHARACTER data type. Multi-character attributes of known length are represented by the STRING data type.

Character and string data have no special formatting requirements, other than the need to occasionally use character references (see Section 2.1.2.1 and Section 2.1.2.2) to represent special characters.

2.2.4 Bytes

Binary data is represented by the BYTE (and BYTE[]) data type.

Binary data MUST be encoded in its entirety using character code references (see ).

2.2.5 Enumerated Types

Enumerated types are represented by the ENUM data type, and consist of an ordered list of acceptable values. Each value has a representative keyword. Within an IODEF document, the enumerated type keywords are used as attribute values.

2.2.6 Date-Time Strings

Date-time strings are represented by the DATETIME data type. Each date-time string identifies a particular instant in time; ranges are not supported.

Date-time strings are formatted according to a subset of ISO 8601:2000 [15] documented in RFC 3339 [14].
2.2.7 NTP Timestamps

NTP timestamps are represented by the NTPSTAMP data type, and are described in detail in RFC 1305 [10] and RFC 2030 [11]. An NTP timestamp is a 64-bit unsigned fixed-point number. The integer part is in the first 32 bits, and the fraction part is in the last 32 bits.

IODEF documents MUST encode NTP timestamps as two 32-bit hexadecimal values, separated by a period ("."). For example, "0x12345678.0x87654321".

2.2.8 Port Lists

A list of network ports are represented by the PORTLIST data type, and consist of a comma-separated list of numbers (individual integers) and ranges (N-M means ports N through M, inclusive). Any combination of numbers and ranges may be used in a single list. For example, "5-25,37,42,43,53,69-119,123-514".

2.2.9 Postal Address

A postal address is represented by the POSTAL data type. The format of this address data is documented in Sections 5.17 - 5.19 of RFC 2256 [12].

2.2.10 Person or Organization

The name of an individual or organization is represented by the NAME data type. The format of the NAME data type is documented in Section 5.4 of RFC 2256 [12].

2.2.11 Telephone and Fax Numbers

A telephone number is represented by the PHONE data type. The format of the PHONE data type is documented in Section 5.21 of RFC 2256 [12].

2.2.12 Email string

An email address is represented by the EMAIL data type. The format
of the EMAIL data type is documented in Section 3.4.1 RFC 2822 [13]

2.2.13 Uniform Resource Identifier strings

A uniform resource identifier (URI) is represented by the URI data type. The format of the URI data type is documented in RFC 2396 [8].

2.2.14 Unique Identifiers

A unique identifier in the context of particular creator of IODEF documents (e.g., a CSIRT) is represented by the UID data type. A globally unique identifier is represented by the GUID data type. The UID and GUID data types are constructed from alphanumerical strings.

3. The IODEF Data Model

In this section, the individual components of the IODEF data model will be discussed in detail. For each class, the semantics will be documented and the relationship between other classes will be presented with an UML diagram.

3.1 IODEF-Document class
The IODEF-Document class is the top level class in the IODEF data model and the DTD. All IODEF documents are instances of the IODEF-Document class.

```
+-----------------+
| IODEF-Document  |
+-----------------+
| STRING version  |<>--{1..*}--[ Incident ]
|                |
+-----------------+

Figure 2: IODEF-Document class
```

The aggregate class that constitutes IODEF-Document is:

Incident
One. The Incident class contains all the incident-related information.

The IODEF-Document class has one attribute:

version
Required. STRING. The version of the IODEF specification to which the IODEF document conforms. The value of this attribute MUST be 1.0

3.2 Incident class

In each exchange of incident related data this data is represented by an instance of the Incident class. This class provides a standardized representation for commonly exchanged incident data and associates a unique identifier with the described activity.
The aggregate classes that constitute Incident are:

IncidentID
One. An incident tracking number assigned to this incident by the party that generated the document.

AlternativeID
Zero or one. A list of incident tracking numbers used by other CSIRTs to refer to same activity as described in the document.

RelatedActivity
Zero or one. A list of incident tracking numbers referencing related incidents.

IncidentData
Zero or more. The event(s) that constitute the incident about which the IODEF-Document conveys information.

AdditionalData
Zero or more. Extension area for data that cannot be represented anywhere else.

The Incident class has two attributes:

purpose
Required. ENUM. The purpose of the IODEF-Document. This attribute is defined as an enumerated list:

1. handling. The IODEF-Document was sent for incident-handling purposes;

2. statistics. The IODEF-Document was sent to be included in a
data-repository for statistical purposes;

3. warning. The IODEF-Document was sent as a warning;

4. other. The IODEF-Document was sent for purposes specified in the Expectation element.

restriction
Optional. ENUM. This attribute indicates the disclosure guidelines to which the sender expects the recipient of the IODEF-Document to adhere. However, it is the choice of the recipient of the document to honor this guideline.

The value of this attribute is logically inherited by the children of this class. That is to say, the disclosure rules applied to this class, also apply to its children.

It is possible to set a granular disclosure policy, since all of the high-level classes have a restriction attribute. Therefore, a child can override the guidelines of a parent class, be it to tighten or relax the disclosure rules (i.e., a child has a weaker policy than an ancestor; or an ancestor has a weak policy, and the children selectively apply more rigid controls). The implicit value of the restriction attribute for a class that did not specify one can be found in the closest ancestor that did specify a value.

This attribute is defined as an enumerated value with a default value of "private".

Note: the default value of the restriction attribute is only defined in the context of the Incident class. In other classes where this attribute is used no default is specified.

1. public. There is no restriction level applied to the information;

2. need-to-know. The information may be shared with other parties that are involved in the incident (e.g., multiple victim sites can be informed of each other);

3. private. The information may not be shared.

4. default. The information can be shared according to an information disclosure policy pre-arranged by the communicating parties.
3.3 IncidentID class

The IncidentID class represents an incident tracking number (UID) that is unique in the context of the CSIRT and identifies the activity characterized in an IODEF-Document.

+------------------+
| IncidentID       |
+------------------+
| UID              |
|                  |
| GUID   name      |
+------------------+

Figure 4: the IncidentID class

The IncidentID class has one attribute:

name
  Required. GUID. An identifier for the CSIRT that created the IODEF-Document.

An implementation strategy for the GUID name attribute would be to use global or regional CSIRT registries such as FIRST or the European Trusted Introducer.

3.4 AlternativeID class

The AlternativeID class references the incident tracking numbers or unique identifiers used by other entities (e.g., CSIRTs) to refer to activity identical to that characterized in this IODEF-Document. Thus, tracking numbers listed as an AlternativeID are the same events detected by another CSIRT, but seem from a different perspective. It follows, the incident tracking numbers of the organization that generated the IODEF-Document should never be considered an AlternativeID.

If the incident is not the identical activity, but is related (e.g., same methodology or intruder), then its incident tracking number
should instead be represented in the RelatedActivity (Section 3.5) class.

The aggregate classes that constitute AlternativeID are:

**IncidentID**
One or more. Unique identifiers assigned by another entity for the identical activity characterized in the IODEF-Document.

The AlternativeID class has one attribute:

**restriction**
Optional. ENUM. This attribute has been defined in Section 3.2.

### 3.5 RelatedActivity class

The RelatedActivity class references the incident tracking numbers or unique identifiers of incidents that are related to the one described in the IODEF document. These references may to local incident tracking numbers, as well as, to those of other CSIRTs.

The specifics of how a CSIRT came to believe that two incidents are related is considered out of scope.
The aggregate classes that constitute RelatedActivity are:

**IncidentID**

One or more. Unique identifiers assigned by the CSIRT.

The RelatedActivity class has one attribute:

**restriction**

Optional. ENUM. This attribute has been defined in Section 3.2.

### 3.6 AdditionalData

The AdditionalData class serves as an extension mechanism for information not otherwise represented in the data model. For relatively simple information, atomic data (integers, strings, etc.) types are provided with a mechanism to annotate their meaning. The class can also be used to extend the data model and the DTD to support proprietary extensions by encapsulating entire XML documents conforming to another DTD (e.g., IDMEF). A detailed discussion for extending the data model and the DTD can be found in Section 4.

Unlike XML, which is self-describing, atomic data must typically be documented to convey its meaning. This information is described in the 'meaning' attribute. Since these description are outside the scope of the specification, some additional coordination may be required to ensure that a recipient of a document using the AdditionalData classes can make sense of the custom extensions.
The AdditionalData class has three attributes:

**restriction**
Optional. ENUM. This attribute has been defined in [Section 3.2](#).

**type**
Required. ENUM. The data type of the element content. The permitted values for this attribute are shown below. The default value is "string".

1. boolean. The element contains a boolean value, i.e., the strings "true" or "false"
2. byte. The element content is a single 8-bit byte (see [Section 2.2.4](#));
3. character. The element content is a single character (see [Section 2.2.3](#));
4. date-time. The element content is a date-time string (see [Section 2.2.6](#));
5. integer. The element content is an integer (see [Section 2.2.1](#));
6. ntpstamp. The element content is a NTP timestamp (see [Section 2.2.7](#));
7. portlist. The element content is a port list (see [Section 2.2.8](#));
8. real. The element content is a real number (see Section 2.2.2);

9. string. The element content is a string (see Section 2.2.3);

10. xml. The element content is XML-tagged data (see Section 4).

meaning
Optional. STRING. A description of the semantics of the custom data in this class.

3.7 IncidentData

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.

Many of the aggregated classes of IncidentData are also found in EventData, albeit with different occurrence indicators. However, the semantics of these classes is quite different. The classes of IncidentData reflect information relevant across the entire incident, while the classes of EventData provide information only relevant to the given event or system node being described. The relationship between the IncidentData and EventData classes is complementary. The latter provides summary information, while the former provides more specific details. For example, the overall impact of the incident (represented in IncidentData) might be denial of service, but it might be worth mentioning that there were specific machines (represented in EventData) which also suffered a root compromise. In another example, an organizational contact can be provided in IncidentData class, while more specific contacts for the individual hosts can be in the EventData class.

IncidentData also ensures that certain mandatory information will be present in the data model.

+------------------+
| IncidentData     |
+------------------+
The aggregate classes that constitute IncidentData are:

Description
Zero or more. STRING. A free-form textual description of the incident activity

Assessment
One or more. A characterization of the impact the incident activity.

Method
Zero or more. The techniques (e.g., tools, vulnerabilities) used by the intruder.
DetectTime
Zero or one. The time the incident activity was first detected.

StartTime
Zero or one. The time the incident activity started.

EndTime
Zero or one. The time the incident activity ended.

ReportTime
One. The time the incident activity was reported.

Contact
One or more. Contact information for the parties involved in the incident.

Expectation
Zero or more. Expected action to be performed by the recipient of the document.

History
Zero or one. Documents significant events or actions that occurred during the course of handling the incident.

EventData
Zero or more. Details on the Data on the (security) events that lead to the incident.

AdditionalData
Zero or more. An area to extend the data model with information that can not be represented elsewhere.

The IncidentData class has one attribute:

restriction
Optional. ENUM. This attribute is defined in Section 3.2.

3.8 Contact class

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.
People and organizations are treated interchangeably as contacts; one can be associated with the other using the recursive definition of the class. The 'type' attribute determines the type of contact information being provided.

The recursive definition of this class (the Contact class is aggregated into the Contact class) provides a way to relate information without requiring the explicit use of identifiers in the classes. When grouping people into organizations it is RECOMMENDED to nest the persons instances into an organization instance of this class.

```
+------------------+
| Contact          |
+------------------+
  ENUM restriction |<>--{0..1}--[ name ]
  ENUM role        |
  ENUM type        |<>--{0..*}--[ Description ]
        |<>--{0..*}--[ RegistryHandle ]
        |<>--{0..1}--[ PostalAddress ]
        |<>--{0..*}--[ Email ]
        |<>--{0..*}--[ Telephone ]
        |<>--{0..1}--[ Fax ]
        |<>--{0..1}--[ Timezone ]
        |<>--{0..*}--[ Contact ]
```

Figure 9: the Contact class

The aggregate classes that constitute the Contact class are:

name
Zero or one. NAME. The name of the contact. The contact may either be an organization or a person. The type attribute dictates the semantics (organization or person).

Description
Zero or one. STRING. Free-form description of the this contact. In the case of a person, this is often the organizational title of
RegistryHandle
Zero or many. The handle name in a registry. Care must be taken to ensure that a handle is meaningful to the recipient. Intra-organizational handles are of not much use for extra-organizational communication.

PostalAddress
Zero or one. POSTAL. The postal address of the contact formatted according to Section 2.2.9.

Email
Zero or many. EMAIL. The email address of the contact formatted according to Section 2.2.12.

Telephone
Zero or many. PHONE. The telephone number of the contact formatted according to .

Fax
Zero or one. PHONE. The facsimile telephone number of the contact formatted according to .

Timezone
Zero or one. STRING. The timezone in which the contact resides.

Contact
Zero or many. Recursive definition of Contact, allowing for grouping of data. An example of this is an organization with multiple contact persons.

The Contact class has three attributes:

restriction
Optional. ENUM. This attribute is defined in Section 3.2.

role
Required. ENUM. Indicates the role the Contact fulfills. This attribute is defined as an enumerated list:

1. creator. The entity that generate the IODEF document.
2. admin. An administrative contact for a host or network.
3. tech. A technical contact for a host or network.
4. irt. The CSIRT involved in handling the incident.
5. cc. An entity that is to be kept informed about the the

handling of the incident.

```plaintext
type
  Required. ENUM. Indicates the type of Contact being provided. This attribute is defined as an enumerated list:

  1. person.

  2. organization.
```

### 3.8.1 RegistryHandle class

The RegistryHandle class represents a handle to an Internet registry or community-specific database. A handle consists of a name specified in the element content, and the database to which it belongs specified in the type attribute.

```
+------------------+
| RegistryHandle   |
+------------------+
| STRING           |
| ENUM type        |
+------------------+

Figure 10: The RegistryHandle class

The RegistryHandle class has one attribute:

```plaintext
type
  Required. ENUM. The database to which the handle belongs. The
default value is 'local'. The possible values are:

1. internic. Internet Network Information Center
2. apnic. Asia Pacific Network Information Center
3. arin. American Registry for Internet Numbers
4. lacnic. Regional Latin-American and Caribbean IP Address Registry
5. ripe. Reseaux IP Europeens
6. ti. TERNEA Trusted Introducer

7. local. A database local to the CSIRT.

### 3.9 Time classes

The data model uses for different classes to represent a timestamp. Their definition is identical, but each is named differently to convey a semantic difference.

The element content of each class is a timestamp formatted according to the DATETIME data type (see Section 2.2.6).

```
+----------------------------------+
| {Start| End| Report| Detect}Time |
+----------------------------------+
| DATETIME                         |
| NTPSTAMP ntpstamp                |
```

Figure 11: the Time classes

The Time classes have one attribute:

ntpstamp
Optional. NTPTIMESTAMP. The NTP timestamp representing the
timestamp in the element content. The NTPSTAMP format of this
attribute's value is described in Section 2.2.7.

The use of the ntpstamp attribute is optional since it is redundant.
However, it has been maintained to ensure compatibility with the
IDMEF [7]. Representing a timestamp in both the element content and
attribute is NOT RECOMMENDED. However, if both are used, their
values MUST be identical.

3.9.1 StartTime

The StartTime class represents timestamp for the start of an
activity.

3.9.2 EndTime

The EndTime class represents the timestamp for the end of an
activity.

3.9.3 DetectTime

The DetectTime class represents the timestamp of when an activity was
first detected.

3.9.4 ReportTime

The ReportTime class represents the timestamp of when a detected
activity was reported.

3.9.5 DateTime

The DateTime class is a generic representation of a timestamp. Its
semantics should be inferred from the parent class into which it is
aggregated.

3.10 Expectation class

The Expectation class conveys to the recipient of the IODEF document
the actions the sender is requesting.
Figure 12: the Expectation class

The aggregate classes that constitute Expectation are:

**Description**
- One or many. STRING. A free-form description of the desired action(s).

**StartTime**
- Zero or one. The time at which the action should be performed. A timestamp that is earlier than the ReportTime specified in the IncidentData class denotes that the expectation should be fulfilled as soon as possible. The absence of this element leaves the execution of the expectation to the discretion of the recipient.

**EndTime**
- Zero or one. The time by which the action should be completed. If the action is not carried out by this time, it should no longer be performed.

**Contact**
- Zero or one. The expected actor for the action. The 'role' attribute of the Contact MUST be set to "actor".

The Expectations class has three attributes:

**restriction**
Optional. ENUM. This attribute is defined in Section 3.2.

priority
Optional. ENUM. Indicates the desired priority of the action. This attribute is an enumerated list with no default value.

1. low. Low priority
2. medium. Medium priority
3. high. High priority

category
Optional. ENUM. Classifies the type of action requested. This attribute is an enumerated list with no default value.

1. nothing. No action is requested. Do nothing with the information.
2. contact-site. Contact the listed site in the recipient's constituency.
3. contact-me. Contact the originator of the document.
4. block. Block or investigate machines listed in the document in the recipient's constituency.

3.11 Method class

The Method class provides information about the methodology used by the intruder to perpetrate the events of the incident. This class can reference well-known vulnerability or exploit databases, list the intruder tools used in the attack, and provide for a free-form description of the activity.
The Method class is composed of two aggregate classes:

Classification
Zero or many. A reference to a well-known vulnerability or exploit databases.

Description
Zero or many. STRING A free-form text description of the methodology used in the incident.

The Method class has one attribute:

restriction
Optional. ENUM. This attribute is defined in Section 3.2.

3.11.1 Classification class

The Classification class is a reference to an external database of computer vulnerabilities, exposures, or viruses. A reference consists of the database name, the entry in the database, and the URI to this entry.

The aggregate classes that constitute Classification:

name
One. STRING. The name of the reference to the database specified in the origin attribute.
url
   One. URI. A URL to additional information about the vulnerability or exposure referenced by the name.

The Classification class has two attribute:

restriction
   Optional. ENUM. This attribute is defined in Section 3.2.

origin
   Required. ENUM. The name of the database to which the reference is being made. The permitted values are shown below.

   1. bugtraqid. Bugtraq
   2. cve. Common Vulnerabilities or Exposures
   3. certcc. CERT Coordination Center Vulnerability Catalog
   4. vendor. A product vendor whose name should be specified in the name class
   5. local. A local database.
   6. other.

3.12 Assessment class

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

Note: The IODEF definition of the Assessment class reuses the IDMEF definition (see Section 4.2.4.5 of [7]), but also extends it.
The aggregate classes that constitute Assessment are:

**Impact**
Zero or many. Technical impact of the activity on the computers and networks.

**TimeImpact**
Zero or many. Impact of the activity measured with respect to time.

**MonetaryImpact**
Zero or many. Impact of the activity measured with respect to money.

**LifeImpact**
Zero or many. Impact of the activity measured with respect to human life.

**Confidence**
Zero or one. An estimate of confidence in the assessment.

The Assessment class has one attribute:

**restriction**
Optional. ENUM. This attribute is defined in [Section 3.2](#).
3.12.1 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.

Attributes allow the impact to be classified according to the consequences on the host and the severity of these consequences. The element content is used for the description.

Note: The IODEF definition of the Impact class reuses the IDMEF definition (see Section 4.2.6.1 of [7]), but also extends it and alters the semantics.

```
+------------------+
| Impact           |
+------------------+
| STRING           |
| ENUM restriction |
| ENUM severity    |
| ENUM completion  |
| ENUM type        |
+------------------+
```

Figure 16: Impact class

The element content may be empty, or contain a free-form description (STRING) of the technical impact.

The Impact class has four attributes:

- **restriction**
  Optional. ENUM. This attribute has been defined in Section 3.2.

- **severity**
  Optional. ENUM. An estimate of the relative severity of the activity. The permitted values are shown below. There is no default value.

  1. low. Low severity
2. medium. Medium severity
3. high. High severity

completion
Optional. ENUM. An indication of whether the creator of the IODEF document believes the activity was successful. The permitted values are shown below. There is no default value.

1. failed. The attempt was not successful
2. succeeded. The attempt succeeded

3.12.2 TimeImpact class

The TimeImpact class describes the non-technical impact of the activity on an organization as a function of time. Different types of time calculations and well as units can be used.
The element content will be a numeric value (REAL) specifying the impact as a function of time. The attributes represent the specific units and metric.

The TimeImpact class has four attributes:

- **restriction**
  Optional. ENUM. This attribute has been defined in Section 3.2.

- **severity**
  Optional. ENUM. An estimate of the relative severity of the activity. The permitted values are shown below. There is no default value.
  1. low. Low severity
  2. medium. Medium severity
  3. high. High severity

- **metric**
  Required. ENUM. Defines the metric in which the time is expressed. The permitted values are shown below. There is no default value.
  1. labor. Total staff-time to recovery from the activity (e.g., 2 employees working 4 hours each would be 8 hours)
2. elapsed. Elapsed time from the beginning of the recovery to its completion.

3. downtime. Duration of time for which some provided service(s) was not available.

units
Required. ENUM. Defines the units in which the metric is expressed. The permitted values are shown below. The default value is "hours".

1. seconds. Seconds
2. minutes. Minutes
3. hours. Hours
4. days. Days

3.12.3 MonetaryImpact class

The MonetaryImpact class describes the financial impact of the activity on an organization. For example, this impact may consider loss due to the cost of the investigation or recovery, diminished productivity of the staff, or a tarnished reputation that will affect future opportunities.

+------------------+
| MonetaryImpact   |
+------------------+
| REAL             |
|                  |
| ENUM restriction |
| ENUM severity    |
| ENUM metric      |
| STRING currency  |
+------------------+
The element content will be a numeric value (REAL) specifying the impact as a function of money. The attributes represent the specific currency and metric.

The MonetaryImpact class has four attributes:

restriction
  Optional. ENUM. This attribute has been defined in Section 3.2.

severity
  Optional. ENUM. An estimate of the relative severity of the activity. The permitted values are shown below. There is no default value.
  1. low. Low severity
  2. medium. Medium severity
  3. high. High severity

currency
  Required. ENUM. Defines the currency in which the monetary impact is expressed. The permitted values are defined in ISO 4217:2001, Codes for the representation of currencies and funds [18]. There is no default value.

3.12.4 LifeImpact class

The LifeImpact class describes the loss of human life or injury due to an incident.
The element content will be a numeric value (INTEGER) specifying the impact as a function of human life. The attributes represent the specific metric.

The LifeImpact class has three attributes:

restriction
  Optional. ENUM. This attribute has been defined in Section 3.2.

severity
  Optional. ENUM. An estimate of the relative severity of the activity. The permitted values are shown below. There is no default value.
  1. low. Low severity
  2. medium. Medium severity
  3. high. High severity

metric
  Required. ENUM. Defines the metric in which the LifeImpact is expressed. The permitted values are shown below. There is no default value.
  1. Deaths
  2. Injuries

3.12.5 Confidence class

The Confidence class represents a best estimate of the validity and accuracy of the described impact (see Section 3.12) of the incident activity. This estimate can be expressed as a category, or a numeric calculation.
Note: The IODEF definition of the Confidence class reuses the IDMEF definition (see Section 4.2.6.3 of [7]), but also extends it and alters the semantics.

The Confidence class has been reused from the IDMEF [7], it has been extended and has been altered.

```
+------------------+
| Confidence       |
+------------------+
| REAL             |
| ENUM restriction |
| ENUM rating      |
+------------------+
```

**Figure 20: Confidence class**

The element content may be empty if the rating attribute is not set to "numeric". Otherwise, a confidence value (REAL) must be provided.

The Confidence class has two attributes:

- **restriction**
  Optional. ENUM. This attribute has been defined in Section 3.2.

- **rating**
  Required. ENUM. Indicates the confidence the CSIRT has in its assessment. The permitted values are shown below. The default value is "numeric."

1. low
2. medium
3. high
4. numeric. The CSIRT has provided a probability value indicating its confidence in its assessment.
5. unknown

This element SHOULD only be used when the CSIRT can produce meaningful information. When only a rough estimate is possible "low", "medium", or "high" SHOULD be used as the rating value.

When a reasonable probability estimate is possible "numeric" SHOULD
be used as the rating value and include a numeric confidence value in the element content. This numeric value is a floating point number between 0.0 and 1.0, inclusive.

Different CSIRTs may compute and represent confidence values in different ways. Care should be taken to take proper notice of the exact meaning of the confidence values of different CSIRTs when comparing confidence values.

3.13 History class

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.

The level of detail maintained in this log is left up to the discretion of those handling the incident.

+------------------+
| History          |
+------------------+
| ENUM restriction |<>--{1..*}--[ HistoryItem ]|
+------------------+

Figure 21: The History class

The class that constitute History are:

HistoryItem

One or many. Entries in the history log of significant events or actions performed by the involved parties.

The History class has one attribute:

restriction

Optional. ENUM. This attribute is defined in Section 3.2.

3.13.1 HistoryItem class
The HistoryItem class is a particular entry in the History (Section 3.13) log that documents a particular significant action or event that occurred in the course of handling the current incident. This details of the entry in this log are a free-form description, but each can also be categorized.

The aggregate classes that constitute HistoryItem are:

IncidentID
Zero or One. In history logs created by multiple parties, the IncidentID provides a way specify which CSIRT created the particular entry and reference this organization's local incident tracking number for this activity. When a single organization is maintaining the history log, this class can be ignored.

DateTime
One. Timestamp of the this entry in the history log (e.g., when the action described in the Description was taken).

Description
One or many. STRING. A free-form textual description of the action or event to be document in the history log.

The HistoryItem class has two attributes:

restriction
Optional. ENUM. This attribute has been defined in Section 3.2.

type
Optional. ENUM. Classifies the type of activity or event being
document in this history log entry. The particular details of the entry are a free-form description documented in the Description class. Possible values are an enumerated list whose default value is "other":

1. triaged. The incident data was received and processed by an IHS
2. notification. Notification to an involved party in the incident was sent (e.g., a CSIRT sending a message to the attacking site).
3. shared-info. Information about this incident was shared with party not directly involved.
4. received-info. Additional information about the incident was received
5. remediation. The incident has been resolved; a short description may be included.
6. other.

3.14 EventData class

The EventData class describes the events of the incident surrounding a particular set of hosts or networks. This description includes the systems from which the activity originated and those targeted, an assessment of the techniques used by the intruder, the impact of the activity on the organization, a list of incident handling tasks performed, and any forensic evidence discovered.
The aggregate classes that constitute EventData are:

Description
Zero or more. STRING. A free-form textual description of the event.

System
Zero or more. The systems (nodes, networks) involved in the event as either sources, targets or intermediaries.

Method
Zero or more. The methods by which the event was staged. Information about tools used and vulnerabilities exploited.

Record
Zero or one. Support data (e.g., log files) that provides information on the events.

StartTime
Zero or one. The time the event started.

EndTime
Zero or one. The time the event ended.

DetectTime
Zero or one. The time the event was detected.

Contact
Zero or more. The different parties involved in the incident

Assessment
Zero or one. Indicates the impact of the incident on the target and the actions taken.
AdditionalData
Zero or one. Anything that can not be put in one of the other elements

Event
Zero or more. Recursive definition of Event, allowing for grouping of data

The EventData class has one attribute:

restriction
Optional. ENUM. This attribute is defined in Section 3.2.

3.15 Relating the IncidentData and EventData classes

At first glance, the duplication in the aggregate classes of IncidentData and EventData are obvious. However, the semantics of these classes are quite different. IncidentData provides summary information about the entire incident, while EventData provides information about a subset of the incident.

For example, note that the Assessment class is aggregated in both classes. Consider a case where IncidentData:Assessment:MonetaryImpact has been assigned a value of x. Now, consider a value of y (where y < x) being assigned to a given MonetaryImpact class that is aggregated in the EventData class. The semantics of these two values is some monetary loss. In the case of the figure in the IncidentData class, this loss is incident-wide. The figure in EventData is a subset of this overall loss, and allows one to associate a particular loss with a given subset of events that constitute the incident. It effectively provides a breakdown (or more specific description) of the overall loss previously specified in the IncidentData class.

3.16 Cardinality of EventData

The recursive definition of this class (the EventData class is aggregated into the EventData class) provides a way to related information without requiring the explicit use of unique attribute identifiers in the classes. The depth of an element in the XML tree
is used to related information.

The EventData class can be thought of as a container describing the properties of an event in an incident. These properties include: the hosts involved, impact of the incident activity on the hosts, forensic logs, etc. One groups (via an instance of the EventData class) hosts (i.e., System class) around these common properties.

A child EventData class (and all its siblings) logically "inherits" the aggregated classes of a parent EventData class. However, the presence of sibling EventData classes (it "never" makes sense to have only one EventData child in an EventData class) means that there are some disjoin properties of the event. These children of the parent EventData class represent these differences, while still retaining a way to represent the common properties (i.e., the parent-child relationship).

For example, an EventData class might be used to describe two machines involved in an incident. This description can be achieved using multiple instances of the System class. It happens that the technical contact (i.e., Contact class) for these two machines is identical, but the impact (i.e., Assessment class) is different. The problem lies in representing two hosts with a common contact, but different impacts without duplicating any information. This event can be represent with the following design represented in Figure 24.
3.17 System class

The System class represents the technical information for a given computer or network involved in the incident.

The systems represented by this class are categorized according to the role they played in the incident via through the category attribute. The value of this category attribute dictates the semantics of the aggregated classes in the System class.

The meaning of the Node, User, Process, and Service class depend on the value of the category attribute of the System class. If the System class category attribute is 'source', then the described aggregated classes denote the machine, user, process, or service from which the activity is originating. With a category attribute value of 'target' or 'intermediary', then the described machine, user, process, or service is the one targeted in the activity.
The aggregate classes that constitute System are:

**Node**
One. A host or network involved in the incident activity.

**User**
Zero or more. The application or operating system user running on the specified host that was involved in the incident.

**Process**
Zero or more. The process targeted or the source of the attack on the specified host.

**Service**
Zero or one. The network service targeted on the host specified in Node.

**FileList**
Zero or one. Information about the files on the host involved in the incident.

The System class has four attribute:

**restriction**
Optional. ENUM. This attribute is defined in Section 3.2.

**category**
Required. ENUM. Classifies the role the System played in the incident activity. The possible values are:
1. **source.** The System was the source of the attack

2. **target.** The System was the target of the attack

3. **intermediate.** The System was an intermediate machine used in the attack.

**interface**
- Optional. STRING. Specifies the interface on which the event(s) on this System originated.

**spoofed**
- Optional. ENUM. An indication of confidence as to whether this System was the true target or attacking host. The permitted values for this attribute are shown below. The default value is "unknown".

1. **unknown.** The accuracy of the category information is unknown

2. **yes.** The category value classifying the host or network as a source or target is probably incorrect. In the case of a source, the System is likely a decoy; with a target, the System was likely not the intended victim.

3. **no.** The category value classifying the host or network as a source or target is believed to be correct.

3.18 **Node class**

The Node class is used to identify a host or network device (e.g., routers, switches).

The base definition of the class is reused from the IDMEF specification, see Section 4.2.7.1 of [7]. However, the class has been extended by adding the NodeRole class.
The aggregate classes that constitute Node are:

Location
Zero or one. STRING. The physical location of the equipment.

name
Zero or one. STRING. The name of the equipment (e.g., fully qualified domain name). This information MUST be provided if no Address information is given.

Address
Zero or more. The network or hardware address of the equipment. Unless a name is provided, at least one address must be specified.

DateTime
Zero or one. A timestamp of when the resolution between the name
and address was performed. This information SHOULD be provided if both an Address and name are given.

**NodeRole**

Zero or more. The intended purpose of the equipment.

The Node class has one attribute:

**category**

Optional. ENUM. The context in which the Address and name classes should be considered, if relevant. The permitted values for this attribute are shown below. The default value is "unknown".

1. unknown. Domain unknown or not relevant
2. ads. Windows 2000 Advanced Directory Services
3. afs. Andrew File System (Transarc)
4. coda. Coda Distributed File System
5. dfs. Distributed File System (IBM)
6. dns. Domain Name System
7. hosts. Local hosts file
8. kerberos. Kerberos realm
9. nds. Novell Directory Services
10. nis. Network Information Services (Sun)
11. nisplus. Network Information Services Plus (Sun)
12. nt. Windows NT domain
13. wfw. Windows for Workgroups
### 3.18.1 Address

The Address class represents a network, hardware, and application address. This class is reused outright from the IDMEF specification, see Section 4.2.7.1.1 of [7].

### 3.18.2 NodeRole class

The NodeRole class describes (based on a pre-defined list) the function performed by a particular host.

---

![Figure 27: The NodeRole class](image)

The element content should be empty in all cases other than when the category attribute is set to "other".

The NodeRole class has one attribute:

**category**

Required. Functionality provided by a node. If a value of "other" is specified, a description SHOULD be provided in the
element's content. The default value is "other".

1. client. Client computer
2. server-internal. Server with internal services
3. server-public. Server with public services
4. www. WWW server
5. mail. Mail server
6. messaging. Messaging server (e.g. NNTP, IRC, IM)
7. streaming. Streaming-media server
8. voice. Voice server (e.g. SIP, H.323)
9. file. File server (e.g. SMB, CVS, AFS)
10. ftp. FTP server
11. p2p. Peer-to-peer node
12. name. Name server (e.g. DNS, WINS)
13. directory. Directory server (e.g. LDAP, finger, whois)
14. credential. Credential server (e.g. domain controller, Kerberos)

15. print. Print server
16. application. Application server
17. database. Database server
18. infra. Infrastructure server (e.g. router, firewall, DHCP)
19. log. Logserver
20. other. other role not in this list
3.19 FileList class

The FileList class describes files and other file-like objects on hosts involved in an incident. This class is reused outright from the IDMEF specification, see Section 4.2.7.5 of [7].

3.20 User

The User class describes an application or operating system user account involved in an incident. This class is reused outright from the IDMEF specification, see Section 4.2.7.2 of [7].

3.21 Process

The Process class describes a running program on a given host involved in an incident. This class is reused outright from the IDMEF specification, see Section 4.2.7.3 of [7].

3.22 Service

The Service class describes a network service of a host. This class is reused outright from the IDMEF specification, see Section 4.2.7.4 of [7].

3.23 Record class

The Record class groups log or audit data that provides a record of the incident activity. The source of this data will typically be the output of monitoring tools (e.g., IDMEF messages generated by an IDS, connection logs from a web server) that were used to uncover the malicious activity. These logs should provide evidence as to why a reporter to CSIRT believes an incident has occurred.
Figure 28: Record class

The aggregate class that constitutes Record is:

RecordData
    One or more. Log or audit data generated by a particular type of sensor.

The Record class has one attributes:

restriction
    Optional. ENUM. This attribute has been defined in Section 3.2.

3.23.1 RecordData class

The RecordData class groups log or audit data from a given sensor (e.g., IDS, firewall log) and provides a way to annotate the output.

Figure 29: The RecordData class

The aggregate classes that constitutes RecordData is:

DateTime
    Zero or one. Timestamp information for the RecordItem data.

Description
    Zero or more. STRING. Free-form textual description of the provided RecordItem data. At minimum, this description should...
convey the significance of the provided RecordItem data.

Analyzer
Zero or one. Information about the sensor used to generate the RecordItem data.

RecordItem
One or more. Log, audit, or forensic data.

The RecordData class has one attributes:

restriction
Optional. ENUM. This attribute has been defined in Section 3.2.

3.23.1.1 Analyzer class

The Analyzer class identifies the sensor (e.g., IDS, firewall, web server) used to generate particular log or audit data. The definition of the class is reused from the IDMEF specification, see Section 4.2.7.3 of [7]. However, in this context, the definition of an analyzer is expanded beyond merely an IDS.

3.23.1.2 RecordItem class

The RecordItem class provides a way to incorporate relevant logs, audit trails, or forensic data to support the conclusions made during the course of analyzing the incident. This data can be directly encapsulated as part of this document, or can be referenced whereby using this class as merely a pointer to the relevant information.

The dtype attribute will dictate the type of log data that will be found in this class. This class is very similar to the AdditionalData class (Section 3.6) in that it is essentially an extension class that can support proprietary representations of security event data, not all of which is necessarily in XML.
The RecordItem class has one attribute:

**type**

Required. The type of data included in the element content. The permitted values for this attribute are shown below. The default value is "string".

1. boolean. The element contains a boolean value, i.e., the strings "true" or "false"

2. byte. The element content is a single 8-bit byte (see Section 2.2.4);

3. character. The element content is a single character (see Section 2.2.3);

4. date-time. The element content is a date-time string (see Section 2.2.6);

5. integer. The element content is an integer (see Section 2.2.1);

6. ntpstamp. The element content is a NTP timestamp (see Section 2.2.7);

7. portlist. The element content is a port list (see Section 2.2.8);

8. real. The element content is a real number (see xref target="dt_real_numbers" />);
9. string. The element content is a string (see Section 2.2.3);
10. file. The element content is a base64 encoded binary file;
11. path. The element content is a filesystem path;
12. url. The element content is a URL (see Section 2.2.13);
13. xml. The element content is XML-tagged data (see Section 4).
4. Extending the IODEF

In order to support the changing activity of CSIRTS, the IODEF data model and DTD will need to evolve along with them. To allow new features to be added, both the data model and the DTD can be extended as described in this section. As these extensions mature, they can then be incorporated into future versions of the specification or published separately.

4.1 Extending the data model

There are two mechanisms for extending the IODEF data model: inheritance and aggregation.

- By using inheritance, new subclasses may be derived and given additional attributes or operations not found in the superclass.
- Aggregation allows for entirely new, self-contained classes to be created and associated with a parent class.

Of the two extension mechanisms, inheritance is preferred, because it preserves the existing data model and the operations (methods) executed on the classes of the model. There are explicit guidelines for extending the XML DTD (see Section 4.2) which set limits on where extensions to the data model may be made.
4.2 Extending the XML DTD

There are two ways to extend the IODEF XML DTD:

1. The AdditionalData (see Section 3.6) and RecordItem (see Section 3.23.1.2) classes allow implementers to include arbitrary "atomic" data. (e.g., integers, strings). This approach SHOULD be used whenever possible.

2. The AdditionalData and RecordItem classes allow implementers to extend the IODEF XML DTD with additional DTDs that describe arbitrarily complex data types and relationships.

The following guidelines MUST be followed when extending the IODEF DTD with another DTD in the extension classes:

1. The IODEF description MUST include a document type declaration (see Section 2.1.1.3);

2. The document type declaration MUST define a parameter entity that contains the location of the extension DTD, and then reference that entity:

```xml
<!DOCTYPE IODEF-Document SYSTEM "/path/to/IODEF-Document.dtd"
[ <!ENTITY % x-extension SYSTEM "/path/to/extension.dtd">
  % x-extension; ]>
```

In this example, the "x-extension" parameter entity is defined and then referenced, causing the DTD for the extension to be read by the XML parser.

The name of the parameter entity defined for this purpose MUST be a string beginning with "x-"; there are no other restrictions on the name (other than those imposed on all entity names by XML). Multiple extensions may be included by defining multiple entities and referencing them. For example:

```xml
<!DOCTYPE IODEF-Document SYSTEM "/path/to/IODEF-Document.dtd"
[ <!ENTITY % x-extension SYSTEM "/path/to/extension.dtd">
  <!ENTITY % x-another SYSTEM "/path/to/another.dtd" />
]>
```
3. Extension DTDs MUST declare all of their elements and attributes in a separate XML namespace. Extension DTDs MUST NOT declare any elements or attributes in the "IODEF" or default namespaces.

For example, the "test" extension might be declared as follows:

```xml
<!ELEMENT test:test ( test:a, test:b, test:c )>
<!ATTLIST test:test
  xmlns CDATA #IMPLIED
  xmlns:test CDATA #IMPLIED
>
<!ELEMENT test:a (#PCDATA)>
<!ATTLIST test:a
  test:attr CDATA #IMPLIED
>
<!ELEMENT test:b (#PCDATA)>
<!ELEMENT test:c (#PCDATA)>
```

4. Extensions MUST only be included in the AdditionalData class of the Incident class whose "type" attribute is "xml". For example:

```xml
<AdditionalData type="xml">
  <test:test
    xmlns:test="http://www.ietf.org/iodef/test.html"
    xmlns="http://www.ietf.org/iodef/test.html">
    <test:a test:attr="...">...</test:a>
    <test:b>...</test:b>
    <test:c>...</test:c>
  </test:test>
</AdditionalData>
```
5. Processing Considerations

This section discusses some of the special considerations that must be taken into account by implementers of the IODEF.
5.1 XML Validity and Well-Formedness

The IODEF documents MUST be well-formed, and when possible and practical the documents SHOULD also be valid.

It is expected that IODEF-compliant applications will normally not include the IODEF DTD in their communications. Instead, the DTD will be referenced in the document type declaration section of the IODEF document (see Section 2.1.1.3).

While an XML document SHOULD contain a document type declaration. This requirement imposes a significant overhead on an IODEF-compliant application in bandwidth consumption and computation for the DTD may need to be downloaded and parsed before use by the XML parser.

Implementers MAY decide to have entities who regularly exchange IODEF message agree out-of-band on the particular document type definition they will be using to exchange messages (the standard one as defined here, or one with extensions), and then omit it from IODEF documents. The method for negotiating this agreement is outside the scope of this document.

NOTE: Care must be taken in negotiating any such agreements, as each entities will have to keep state on this agreed upon document type definition. The management complexity of these negotiations grows more complex as entities make such arrangements with many collaborators.

5.2 Unrecognized Data and XML Tags

On occasion, an IODEF-compliant application may receive a well-formed, or well-formed and valid IODEF document containing tags or content in the tags that are not expected. These spurious conditions might include:

- Unrecognized tags used in one of the extension classes (i.e., AdditionalData or RecordItem);
- Unrecognized tags outside of the extension classes; or
- Well-formed and valid document where element or attribute values to not conform to the expected values identified by an enumerated list;
IODEF-compliant applications MUST continue to process IODEF documents that contain unknown tags, provided that these documents are well-formed. It is up to the individual application to decide how to process any content from the unknown tag.
6. Internationalization issues

Internationalization and localization is of specific concern to the IODEF. It is only through collaboration, often across language barriers, that certain incidents be resolved.

XML already supports different character encodings. This flexibility will allow information encoded in the IODEF to be in most written languages. Furthermore, XML also provides the xml:lang attribute through which the type of language being used in a given element can be specified. By including this attribute in the %attlist.global entity found in all elements, users of the IODEF can use different languages in the same document.

The data model ensures that the cardinality of the Description class is always one-to-many with its parent. One of the intents for this design was to allow the same description to be repeated in another instance of the Description class, but in a different language. Parsers of the IODEF document, could extract only the elements with the relevant language.

Supporting different languages allows CSIRTs to localize the IODEF. However, it does not aid data interchange if the recipient of a document does not understand the underlying language. In order to ensure that the recipient can at least crudely approximate the contents of the document, the data model relies on enumerated attributes that are standardized to convey meaning (e.g., %attlist.purpose).
7. Examples

These examples provide an idea of what IODEF-Documents can look like. It must be stressed that as IODEF is a data-exchange-format, it does not specify detailed rules on which elements and attributes to use under all imaginable circumstances.

7.1 Code Red detection notification

The following message is a typical example of an incident where one host is infected with a worm. The initial report is sent in by email, the subsequently shown IODEF-Document illustrates the communication between the responsible CSIRT and its constituent. The constituent is a contact for the CSIRT and responsible for coordinating the required actions at his site.

From e-citizen@hisdomain.de
Date: 13 Sep 2001 23:19:24 -0000
From: e-citizen@hisdomain.de
To: cert-for-ourdornial.pl@ourdornial.pl
Subject: 10.1.1.2 - Code Red Virus detected

Automated message,
you don't have to reply to this email.

Your system with the IP number 10.1.1.2 seems to be infected with the Code Red virus.

For more information see http://www.incidents.org/react/code_redII.php

Please fix the problem or inform a person who is responsible for that machine to do so.

>From our web server logs (Port 80):
Figure 35: Code Red detection notification: initial report

```xml
<IODEF-Document version="1.0">
  <Incident restriction="need-to-know" purpose="handling">
    <IncidentID name="CERT-FOR-OUR-DOMAIN.PL">CERT-FOR-OUR-DOMAIN.PL#189</IncidentID>
    <IncidentData>
      <Description>Host sending out Code Red probes</Description>
      <ReportTime>2001-09-13T23:19:24+00:00</ReportTime>
      <Expectation category="other">
        <Description>Track and clean host</Description>
      </Expectation>
      <Assessment>
        <Impact severity="low" completion="failed" type="none"/>
      </Assessment>
      <Contact role="creator" role="irt" type="organization">
        <name>CERT-FOR-OUR-DOMAIN.PL</name>
        <Email>cert-for-our-domain.pl@ourdomain.pl</Email>
      </Contact>
      <Contact role="tech" type="organization">
        <name>Constituency-contact for 10.1.1.2</name>
        <Email>Constituency-contact@10.1.1.2.pl</Email>
      </Contact>
    </IncidentData>
    <History>
      <HistoryItem type="notification">
        <IncidentID name="CERT-FOR-OUR-DOMAIN.PL">CERT-FOR-OUR-DOMAIN.PL#189</IncidentID>
        <Description>Notification sent to Constituency-contact@10.1.1.2.pl</Description>
        <DateTime>2001-09-14T08:19:01+00:00</DateTime>
      </HistoryItem>
    </History>
  </IncidentData>
</IODEF-Document>
```
Figure 36: Code Red detection notification: CSIRT response

7.2 IODEF-Document with XML signature

7.3 IODEF-Document encrypted using XML encryption

7.4 IODEF-Document encrypted and signed using XML signature & encryption
8. The IODEF Document Type Definition

<?xml version="1.0" encoding="UTF-8"?>
<!--
*****************************************************************************
*****************************************************************************
*** IncidentData Exchange Format XML DTD ***
*** Version 01, September 2003 ***
*****************************************************************************
*****************************************************************************
-->
<!ENTITY % attlist.iodef "
version CDATA #FIXED '0.20'
<!DOCTYPE IODEF-Document [
<!ELEMENT IODEF-Document (Incident+)>
<!ATTLIST IODEF-Document
%attlist.iodef;
xmlns:iodef CDATA #FIXED "urn:iana:xml:ns:iodef"
]>

<!ELEMENT Incident (IncidentID, AlternativeID?, RelatedActivity?, IncidentData, AdditionalData?)>
<!ATTLIST Incident
restriction %attvals.restriction; "private"
purpose %attvals.purpose; #REQUIRED
>
<!ELEMENT IncidentID (#PCDATA)>
<!ATTLIST IncidentID
name CDATA #IMPLIED
>

<!ELEMENT AlternativeID (#PCDATA)>
<!ATTLIST AlternativeID
name CDATA #IMPLIED
>

<!ELEMENT RelatedActivity (#PCDATA)>
<!ATTLIST RelatedActivity
name CDATA #IMPLIED
>

<!ELEMENT IncidentData (#PCDATA)>
<!ATTLIST IncidentData
name CDATA #IMPLIED
>

<!ELEMENT AdditionalData (#PCDATA)>
<!ATTLIST AdditionalData
name CDATA #IMPLIED
>
<!ELEMENT AlternativeID (IncidentID+)>
<!ATTLIST AlternativeID
    restriction %attvals.restriction; #IMPLIED>

<!--
====================================================================
==  RelatedActivity class                                         ==
====================================================================
-->
<!ELEMENT RelatedActivity (IncidentID+)>
<!ATTLIST RelatedActivity
    restriction %attvals.restriction; #IMPLIED>

<!--
====================================================================
===  AdditionalData class                                        ===
====================================================================
-->
<!ELEMENT AdditionalData ANY>
<!ATTLIST AdditionalData
    restriction %attvals.restriction; #IMPLIED
    type %attvals.dtype; #REQUIRED
    meaning CDATA #IMPLIED>

<!--
====================================================================
===  IncidentData class                                          ===
====================================================================
-->
<!ELEMENT IncidentData (Description*, Contact+, ReportTime, DetectTime?, StartTime?, EndTime?)>
<!ATTLIST IncidentData
    restriction %attvals.restriction; #IMPLIED>

<!--
====================================================================
===  Contact class                                               ===
===    - Name
===    - RegistryHandle
===    - PostalAddress
===    - Email
===    - Telephone

-->
<!ELEMENT Contact (Name?, RegistryHandle?, PostalAddress?, Email?, Telephone?)>
<!ATTLIST Contact
    restriction %attvals.restriction; #IMPLIED
    type %attvals.dtype; #REQUIRED
    meaning CDATA #IMPLIED>
---
  - Fax
  - TimeZone
  - Contact (recursive)
---
<!ELEMENT Contact (Name?, Description*, RegistryHandle*, PostalAddress?, Email*, Telephone*, Fax*)>
<!ATTLIST Contact
  contactrole (creator | admin | tech | irt | cc) #REQUIRED
  contacttype (person | organization) #REQUIRED
  restriction %attvals.restriction; #IMPLIED>

<!ELEMENT RegistryHandle (#PCDATA)>
<!ATTLIST RegistryHandle
type %attvals.registrytype; "local">

<!ELEMENT PostalAddress (#PCDATA)>
<!ATTLIST PostalAddress
  lang NMTOKEN #IMPLIED>

<!ELEMENT Email (#PCDATA)>
<!ELEMENT Telephone (#PCDATA)>
<!ELEMENT Fax (#PCDATA)>

---

---
  - Time-based classes
  ---
  - DateTime
  ---
  - ReportTime
  ---
  - DetectTime
  ---
  - StartTime
  ---
  - EndTime
---

<!ELEMENT DateTime (#PCDATA)>
<!ATTLIST DateTime
  ntpstamp CDATA #IMPLIED>

<!ELEMENT ReportTime (#PCDATA)>
<!ATTLIST ReportTime
  ntpstamp CDATA #IMPLIED>

<!ELEMENT DetectTime (#PCDATA)>
<!ATTLIST DetectTime
  ntpstamp CDATA #IMPLIED>

<!ELEMENT StartTime (#PCDATA)>
<!ATTLIST StartTime
  ntpstamp CDATA #IMPLIED>

>
<!ELEMENT EndTime (#PCDATA)>  
<!ATTLIST EndTime  
  ntpstamp CDATA #IMPLIED>  

<!--  
====================================================================  
===  History class                                               ===  
====================================================================  
-->  
<!ELEMENT History (HistoryItem+)>  
<!ATTLIST History  
  restriction %attvals.restriction; #IMPLIED>  

<!ELEMENT HistoryItem (DateTime, IncidentID?, Description+)>  
<!ATTLIST HistoryItem  
  type %attvals.historycat; #IMPLIED  
  restriction %attvals.restriction; #IMPLIED>  

<!--  
====================================================================  
===  Expectation class                                           ===  
====================================================================  
-->  
<!ELEMENT Expectation (Description+, Contact?, StartTime?, EndTime?)>  
<!ATTLIST Expectation  
  priority %attvals.priority; #IMPLIED  
  restriction %attvals.restriction; #IMPLIED  
  category %attvals.expectcat; #IMPLIED>  

<!--  
====================================================================  
===  Method class                                                ===  
===    - Classification                                          ===  
====================================================================  
-->  
<!ELEMENT Method (Classification*, Description*)>  
<!ATTLIST Method  
  restriction %attvals.restriction; #IMPLIED>  

<!ELEMENT Classification (name, url?)>  
<!ATTLIST Classification  
  restriction %attvals.restriction; #IMPLIED>
origin %attvals.origin; "other"
>
<![--
====================================================================
===  Assessment class                                          ===
===    - Impact
===    - TimeImpact
===    - MonetaryImpact
===    - LifeImpact
===    - Confidence
====================================================================
-->
<!ELEMENT Assessment (Impact*, TimeImpact*, MonetaryImpact*, LifeImpact*, Confidence?)>
<!ATTLIST Assessment
restriction %attvals.restriction; #IMPLIED
>
<!ELEMENT Impact (#PCDATA)>
<!ATTLIST Impact
restriction %attvals.restriction; #IMPLIED
severity %attvals.severity; #IMPLIED
completion %attvals.completion; #IMPLIED
type %attvals.impacttype; "unknown"
lang NMTOKEN #IMPLIED
>
<!ELEMENT TimeImpact (#PCDATA)>
<!ATTLIST TimeImpact
restriction %attvals.restriction; #IMPLIED
severity %attvals.severity; #IMPLIED
unit (labor | elapsed | downtime) #REQUIRED
metric (days | hours | minutes | seconds) "hours"
>
<!ELEMENT MonetaryImpact (#PCDATA)>
<!ATTLIST MonetaryImpact
restriction %attvals.restriction; #IMPLIED
severity %attvals.severity; #IMPLIED
currency CDATA #REQUIRED
>
<!ELEMENT LifeImpact (#PCDATA)>
<!ATTLIST LifeImpact
restriction %attvals.restriction; #IMPLIED
severity %attvals.severity; #IMPLIED
>>
metric (deaths | injuries) #REQUIRED
>
<!ELEMENT Confidence EMPTY>
<!ATTLIST Confidence
  rating %attvals.rating; #REQUIRED
>
<--
====================================================================
=== EventData class                                              ===
====================================================================
-->
<!ELEMENT EventData (Description*, Contact*, ReportTime?, DetectTime?, StartTime?, EndTime?,
Internet-Draft IODEF Data Model and Implementation September 2003

<!ATTLIST EventData
  restriction %attvals.restriction; #IMPLIED
>
<--
====================================================================
===  System class                                                ===
===  Note. Represents merged Source and Target classes of IDMF
===  (sections 4.2.4.3, 4.2.4.4)
====================================================================
-->
<!ELEMENT System (Node, User*, Process*, Service*, FileList?)>
<!ATTLIST System
  category %attvals.systemcat; #IMPLIED
  spoofed %attvals.spoofed; "unknown"
  interface CDATA #IMPLIED
  restriction %attvals.restriction; #IMPLIED
>
<--
====================================================================
===  FileList class               IDMEF (4.2.7.5)        ===
===    - File
===    - access-time
===    - change-time
===    - create-time
===    - modify-time
===    - c-major-device
===    - c-minor-device
===    - data-size
===    - disk-size
====================================================================

Internet-Draft IODEF Data Model and Implementation September 2003

category %attvals.nodecat; "unknown"
>
<!ELEMENT Address (address, netmask?)>
<!ATTLIST Address
ident CDATA "0"
category %attvals.addrcat; "unknown"
 vlan-name CDATA #IMPLIED
 vlan-num CDATA #IMPLIED
>
<!ELEMENT address (#PCDATA)>
<!ELEMENT netmask (#PCDATA)>
<!ELEMENT Location (#PCDATA)>
<!ATTLIST Location
 lang NMTOKEN #IMPLIED
>
<!ELEMENT NodeRole (#PCDATA)>
<!ATTLIST NodeRole
category %attvals.noderolecat; "other"
 lang NMTOKEN #IMPLIED
>
<!--
====================================================================
===  User class                                                 ===
====================================================================
-->
=== <!ELEMENT Process (name, pid?, path?, arg*, env*) >
====================================================================

<!ELEMENT Process (name, pid?, path?, arg*, env*)>
<!ATTLIST Process
  ident CDATA "0"
>
<!ELEMENT env (#PCDATA)>
<!ELEMENT pid (#PCDATA)>
<--
====================================================================

=== Service Class
===
---
  - port
  - portlist
  - protocol
  - SNMPService
  - WebService
    ---
      - url, cgi, arg, http-method
      - SNMPService
      ---
        - oid, community, command
===
IDMEF (4.2.7.4)
====================================================================
<--

<!ELEMENT Service (((name?, port), portlist), protocol?, SNMPService?, WebService?)>
<!ATTLIST Service
  ident CDATA "0"
>
<!ELEMENT port (#PCDATA)>
<!ELEMENT portlist (#PCDATA)>
<!ELEMENT protocol (#PCDATA)>
<--
=== Web Service ===
---

<!ELEMENT oid (#PCDATA)>  
<!ELEMENT community (#PCDATA)>  
<!ELEMENT command (#PCDATA)>  
<!--
====================================================================
===  Record class                                                ===
===    - RecordData
===    - Analyzer
===    - RecordItem
====================================================================
-->  
<!ELEMENT Record (RecordData+)>  
<!ATTLIST Record
    restriction %attvals.restriction; #IMPLIED
>  
<!ELEMENT RecordData (Description*, DateTime?, Analyzer?, RecordItem+)>  
<!ATTLIST RecordData
    ident CDATA "0"
    restriction %attvals.restriction; #IMPLIED
>  
<!--Element Analyzer of IODEF is re-used from IDMEF (4.2.4.1) -->  
<!ELEMENT Analyzer (Node?, Process?)>  
<!ATTLIST Analyzer
    analyzerid CDATA "0"
    manufacturer CDATA #IMPLIED
    model CDATA #IMPLIED
    version CDATA #IMPLIED
    class CDATA #IMPLIED
    ostype CDATA #IMPLIED
    osversion CDATA #IMPLIED
>  
<!ELEMENT RecordItem ANY>  
<!ATTLIST RecordItem
    dtype %attvals.dtype; #REQUIRED
>  
<!--
====================================================================
=== Simple classes containing multilingual content                  ===
===    - Description
===    - Contact.Name
====================================================================
-->
<--
<!ELEMENT Description ANY>
<!ATTLIST Description
  preserve %attvals.preserve; #IMPLIED
  transform %attvals.transform; #IMPLIED
  lang NMTOKEN #IMPLIED>
>
<!ELEMENT Name ANY>
<!ATTLIST Name
  preserve %attvals.preserve; #IMPLIED
  transform %attvals.transform; #IMPLIED
  lang NMTOKEN #IMPLIED>
>
<!--
====================================================================
=== Miscellaneous simple classes                                 ===
===   - path
===   - url
====================================================================
-->
<!ELEMENT name (#PCDATA)>
<!ATTLIST name
  lang NMTOKEN #IMPLIED>
>
<!ELEMENT path (#PCDATA)>
<!ELEMENT url (#PCDATA)>

<!--
====================================================================
=== Attribute list declarations.                      ===
===            for the many element-specific attribute lists.    ===
====================================================================
-->

| Attributes of the IODEF element. In general, the fixed value
| of this attribute will change each time a new version of
| the DTD is released.
-->

<!--
====================================================================
=== SECTION 2. Attribute value declarations. Enumerated values ===
=== for the many element-specific attribute lists.               ===
====================================================================
-->

| Defines purpose of the Incident
-->

<!ENTITY % attvals.purpose "
  ( handling | statistics | warning | other )
"">
<!--
| Defines restriction on access to an element's content
-->
<!ENTITY % attvals.restriction "
( default | public | need-to-know | private ) "">
<!--
| Values for the Expectation.expectcat attributes
-->
<!ENTITY % attvals.expectcat "
( nothing | contact-site | contact-me | block | other ) "">
<!--
| Values for the AdditionalData.type attribute.
-->
<!ENTITY % attvals.adtype "
( boolean | byte | character | date-time | integer |
  ntpstamp | portlist | real | string | xml ) "">
<!--
| Values for the RecordItem.type attribute
-->
<!ENTITY % attvals.dtype "
( boolean | byte | character | date-time | integer |
  ntpstamp | portlist | real | string | file | path | url |
  xml ) "">
<!--
| Values for the History.type attribute.
-->
<!ENTITY % attvals.historycat "
( triaged | notification | shared-info | received-info |
  remediation | other ) "">
<!--
| Values for the Address.category attribute.
-->
<!ENTITY % attvals.addrcat "
( unknown | atm | e-mail | lotus-notes | mac | sna | vm |
  ipv4-addr | ipv4-addr-hex | ipv4-net | ipv4-net-mask |
  ipv6-addr | ipv6-addr-hex | ipv6-net | ipv6-net-mask ) "">
<!--
| Values for the Id.type attribute.
-->

<!-- Values for the Impact.completion attribute. -->
<!ENTITY % attvals.completion "
  ( failed | succeeded )">

<!-- Values for the File.category attribute. -->
<!ENTITY % attvals.filecat "
  ( current | original )">

<!-- Values for the Impact.type attribute. -->
<!ENTITY % attvals.impacttype "
  ( none | admin | dos | file | recon | user | unknown | other )">

<!-- Values for the Linkage.category attribute. -->
<!ENTITY % attvals.linkcat "
  ( hard-link | mount-point | reparse-point | shortcut | stream | symbolic-link )">

<!-- Values for the RegistryHandle.type attribute. -->
<!ENTITY % attvals.registrytype "
  ( internic | apnic | arin | lacnic | ripencc | ti | local )">

<!-- Values for the Confidence.rating attribute. -->
<!ENTITY % attvals.rating "
  ( low | medium | high | numeric | unknown )
"
Values for the Impact.severity attribute.

Values for the Node.category attribute.

Values for the NodeRole.category attribute.

Values for the Classification.origin attribute.

Values for the User.category attribute.

Values for the System.spoofed attribute.
9. Security considerations

Due to the sensitive nature of some of the data that might be represented in the IODEF, the integrity, confidentiality, and non-repudiation of these documents in transit SHOULD be ensured. Although this protection can be provided by the transport mechanism, applying this security to an instance of the IODEF itself is RECOMMENDED. However, the specific protective measures applied to an IODEF document (be in through XML or the underlying transport protocol) should be driven by the requirements of the collaborators.

The applied protective measures MUST use cryptographic techniques. XML Digital Signatures [16] SHOULD be used for ensuring integrity and non-repudiation, and XML Encryption [17] SHOULD be used to ensure the confidentiality of an IODEF document. Examples using signatures and encryption on an IODEF document can be found in the Examples chapter (Section 7):
o IODEF-Document with XML signature (Section 7.2)

o IODEF-Document encrypted using XML encryption (Section 7.3)

o IODEF-Document encrypted and signed using XML signature & encryption (Section 7.4)

Information on the implementation-specifics of applying XML Digital Signatures and XML Encryption to an IODEF-Document can be found in the IODEF Implementation Guide [20].

When using cryptographic techniques the issue of key management (whether symmetric or public key cryptography is used) must be addressed.

Overall security measures must be applied to secure the IODEF-Document processing environment. The definition of these measures is outside the scope of this memo.
11. Acknowledgments

The following groups contributed substantially to this document and should be recognized for their efforts. This document would not exist
without their help:

- the Incident Object Description and Exchange Format Working-Group of the TERENA task-force (TF-CSIRT)
- the eCSIRT.net project
Normative References


[15] International Organization for Standardization, "International


Informative References


Authors' Addresses

Jan Meijer
SURFnet bv
P.O. Box 19035
Utrecht NL-3501 DA
Netherlands

Phone: +31 30 2 305 305
EMail: jan.meijer@surfnet.nl

Roman Danyliw
CERT Coordination Center
4500 Fifth Ave.
Pittsburgh, PA 15213
USA

Phone: +1 412 268 7090
EMail: rdd@cert.org

Yuri Demchenko
Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement

Copyright (C) The Internet Society (2003). All Rights Reserved.