The Incident Object Description Exchange Format Data Model and XML Implementation
draft-ietf-inch-iodef-07.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on December 10, 2006.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

The purpose of the Incident Object Description Exchange Format (IODEF) is to define a data representation that provides a framework for sharing information commonly exchanged by Computer Security Incident Response Teams (CSIRTs) about computer security incidents.
The IODEF satisfies the requirements specified in RFCXXX [17].

This Internet-Draft describes a data model for representing 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 .................................................. 5
   1.1. Terminology .............................................. 5
   1.2. Overview ................................................ 5
   1.3. About the IODEF Data Model .............................. 5
   1.4. About the IODEF Implementation .......................... 6
   1.5. Related Work ............................................ 7
2. Formatting Issues .............................................. 8
   2.1. IODEF XML Documents ..................................... 8
      2.1.1. The Document Prolog ............................... 8
      2.1.2. Languages in the IODEF ............................ 9
   2.2. IODEF Data Types ......................................... 9
      2.2.1. Integers ........................................... 9
      2.2.2. Real Numbers ....................................... 9
      2.2.3. Characters and Strings ............................ 10
      2.2.4. Multilingual Strings ............................... 10
      2.2.5. Bytes ................................................ 10
      2.2.6. Hexadecimal Bytes .................................. 10
      2.2.7. Enumerated Types ................................... 11
      2.2.8. Date-Time Strings .................................. 11
      2.2.9. Timezone string .................................... 11
      2.2.10. Port Lists ......................................... 11
      2.2.11. Postal Address .................................... 11
      2.2.12. Person or Organization ............................. 12
      2.2.13. Telephone and Fax Numbers .......................... 12
      2.2.14. Email string ....................................... 12
      2.2.15. Uniform Resource Identifier strings ............... 12
      2.2.16. Unique Identifiers ................................ 12
3. The IODEF Data Model .......................................... 13
   3.1. IODEF-Document class .................................... 13
   3.2. Incident class ........................................... 13
   3.3. IncidentID class ......................................... 16
   3.4. AlternativeID class ..................................... 17
3.5. RelatedActivity class .................................. 18
3.6. AdditionalData ........................................... 18
3.7. Contact class .............................................. 20
  3.7.1. RegistryHandle class ................................ 23
3.8. Time classes .............................................. 24
  3.8.1. StartTime ............................................. 24
  3.8.2. EndTime ............................................... 24
  3.8.3. DetectTime ............................................ 24
  3.8.4. ReportTime ............................................ 24
  3.8.5. DateTime ............................................... 24
3.9. Method class .............................................. 24
  3.9.1. Classification class ................................ 25
3.10. Assessment class ........................................ 26
  3.10.1. Impact class ......................................... 27
  3.10.2. TimeImpact class .................................... 28
  3.10.3. MonetaryImpact class ................................ 30
  3.10.4. Confidence class .................................... 31
3.11. History class ............................................ 32
  3.11.1. HistoryItem class ................................... 32
3.12. EventData class ........................................ 33
  3.12.1. Relating the Incident and EventData classes ....... 35
  3.12.2. Cardinality of EventData ............................ 35
3.13. Expectation class ........................................ 36
3.14. Flow class ............................................... 39
3.15. System class ............................................. 39
3.16. Node class ............................................... 41
  3.16.1. Counter class ........................................ 42
  3.16.2. Address .............................................. 43
  3.16.3. NodeRole class ....................................... 44
3.17. Service class ............................................ 45
  3.17.1. Application class .................................... 47
3.18. OperatingSystem class ................................... 48
3.19. Record class ............................................. 48
  3.19.1. RecordData class .................................... 49
  3.19.2. RecordPattern class .................................. 49
  3.19.3. RecordItem class .................................... 51
4. Extending the IODEF ........................................ 52
  4.1. Extending the data model ............................... 52
  4.2. Extending the XML Schema ............................... 52
5. Processing Considerations .................................. 56
6. Internationalization issues ................................ 57
7. Examples ......................................................... 58
  7.1. Code Red detection notification ......................... 58
  7.2. IODEF-Document with XML signature ..................... 60
  7.3. IODEF-Document encrypted using XML encryption .......... 60
  7.4. IODEF-Document encrypted and signed using XML
       signature & encryption .................................. 60
8. The IODEF Document Schema .................................. 61
9. Security considerations ....................................... 78
10. IANA considerations ......................................... 79
11. Acknowledgments ............................................. 80
12. References .................................................. 81
    12.1. Normative References .................................. 81


Internet-Draft IODEF Data Model June 2006

    12.2. Informative References ............................... 82
Authors' Addresses ............................................ 83
Intellectual Property and Copyright Statements ................. 84
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 [4].

Definitions for some of the common computer security-related terminology used in this document can be found in Section 2 of [17].

1.2. Overview

The Incident Object Description Exchange Format (IODEF) is a format for representing computer security information exchanged between Computer Security Incident Response Teams (CSIRTs). It provides a transport representation conforming to the requirements specified in [17], the Requirements for Format for Incident Report Exchange.

The overriding purpose of the IODEF is to expand and enhance the operational capabilities of CSIRTs. Community adoption of the IODEF provides an improved ability to resolve incidents by simplifying
collaboration and data sharing. This structured format provided by the IODEF allows for:

- increased automation in processing of incident data since the resources of security analysts to parse free-form textual documents will be reduced;
- decreased effort in normalizing similar data (even when highly structured) from different sources; and
- a common format on which to build interoperable tools for incident handling and subsequent analysis specifically when data comes from multiple constituencies.

Terminology, notation, and conventions of the data model and XML Schema are presented in Sections 2. The data model is described in Section 3, and the implementation considerations are covered in Sections 4 through 6. Section 7 provides several examples of IODEF documents. Section 8 formally specifies the XML Schema implementation of the data model. Sections 9 and 10 address the security and IANA considerations, respectively.

1.3. About the IODEF Data Model

The IODEF data model is a data representation that provides a framework for sharing information commonly exchanged by CSIRTs about computer security incidents. A number of considerations were made in the design of the data model.

- The intent of the data model is to support the automated processing of incident data. Hence, little consideration was made to ensure human-readability. Despite the still prevalent practice of manual incident report generation, this model is sufficiently complex that it will be unwieldy to create and process without software.
- The data model serves as a transport format. Therefore, its specific representation is not the optimal representation for on-disk storage, long-term archiving, or in-memory processing.
- Since there is no precise, widely agreed upon definition for an
incident, the data model does not attempt to dictate one through its implementation. Rather, a broad understanding is assumed that is flexible enough to encompass most of the CSIRT community.

- Describing an incident for all definitions would require an incredibly complex data model. Therefore, the IODEF data model only intends to be a framework to convey commonly exchanged incident information. However, it ensures that there are ample mechanisms for extensibility to support organization-specific information, and techniques to reference information kept outside of the explicit data model.

- Incidents have a life-cycle that dictates the exact type, quantity, and detail of the data that will be present at a given time (e.g., newly reported incidents may only contain the most rudimentary details, but closed incidents may contain a detailed analysis). The data model deals with this situation.

- Communication and coordination are central to the role of a CSIRT. Hence, tracking the source of all data is central to handling the incident. Therefore, the data model provides ways to explicitly bind information to a source, and accommodates differences in the types of parties involved in the incident (e.g., varying levels of confidence in information, different data sharing arrangements).

1.4. About the IODEF Implementation

The IODEF implementation uses the Extensible Markup Language (XML) [1], specifies an XML Schema, and registers an application-specific namespace [2].

For clarity in this document, the terms "XML" and "XML documents" will be used when referring to the Extensible Markup Language (XML).
o all the necessary features to define and extend a specific markup language for describing security incidents;

o a well understood technique for supporting internationalization and localization;

o a base of related technologies such as XSL [3], XPATH, and XML-SIG that the aid in the manipulation and use of the incident data; and

o a broad community of developers who already understand how to build systems around data exchanged in this format.

While XML provides a useful implementation language for IODEF, this implementation also dictates several limitations.

o XML is a text representation making it inherently inefficient either when binary data must be embedded or very large volumes of data must be exchanged.

o The data model is designed as a transport representation, and the use of XML further reinforces the inefficiency of using the IODEF for other purposes. Due to the overhead of the parser, XML is not an optimal in-memory representation. Furthermore, storing, searching, and retrieving native XML documents is problematic on a large scale dictating that this format is also a poor choice as a storage and archive format.

1.5. Related Work

The IODEF is only one of several security relevant data representations being standardized. Specifically, the complementary nature of the Intrusion Detection Message Exchange Format [19] bears mention given that many incidents represented in the IODEF may have first been discovered through the use of intrusion detection system output formatted according to the IDMEF. Given this relationship, the IODEF data model makes use of certain classes defined in the IDMEF data model.
2.1. IODEF XML Documents

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

This section describes the XML notations and conventions used in this document and explains particular issues related to using them to describe the IODEF data model and syntax. For readers unfamiliar with these notations, [18] 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 MUST begin with an XML declaration, and MUST specify the XML version used. If UTF-8 encoded is not used, the character encoding MUST also be explicitly specified.

The XML declaration with no character encoding will read as follows:

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

When a character encoding is specified, the XML declaration will read like the following:

```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 [7].

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).

2.1.1.2. IODEF Namespace

Each IODEF document must use the IODEF namespace "iodef" as follows:
2.1.2. Languages in the IODEF

IODEF messages SHOULD specify the language in which their contents are encoded. In general, the language can be specified with the attribute "language" that has reserved type "xs:lang" in the top-level element and letting all other elements "inherit" that definition.

The valid language codes for the "xs:lang" are described in [RFC 3066][5]. If no language is specified, English "en-US" SHOULD be assumed.

For the IODEF classes that support free-form text in a language that differ from the rest of the document, this language can be specified by local attribute "xs:lang".

2.2. IODEF Data Types

The IODEF data model defines a number of 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".

The INTEGER data type is implemented as an "xs:integer" in Schema

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.

The REAL data type is implemented as an "xs:float" in Schema.

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 to represent special characters.

The CHARACTER and STRING data types are implement as an "xs:string" in Schema.

2.2.4. Multilingual Strings

STRING data that represents multi-character attributes in a language different than the default encoding of the document are of the ML_STRING data type.

The ML_STRING data type is implemented as an "xs:string" in Schema. Likewise, all elements that are of this type also have a corresponding "lang" attribute to dictate the language per Section 2.1.2.

2.2.5. Bytes

Binary octets encoded using character code references (see ) is represented by the BYTE (and BYTE[]) data type.

The BYTE data type is implemented as an "xs:string" in Schema.
2.2.6. Hexadecimal Bytes

Binary octets encoded using a notation where each octet is encoded as a character tuple consisting of two hexadecimal digits is represented by the HEXBIN (and HEXBIN[]) data type.

The HEXBIN data type is implemented as an "xs:string" in Schema.

2.2.7. 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 Schema, the enumerated type keywords are used as attribute values.

The ENUM data type is implemented as a series of "xs:NMTOKEN" in Schema.

2.2.8. 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.


The DATETIME data type is implemented as an "xs:dateTime" in Schema.

2.2.9. Timezone string

A timezone offset from UTC is represented by the TIMEZONE data type. It is formatted according to the following regular expression: "Z|[-+\d:]+(\d|0[0-9]|1[0-4]):[0-5][0-9]".

The TIMEZONE data type is implemented as an "xs:string" with a regular expression constraint in Schema. This regular expression is identical to the timezone representation implemented in an "xs:dateTime".
2.2.10. 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".

The PORTLIST data type is implemented as an "xs:string" in Schema.

2.2.11. Postal Address

A postal address is represented by the POSTAL data type. This data type is an ML_STRING whose format is documented in Sections 6.27 of RFC 2252 [9].

The POSTAL data type is implemented as an "xs:string" in Schema.

2.2.12. Person or Organization

The name of an individual or organization is represented by the NAME data type. This data type is an ML_STRING whose format is documented in Section 5.4 of RFC 2256 [8].

The NAME data type is implemented as an "xs:string" in Schema.

2.2.13. 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 6.30 of RFC 2252 [8].

The PHONE data type is implemented as an "xs:string" in Schema.

2.2.14. 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 [10].

The EMAIL data type is implemented as an "xs:string" in Schema.

2.2.15. 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 [6].

The URI data type is implemented as an "xs:string" in Schema.

2.2.16. 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 alphanumeric strings.

The UID and GUID data types are implemented as an "xs:string" in Schema.

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 with other classes will be depicted with UML.

3.1. IODEF-Document class

The IODEF-Document class is the top level class in the IODEF data model. All IODEF documents are an instance of this class.

+-----------------+
| IODEF-Document  |
+-----------------+
| STRING version  |<--{1..*}--[ Incident ]
| ENUM lang       |<--{0..*}--[ ds:Signature ]
Figure 1: IODEF-Document class

The aggregate class that constitutes IODEF-Document is:

Incident
   One or more. The information related to a single incident.

Signature
   Zero or more. Cryptographic signature per [13] to ensure the integrity and authenticity of the document.

The IODEF-Document class has two attribute:

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

lang
   Required. ENUM. A valid language code per RFC 3066 [5]. The interpretation of this code is described in Section 2.1.2.

3.2. Incident class

Every incident is represented by an instance of the Incident class. This class provides a standardized representation for commonly exchanged incident data and associates a CSIRT assigned unique identifier with the described activity.
Figure 2: the Incident class

The aggregate classes that constitute Incident are:

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

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

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

DetectTime
Zero or one. The time the incident was first detected.

StartTime
Zero or one. The time the incident started.

EndTime
Zero or one. The time the incident ended.

ReportTime
One. The time the incident was reported.

Description
Zero or more. ML_STRING. A free-form textual description of the
incident.
The Incident class has three attributes:

- **purpose**
  - Required. ENUM. The purpose attribute represents the reason why the IODEF document was created. It is closely related to the Expectation class ([Section 3.13](#)). This attribute is defined as an enumerated list:
    1. traceback. The document was sent for trace-back purposes;
    2. mitigation. The document was sent to request aid in mitigating the described activity;
    3. reporting. The document was sent to comply with reporting requirements;
    4. other. The document was sent for purposes specified in the Expectation class.

---

lang
  Required. ENUM. A valid language code per RFC 3066 [5].

restriction
  Optional. ENUM. This attribute indicates the disclosure guidelines to which the sender expects the recipient of the IODEF-Document to adhere. Naturally, this provides no real security since 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 (i.e., children of the Incident class) have a restriction attribute. Therefore, a child can override the guidelines of a parent class, be it to restrict 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 that 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 are no restrictions placed in 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 3: the IncidentID class

The IncidentID class has one attribute:

name

Required. GUID. An identifier for the CSIRT that created the
IODEF-Document. In order to have a globally unique CSIRT name,
the domain name (DNS) of the CSIRT MUST be used.

3.4. AlternativeID class

The AlternativeID class lists the incident tracking numbers used by
other CSIRTs to refer to activity described in this IODEF document.
Thus, a tracking number listed as an AlternativeID references the
same incident detected by another CSIRT. The incident tracking
numbers of the CSIRT that generated the IODEF document should never
be considered an AlternativeID.

+------------------+
| AlternativeID    |
+------------------+
| ENUM restriction |<>--{1..*}--[ IncidentID ] |
|                  |
+------------------+

Figure 4: the AlternativeID class

The aggregate class that constitutes AlternativeID is:
IncidentID
One or more. The incident tracking number of another CSIRT.

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 lists the incident tracking numbers of incidents that are related to the one described in the IODEF document. These references may be 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.

+------------------+
| RelatedActivity  |
+------------------+
| ENUM restriction |<--{1..*}--[ IncidentID ]
|                 |
+------------------+

Figure 5: RelatedActivity class

The aggregate class that constitutes RelatedActivity is:

IncidentID
One or more. The incident tracking number of a related incident.

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 types (e.g., integers, strings) are provided with a mechanism to annotate their meaning. The class can also be used to extend the data model (and the associated Schema) to support proprietary extensions by encapsulating entire XML documents conforming to another Schema (e.g., IDMEF). A detailed discussion for extending the data model and the Schema can be found in Section 4.

Unlike XML, which is self-describing, atomic data must 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.

Figure 6: the AdditionalData class

The AdditionalData class has three attributes:

dtype
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

+------------------+
| AdditionalData   |
+------------------+
| ANY              |
|                  |
| ENUM dtype       |
| STRING meaning   |
| STRING formatid  |
| ENUM restriction |
+------------------+
strings "true" or "false"

2. byte. The element content is a single 8-bit byte (see Section 2.2.5);

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 2.2.8);

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

6. portlist. The element content is a port list (see Section 2.2.10);

7. real. The element content is a real number (see Section 2.2.2);

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

9. file. The element content is a base64 encoded binary file;

10. frame. The element content is a hexbin-encoded layer-2 frame (see Section 2.2.6)

11. packet. The element content is a hexbin-encoded layer-3 packet (see Section 2.2.6)

12. ipv4-packet. The element content is an IPv4 hexbin-encoded packet (see Section 2.2.6)

13. ipv6-packet. The element content is an IPv6 hexbin-encoded packet (see Section 2.2.6)

14. path. The element content is a filesystem path;

15. url. The element content is a URL (see Section 2.2.15);

16. csv. The element content is a common separated value list;
17. winreg. The element content is a Windows registry key;
18. xml. The element content is XML-tagged data (see Section 4);
19. other. The element content should be interpreted using the formatid attribute.

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

formatid
Optional. STRING. An identifier referencing the format and semantics of the encapsulated data.

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

3.7. Contact class

The Contact class describes contact information for organizations and personnel involved in the incident. This class allows for the naming of the involved party, specifying contact information for 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 Contact class is aggregated into the Contact class).

The 'type' attribute disambiguates the type of contact information being provided.

This recursive definition provides a way to relate information without requiring the explicit use of identifiers in the classes. For example, separate contact information for two individuals from the same organization would not require duplicating the organization information.

+------------------+
| Contact           |
+------------------+
Figure 7: The Contact class

The aggregate classes that constitute the Contact class are:

**ContactName**

Zero or one. ML_STRING. The name of the contact. The contact may either be an organization or a person. The type attribute disambiguates the semantics.

**Description**

Zero or many. ML_STRING. A free-form description of this contact. In the case of a person, this is often the organizational title of the individual.

**RegistryHandle**

Zero or many. A handle name in a registry.

**PostalAddress**

Zero or one. POSTAL. The postal address of the contact formatted according to Section 2.2.11.

**Email**

Zero or many. EMAIL. The email address of the contact formatted according to Section 2.2.14.

**Telephone**
Zero or many. PHONE. The telephone number of the contact formatted according to Section 2.2.13.

Fax
Zero or one. PHONE. The facsimile telephone number of the contact formatted according to Section 2.2.13.

Timezone
Zero or one. TIMEZONE. The timezone in which the contact resides formatted according to Section 2.2.9.

Contact
Zero or many. Contact instances contained within another Contact instance inherit the values of the parent(s). This recursive definition can be used to group common data pertaining to multiple points of contact, and is especially useful when listing multiple points of contact at the same organization. When Contact elements are defined recursively, only the leaf instances (those Contact instances not containing other Contact instances) represent actual points of contact.

The Contact class has three attributes:

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 handling of the incident.

type
Required. ENUM. Indicates the type of contact being described. This attribute is defined as an enumerated list:
1. person.
2. organization.

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

3.7.1. 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 registry    |
+------------------+
```

Figure 8: The RegistryHandle class

The RegistryHandle class has one attribute:

registry
  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. afrinic. African Internet Numbers Registry
7. local. A database local to the CSIRT.
3.8. Time classes

The data model uses five different classes to represent a timestamp. Their definition is identical, but each has a distinct name to convey a difference in semantics.

The element content of each class is a timestamp formatted according to the DATETIME data type (see Section 2.2.8).

+----------------------------------+
| {Start| End| Report| Detect}Time |
+----------------------------------+
| DATETIME                          |
+----------------------------------+

Figure 9: the Time classes

3.8.1. StartTime

The StartTime class represents the time the incident began.

3.8.2. EndTime

The EndTime class represents the time the incident ended.

3.8.3. DetectTime

The DetectTime class represents the time the first activity of the incident was detected.

3.8.4. ReportTime

The ReportTime class represents the time the incident was reported. This timestamp SHOULD coincide to the time at which the IODEF document is generated.

3.8.5. 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.
3.9. Method class

The Method class describes the methodology used by the intruder to perpetrate the events of the incident. This class can reference well-known vulnerability or exploit databases; the intruder tools used in the attack; and provide a free-form description of the activity.

+------------------+
| Method           |
+------------------+
| ENUM restriction |<>--{0..*}--[ Classification ]
|                  |<>--{0..*}--[ Description ]
+------------------+

Figure 10: The Method class

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. ML_STRING. A free-form text description of the methodology used by the intruder.

The Method class has one attribute:

restriction

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

3.9.1. 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 name in the database, and the URI to this entry.

+------------------+
Figure 11: The Classification class

The aggregate classes that constitute Classification:

name
   One. STRING. The key into the database specified in the origin attribute.

url
   Zero or many. URI. A URL to additional information about the vulnerability or exposure referenced by the name.

The Classification class has one attribute:

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. Mitre 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. A custom database whose URL is specified in the url class, and the name of the entry is specified in the name class.
3.10. Assessment class

The Assessment class describes the technical and non-technical repercussions of the incident on the CSIRT's constituency.

This class was derived from the IDMEF[19].

```
+------------------+
| Assessment       |
+------------------+
| ENUM restriction |<>--{0..*}--[ Impact         ]
|                  |<>--{0..*}--[ TimeImpact    ]
|                  |<>--{0..*}--[ MonetaryImpact ]
|                  |<>--{0..1}--[ Confidence    ]
+------------------+
```

Figure 12: Assessment class

The aggregate classes that constitute Assessment are:

Impact
Zero or many. Technical impact of the incident on a network.

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 financial loss.

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.
3.10.1. Impact class

The Impact class allows for categorizing and describing the technical impact of the incident on the network of an organization.

This class is based on the IDMEF [19].

+------------------+
| Impact           |
+------------------+
| ML_STRING        |
| ENUM severity    |
| ENUM completion  |
| ENUM type        |
+------------------+

Figure 13: Impact class

The element content will be a free-form textual description of the impact.

The Impact class has three attributes:

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 whether the described activity was successful. The permitted values are shown below. There is no
default value.

1. failed. The attempted activity was not successful.

2. succeeded. The attempted activity succeeded.

type
  Required. ENUM. Classifies the malicious activity into incident categories. The permitted values are shown below. The default value is "other".

1. admin. Administrative privileges were attempted.

2. dos. A denial of service was attempted.

3. file. An action on a file or database was attempted.

4. recon. Reconnaissance activity was attempted.

5. user. User privileges were attempted.

6. unknown. The classification of this activity is unknown.

7. other. Other activity not captured in the above categories.

3.10.2. TimeImpact class

The TimeImpact class describes the impact of the incident on an organization as a function of time. It provides a way to convey down time and recovery time.
The element content will be a numeric value (REAL) specifying a unit of time. The unit and metric attributes will imply the semantics of the element content.

The TimeImpact class has three attributes:

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.

duration
    Required. ENUM. Defines a unit of time, that when combined with the metric attribute, fully describes a metric of impact that will be conveyed in the element content. The permitted values are shown below. The default value is "hours".
1. second. Seconds.
2. minute. Minutes.
3. hour. Hours.
4. day. Days.
5. month. Month.
6. quarter. Quarter.
7. year. Year.

3.10.3. MonetaryImpact class

The MonetaryImpact class describes the financial impact of the activity on an organization. For example, this impact may consider losses 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 severity    |
| STRING currency  |
+------------------+
```

Figure 15: MonetaryImpact class

The element content will be a numeric value (REAL) specifying a unit of currency described in the currency attribute.

The MonetaryImpact class has two attributes:

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 [15]. There is no default value.

3.10.4. Confidence class

The Confidence class represents a best estimate of the validity and accuracy of the described impact (see Section 3.10) of the incident activity. This estimate can be expressed as a category or a numeric calculation.

This class is based upon the IDMEF [19]).

+------------------+
| Confidence       |
|                  |
| REAL             |
| ENUM rating      |
+------------------+

Figure 16: Confidence class

The element content expresses a numerical assessment in the confidence of the data when the value of the rating attribute is "numeric". Otherwise, this element should be empty.

The Confidence class has one attribute.

rating
Required. ENUM. A rating of the analytical validity of the specified Assessment. The permitted values are shown below. There is no default value.

1. low. Low confidence in the validity
2. medium. Medium confidence in the validity

3. high. High confidence in the validity

4. numeric. The element content contains a number that conveys the confidence of the data. The semantics of this number outside the scope of this specification.

3.11. History class

The History class is a log of the significant events or actions performed by the involved parties 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 17: The History class

The class that constitutes History is:

HistoryItem

One or many. Entry 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.

3.11.1. HistoryItem class
The HistoryItem class is an entry in the History (Section 3.11) log that documents a particular action or event that occurred in the course of handling the incident. The details of the entry are a free-form description, but each can be categorized with the type attribute.

The aggregate classes that constitute HistoryItem are:

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

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

Description
   One or many. ML_STRING. A free-form textual description of the action or event.

The HistoryItem class has two attributes:

+------------------+
| HistoryItem      |
+------------------+
| ENUM restriction |<>----------[ DateTime    ]
| ENUM action      |<>--{0..1}--[ IncidentId  ]
|                  |<>--{1..*}--[ Description ]

Figure 18: HistoryItem class
restriction
Optional. ENUM. This attribute has been defined in Section 3.2.

action
Optional. ENUM. Classifies a performed action or occurrence documented in this history log entry. As activity will likely have been instigated either through a previously conveyed expectation, or internal investigation, this attribute is identical to category attribute of the Expectation class. The different is only one of tense. When an action is in this class, it has been completed. See Section 3.13.

3.12. 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, and any forensic evidence discovered.

```
+------------------+
| EventData        |
+------------------+
| ENUM restriction |<>--{0..*}--[ Description ] |
|                   |<>--{0..1}--[ DetectTime ]  |
|                   |<>--{0..1}--[ StartTime ]   |
|                   |<>--{0..1}--[ EndTime ]     |
|                   |<>--{0..*}--[ Contact ]     |
|                   |<>--{0..1}--[ Assessment ]  |
|                   |<>--{0..*}--[ Method ]      |
|                   |<>--{0..*}--[ Flow ]        |
|                   |<>--{0..*}--[ Expectation ] |
|                   |<>--{0..1}--[ Record ]      |
|                   |<>--{0..*}--[ EventData ]   |
|                   |<>--{0..*}--[ AdditionalData ] |
```

Figure 19: The EventData class
The aggregate classes that constitute EventData are:

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

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

- **StartTime**
  - Zero or one. The time the event started.

- **EndTime**
  - Zero or one. The time the event ended.

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

- **Assessment**
  - Zero or one. The impact of the incident on the target and the actions taken.

- **Method**
  - Zero or more. The methodology used by the intruders.

- **Flow**
  - Zero or more. A description of the systems or networks involved.

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

- **Record**
  - Zero or one. Supportive data (e.g., log files) that provides additional information about the event.

- **EventData**
  - Zero or more. EventData instances contained within another EventData instance inherit the values of the parent(s); this recursive definition can be used to group common data pertaining
to multiple events. When EventData elements are defined recursively, only the leaf instances (those EventData instances not containing other EventData instances) represent actual events.

AdditionalData
Zero or one. An extension mechanism for data not explicitly represented in the data model.

The EventData class has one attribute:

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

3.12.1. Relating the Incident and EventData classes

There is substantial overlap in the Incident and EventData classes. Nevertheless, the semantics of these classes are quite different. The Incident class provides summary information about the entire incident, while the EventData class provides information about the individual events comprising the incident. In the most common case, the EventData class will provide more specific information for the general description provided in the Incident class. However, it may also be possible that the overall summarized information about the incident conflicts with some individual information in an EventData class when there is a substantial composition of various events in the incident. In such a case, the interpretation of the more specific EventData MUST supersede the more generic information provided in IncidentData.

3.12.2. Cardinality of EventData

The EventData class can be thought of as a container for 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. With an instance of the EventData class, hosts (i.e., System class) are grouped around these common properties.

The recursive definition (or instance property inheritance) of the EventData 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 or duplicating information. Instead, the relative depth (nesting) of a class is used to group (relate) information.

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 Flow class. It happens that there is a common technical contact (i.e., Contact class) for these two machines, but the impact (i.e., Assessment class) on them is different. A depiction of the representation for this situation can be found in Figure 20.

```
+------------------+
| EventData        |
+------------------+

| <>-----[ Contact ] |
| <>-----[ EventData ]<>-----[ Flow ] |
|                      [                   ]<>-----[ Assessment ] |
| <>-----[ EventData ]<>-----[ Flow ] |
|                      [                   ]<>-----[ Assessment ] |
```

Figure 20: Recursion in the EventData class

3.13. Expectation class

The Expectation class conveys to the recipient of the IODEF document the actions the sender is requesting.
Figure 21: the Expectation class

The aggregate classes that constitute Expectation are:

Description

One or many. ML_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 Incident 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 Expectations class has three attributes:

restriction

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

severity

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
action
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-source-site. Contact the site(s) identified as the source of the activity.

3. contact-target-site. Contact the site(s) identified as the target of the activity.

4. contact-sender. Contact the originator of the document.

5. investigate. Investigate the machine(s) listed in the document.

6. block-host. Block traffic from the machine(s) listed as sources in the document.

7. block-network. Block traffic from the network(s) listed as sources in the document.

8. block-port. Block the port listed as sources in the document.

9. rate-limit-host. Rate-limit the traffic from the machine(s) listed as sources in the document.

10. rate-limit-network. Rate-limit the traffic from the network(s) listed as sources in the document.

11. rate-limit-port. Rate-limit the port(s) listed as sources in the document.

12. remediate-other. Remediate the activity in a way other than by rate limiting or blocking.


14. status-new-info. Conveys that new information was received for this incident.

15. other. Perform some custom action described in the
3.14. Flow class

The Flow class groups the source and target hosts or networks (represented by System) in an event.

+------------------+
| Flow             |
+------------------+
|                  |<>--{1..*}--[ System ]
+------------------+

Figure 22: the Flow class

The aggregate class that constitutes Flow is:

System
   One or More. A host or network involved in the incident activity.

The Flow System class has no attributes.

3.15. System class

The System class represents a computer or network involved in the incident.

The systems represented by this class are categorized according to the role they played in the incident through the category attribute. The value of this category attribute dictates the semantics of the aggregated classes in the System class. If the category attribute has a value of 'source', then the aggregated classes denote the machine and service from which the activity is originating. With a category attribute value of 'target' or 'intermediary', then the machine 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.

Service
Zero or more. A network service running on the system.

OperatingSystem
Zero or one. The operating system running on the system.

Counter
Zero or more. A counter with which to summarize properties of this host or network.

The System class has four attribute:

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

category
Required. ENUM. Classifies the role the host or network played in the incident. 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 intermediary in the attack.

interface
Optional. STRING. Specifies the interface on which the event(s) on this System originated. If the Node class specifies a network rather than a host, this attribute has no meaning.
spoofed
Optional. ENUM. An indication of confidence in 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 attribute value is unknown

2. yes. The category attribute value 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 attribute value is believed to be correct.

3.16. Node class

The Node class identifies a host, network device, or network.

This class was derived from the IDMEF [19].

+---------------------+
|    Node      |
+---------------------+
| {0..1}--[ name     ]
| {0..*}--[ Address  ]
| {0..1}--[ Location ]
| {0..1}--[ DateTime ]
| {0..*}--[ NodeRole ]
| {0..*}--[ Counter  ]
+---------------------+

Figure 24: The Node class

The aggregate classes that constitute Node are:

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 hardware, network, or application address of the Node. Unless a name is provided, at least one address must be specified.

Location
Zero or one. ML_STRING. A free-from description of the physical location of the equipment.

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.

Counter
Zero or more. A counter with which to summarizes properties of this host or network.

3.16.1. Counter class

The Counter class summarize multiple occurrences of some event, or conveys counts or rates on various features (e.g., packets, sessions, events).

The value of the counter is the element content, with its units represented in the type attribute. A rate for a given feature can be expressed by setting the duration attribute. The complete semantics are entirely context dependant based on the class in which the Counter is aggregated.

+------------------+
| Counter          |
+------------------+
| REAL             |
|                  |
Figure 25: the Counter class

The Counter class has three attribute:

**type**
- Required. ENUM. Specifies the units of the element contents.
- 1. byte. Count of bytes.
- 3. flow. Count of flow (e.g., NetFlow records).
- 4. session. Count of sessions
- 5. alert. Count of notifications generated by another system (e.g., IDS or SIM).
- 6. message. Count of messages (e.g., mail messages).
- 7. event. Count of events
- 8. other. User defined count

**meaning**
- Optional. STRING. Describes the semantics of the element content if the type attribute is set to other.

**duration**
- Optional. ENUM. If present, the Counter class represents a rate rather than a count over the entire event. In that case, this attribute specifies the denominator of the rate (where the type attribute specified the nominator). The possible values of this attribute are defined in Section 3.10.2
3.16.2. Address

The Address class represents a hardware (layer-2), network (layer-3), or application (layer-7) address.

This class was derived from the IDMEF [19].

+------------------+
|     Address      |
+------------------+
| ENUM category    |
| STRING vlan-name |
| INTEGER vlan-num |
+------------------+

Figure 26: the Address class

The Address class has three attributes:

category
   Required. ENUM. The type of address represented. The permitted values for this attribute are shown below. The default value is "ipv4-addr".

1.   asn. Autonomous System Number
2.   atm. Asynchronous Transfer Mode (ATM) address
3.   e-mail. Electronic mail address (RFC 822)
4.   ipv4-addr. IPv4 host address in dotted-decimal notation (a.b.c.d)
5.   ipv4-net. IPv4 network address in dotted-decimal notation, slash, significant bits (a.b.c.d/nn)
6.   ipv4-net-mask. IPv4 network address in dotted-decimal notation, slash, network mask in dotted-decimal notation (a.b.c.d/w.x.y.z)
7. ipv6-addr. IPv6 host address
8. ipv6-net. IPv6 network address, slash, significant bits
9. ipv6-net-mask. IPv6 network address, slash, network mask
10. mac. Media Access Control (MAC) address

vlan-name
Optional. STRING. The name of the Virtual LAN to which the address belongs.

vlan-num
Optional. STRING. The number of the Virtual LAN to which the address belongs.

3.16.3. NodeRole class

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

+---------------+
| NodeRole      |
+---------------+
  +---------------+
  | ML_STRING     |
  | ENUM category |
  | ENUM lang     |
  +---------------+

Figure 27: The NodeRole class

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

The NodeRole class has two attributes:

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

lang
  Required. ENUM. A valid language code per RFC 3066 [5].

3.17. Service class

The Service class describes a network service of a host or network.
The service is identified by specific port or list of ports, along with the application listening on that port.

When Service occurs as an aggregate class of a System that is a source, then this service is the one from which activity of interest is originating. Conversely, when Service occurs as an aggregate class of a System that is a target, then that service is the one to which activity of interest is directed.

This class was derived from the IDMEF [19].

+---------------------+         +---------------------+
|   Service           |         |   Service           |
|---------------------|         |---------------------|
| INTEGER ip_version  |<--{0..1}--[ port      | INTEGER ip_version  |
| INTEGER ip_protocol |<--{0..1}--[ portlist  | INTEGER ip_protocol |
|                     |<--{0..1}--[ ProtoCode  |
|                     |<--{0..1}--[ ProtoType  |
|                     |<--{0..1}--[ ProtoFlags |
|                     |<--{0..1}--[ Application|

Figure 28: The Service class

The aggregate classes that constitute Service are:

port
   Zero or one. INTEGER. A port number.

portlist
   Zero or one. PORTLIST. A list of port numbers formatted according to Section 2.2.10.

ProtoCode
   Zero or one. INTEGER. A layer-4 protocol-specific code field.

ProtoType
   Zero or one. INTEGER. A layer-4 protocol specific type field.

ProtoFlags
Zero or one. INTEGER. A layer-4 protocol specific flag field.

**Application**

Zero or more. The application bound to the specified port or portlist.

The Service class must specify either a port or portlist.

The Service class has two attributes:

- `ip_version` Required. INTEGER. The IP version number.
- `ip_protocol` Required. INTEGER. The IANA protocol number.

### 3.17.1. Application class

The Application class describes an application running on a System providing a Service.

```
+--------------------+
|   Application      |
+--------------------+
 | STRING swid        |<>--{0..1}--[ url ]
 | STRING configid    |
 | STRING vendor      |
 | STRING family      |
 | STRING name        |
 | STRING version     |
 | STRING patch       |
+--------------------+
```

Figure 29: The Application class

The aggregate classes that constitute Application are:

- `url` Zero or one. URI. A uri describing the application.
The Application class has seven attributes:

- **swid**: Optional. STRING. An identifier that can be used to reference this software.
- **configid**: Optional. STRING. An identifier that can be used to reference a particular configuration.
- **vendor**: Optional. STRING. Vendor name.
- **family**: Optional. STRING. Family of the software.
- **name**: Optional. STRING. Name of the software.
- **version**: Optional. STRING. Version of the software.
- **patch**: Optional. STRING. Patch or service pack level.

### 3.18. OperatingSystem class

The OperatingSystem class describes the operating system running on a System. The definition is identical to the Application class ([Section 3.17.1](#)).

### 3.19. Record class

The Record class is a container class for log and audit data that provides supportive information about the incident. The source of this data will often 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 CSIRT believes an incident has occurred.
The aggregate class that constitutes Record is:

RecordData  
One or more. Log or audit data generated by a particular type of sensor. Separate instances of the RecordData class SHOULD be used for each sensor type.

The Record class has one attribute:

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

3.19.1. 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.

+------------------+
| RecordData       |<--{0..1}--[ DateTime        ]
|                  |<--{0..*}--[ Description     ]
|                  |<--{0..1}--[ Application     ]
|                  |<--{0..*}--[ RecordPattern   ]
|                  |<--{1..*}--[ RecordItem      ]
+------------------+
Figure 31: The RecordData class

The aggregate classes that constitutes RecordData is:

DateTime
Zero or one. Timestamp of the RecordItem data.

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

Application
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.

3.19.2. RecordPattern class

The RecordPattern class describes where in the content of the RecordItem relevant information can be found. It provides a way to reference subsets of information, identified by a pattern, in a large log file, audit trail, or forensic data.

+------------------+
| RecordPattern    |
+------------------+
| STRING           |
| ENUM type        |
| INTEGER offset   |
| ENUM offsetunit  |
| INTEGER instance |

Figure 32: The RecordPattern class

The specific pattern to search with in the RecordItem is defined in the body of the element. It is further annotated by four attributes:

type
  Required. ENUM. Describes the type of pattern that is being specified in the body of the element. The default is "regex".
  1. regex. POSIX regular expression
  2. binary. Binhex encoded binary pattern
  3. xpath. W3C XPath

offset
  Optional. INTEGER. Amount of units (determined by the offsetunit attribute) to seek into the RecordItem data before matching the pattern.

offsetunit
  Optional. ENUM. Describes the units of the offset attribute. The default is "line".
  1. line. Offset is a count of lines.
  2. binary. Offset is a count of bytes

instance
  Optional. INTEGER. Number of types to apply the specified pattern.

3.19.3. 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. The class supports both the
direct encapsulation of the data, as well as, provides primitives to reference data stored elsewhere.

This class is identical to AdditionalData class (Section 3.6).
4. Extending the IODEF

In order to support the changing activity of CSIRTS, the IODEF data model will need to evolve along with them. To allow new features to be added, both the data model and the Schema can be extended as described in this section. As these extensions mature, they can 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 Schema (see Section 4.2) which set limits on where extensions to the data model may be made.

4.2. Extending the XML Schema

XML Schema provides a flexible way of extending the IODEF data model by defining extension schemas in a separate namespace.

The following guidelines MUST be followed when extending the IODEF Schema with another schema:

1. The IODEF extension Schema MUST include an extension namespace definition and provide a reference to this schema's location per the XML Schema specification:

```
<xs:schema targetNamespace="http://iana.org/iodef-ext1"
    xmlns:iodef-ext1="http://iana.org/iodef-ext1"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified">
```

2. The import of the base namespace and declaration of the base IODEF schema can be added to the extension Schema
<xs:schema targetNamespace="http://iana.org/iodef-ext1"
    xmlns:iodef-ext1="http://iana.org/iodef-ext1"
    xmlns:iodef="http://iana.org/iodef"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified"/>

3. The location of the extension schema should be referenced in the XML document that uses it.

<IODEF-Document xmlns:iodef-ext1="http://iana.org/iodef-ext1"
    xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://iana.org/iodef-ext1
    http://iana.org/iodef-ext1/ietf-inch-iodef-ext1.xsd">

4. It is RECOMMENDED that all extensions start with "iodef-" prefix and add specific extension abbreviation such as "ext1".

5. It may be convenient to add a reference to the extension schema and import this extension namespace to the base IODEF schema.

Elements defined in the extension schema can be used in any place in final IODEF document. In the example below, the "iodef-xws" extension is defined by the schema that contains one element "iodef-xws:Principal". This element is composed of the NameIdentifier element of XML type NCName and imports "iodef:Description" element from the master IODEF schema.
In the example below, the above defined extension is used in the iodef:System element.
<IODEF-Document xmlns:iodef="http://iana.org/iodef"
xmlns:iodef-xws="http://iana.org/iodef-xws"
xsi:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iana.org/iodef
http://iana.org/iodef-xws/draft-ietf-inch-iodef-1.0.xsd"
xsi:schemaLocation="http://iana.org/iodef-xws
http://iana.org/iodef-xws/draft-ietf-inch-iodef-xws.xsd" version="1.0">
  <Incident restriction="private" purpose="traceback">
    <IncidentID Issuer="String" restriction="default">Text</IncidentID>
  </Incident>

  <iodef:System restriction="default" interface="VLAN" systemcat="source" spoofed="unknown">
  </iodef:System>

  <iodef-xws:Principal iodef-xws:principalcat="other">
    <NameIdentifier>CN=Joe Smith, OU=AIRG, O=UvA, S=NH, L=Holland, C=NL</NameIdentifier>
  </iodef-xws:Principal>

</iodef:System>
5. Processing Considerations

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

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 validate document where element or attribute values do 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-
6. Internationalization issues

Internationalization and localization is of specific concern to the IODEF, since it is only through collaboration, often across language barriers, that certain incidents be resolved. The IODEF supports this goal by depending on XML constructs, and through explicit design choices in the data model.

The IODEF leverages that XML natively supports different character encodings that is specified for whole document. The default encoding is UTF-8 whereby allowing information encoded in an IODEF document to be in all languages that are supported by UCS/Unicode. In order to disambiguate the explicit language on a per-element basis, the xs:language attribute is used.
For the languages that do not use UTF-8 encoding (e.g., Chinese Big5 or Japanese ISO-2022-JP), the IODEF schema uses the MultilingTextType type that allows a binary transformation of non-UTF-8 encoded text.

The intent of the data model was to provide internationalization and localization, but not to the detriment of inter-operability. While IODEF does support different languages, the data model also relies heavily on standardized enumerated attributes that can crudely approximate the contents of the document. With this approach, a CSIRT should be able to make some sense of an IODEF document it might receive that uses a language unfamiliar to its analysts.

Likewise, the data model was designed so that classes where free-text might be used for descriptive purposes always have a one-to-many cardinality with its parent (i.e., Description class). The primary intent of this design was to allow the same description to be repeated in another instance of the class but in a different language. This approach allows recipients speaking different languages to receive the identical document, but allows the IODEF parser to select the appropriate language.

7. Examples

This section provides representative examples of incident data converted to an IODEF document.

7.1. Code Red detection notification
The following email message is a typical example of an incident report where one host is infected with a worm. The original report sent by email is presented in Figure 38, and the corresponding equivalent as an IODEF document is shown below.

From e-citizen@domain.com
Date: 13 Sep 2001 23:19:24 -0000
To: cert-domain@domain.com
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. There have been 57 separate instances of this activity detected from our web server logs (Port 80):

From our web server logs (Port 80):

Figure 38: Code Red detection notification: initial report

<?xml version="1.0" encoding="UTF-8"?>
<IODEF-Document version="1.00"
xmlns="draft-ietf-inch-iodef-070.xsd"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="draft-ietf-inch-iodef-070.xsd">

<Incident purpose="reporting">
   <IncidentID name="CERT-DOMAIN.COM">CERT-DOMAIN.COM#189</IncidentID>
   <ReportTime>2001-09-13T23:19:24+00:00</ReportTime>
</Incident>


Internet-Draft IODEF Data Model June 2006
<Description>Host sending out Code Red probes</Description>

<Assessment>
  <Impact completion="failed" type="admin"/>
</Assessment>

<Contact role="creator" type="organization">
  <ContactName>CERT-FOR-OUR-DOMAIN.PL</ContactName>
  <Email>cert-for-our-domain.pl@ourdomain.pl</Email>
</Contact>

<Contact role="tech" type="organization">
  <ContactName>Constituency-contact for 10.1.1.2</ContactName>
  <RegistryHandle registry="apnic">example-foo</RegistryHandle>
  <Email>Constituency-contact@10.1.1.2.pl</Email>
</Contact>

<EventData>
  <Flow>
    <System category="source">
      <Node>
        <Address category="ipv4-addr">10.1.1.2</Address>
        <Counter type="event">57</Counter>
      </Node>
    </System>

    <System category="target">
      <Node>
        <Address category="ipv4-net">10.5.0.0/16</Address>
      </Node>
      <Service ip_version="4" ip_protocol="6">
        <port>80</port>
      </Service>
    </System>
  </Flow>

  <Expectation action="block-host">
    <Description>Track and clean host</Description>
  </Expectation>

  <Record>
    <RecordData>
      <DateTime>2001-09-13T18:11:21+02:00</DateTime>
      <Description>Web-server logs</Description>
      <RecordItem dtype="string">
      </RecordItem>
      <RecordItem dtype="url">http://mydomain.com/logs/httpd_access</RecordItem>
    </RecordData>
  </Record>
</EventData>

<History>
  <HistoryItem action="contact-source-site">
    <DateTime>2001-09-14T08:19:01+00:00</DateTime>
    <Description>Notification sent to constituency-contact@10.1.1.2</Description>
  </HistoryItem>
</History>
Figure 39: 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 Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:iodef="draft-ietf-inch-iodef-070.xsd"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    targetNamespace="draft-ietf-inch-iodef-070.xsd"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified">
    <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
        schemaLocation="http://www.w3.org/TR/xmldsig-core/xmldsig-core-schema.xsd"/>
    <!--
    ********************************************************************
    ********************************************************************
    *** Incident Object Description and Exchange Format XML Schema ***
    ***               Version 06, May 2006                           ***
    *** draft-ietf-inch-iodef-06 ***
    ********************************************************************
    ********************************************************************
    -->
    <!--
    =====================================================================
    == IODEF-Document class                                           ==
    =====================================================================
    -->
    <xs:annotation>
        <xs:documentation>Root Element IODEF-Document</xs:documentation>
    </xs:annotation>
    <xs:element name="IODEF-Document">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="iodef:Incident"/>
                <xs:element ref="ds:Signature" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="version" type="xs:string" fixed="1.00"/>
        </xs:complexType>
    </xs:element>
</xs:schema>
```
<xs:element name="Incident">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:IncidentID" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:AlternativeID" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:RelatedActivity" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:DetectTime" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:StartTime" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:EndTime" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:ReportTime" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Description" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Assessment" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Method" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Contact" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:EventData" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:History" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:AdditionalData" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="purpose" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="traceback"/>
          <xs:enumeration value="mitigation"/>
          <xs:enumeration value="reporting"/>
          <xs:enumeration value="other"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="lang" type="xs:language"/>
    <xs:attribute name="restriction" type="iodef:restriction-type" default="private"/>
  </xs:complexType>
</xs:element>
<!--
====================================================================
===  Incident class                                              ===
====================================================================
-->

== IncidentID class ==

-->
<xs:element name="IncidentID" type="iodef:IncidentIDType"/>
<xs:complexType name="IncidentIDType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="name" type="xs:string" use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

-->

== AlternativeID class ==

-->
<xs:element name="AlternativeID">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:IncidentID" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="restriction" type="iodef:restriction-type"/>
  </xs:complexType>
</xs:element>

-->

== RelatedActivity class ==

-->
<xs:element name="RelatedActivity">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:IncidentID" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="restriction" type="iodef:restriction-type"/>
  </xs:complexType>
</xs:element>

-->

== AdditionalData class ==
<xs:element name="Contact">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:ContactName" minOccurs="0"/>
      <xs:element ref="iodef:Description" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:RegistryHandle" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:PostalAddress" minOccurs="0"/>
      <xs:element ref="iodef:Email" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Telephone" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Fax" minOccurs="0"/>
      <xs:element ref="iodef:Timezone" minOccurs="0"/>
      <xs:element ref="iodef:Contact" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="role">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="creator"/>
          <xs:enumeration value="admin"/>
          <xs:enumeration value="tech"/>
          <xs:enumeration value="irt"/>
          <xs:enumeration value="cc"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="type">
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="creator"/>
        <xs:enumeration value="admin"/>
        <xs:enumeration value="tech"/>
        <xs:enumeration value="irt"/>
        <xs:enumeration value="cc"/>
      </xs:restriction>
    </xs:attribute>
  </xs:complexType>
</xs:element>
<xs:simpleType>
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="person"/>
    <xs:enumeration value="organization"/>
  </xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>
<xs:element name="ContactName" type="iodef:MLStringType"/>
<xs:element name="RegistryHandle">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="registry">
          <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="internic"/>
              <xs:enumeration value="apnic"/>
              <xs:enumeration value="arin"/>
              <xs:enumeration value="lacnic"/>
              <xs:enumeration value="ripe"/>
              <xs:enumeration value="afrinic"/>
              <xs:enumeration value="local"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="PostalAddress" type="iodef:MLStringType"/>
<xs:element name="Email" type="iodef:MLStringType"/>
<xs:element name="Telephone" type="xs:string"/>
<xs:element name="Fax" type="xs:string"/>
</xs:attribute>

---

Time-based classes

---
<xs:element name="DateTime" type="xs:dateTime"/>
<xs:element name="ReportTime" type="xs:dateTime"/>
<xs:element name="DetectTime" type="xs:dateTime"/>
<xs:element name="StartTime" type="xs:dateTime"/>
<xs:element name="EndTime" type="xs:dateTime"/>
<xs:element name="Timezone" type="iodef:TimezoneType"/>
<xs:simpleType name="TimezoneType">
  <xs:restriction base="xs:string">
    <xs:pattern value="Z|[\+\-](0[0-9]|1[0-4]):[0-5][0-9]"/>
  </xs:restriction>
</xs:simpleType>

-->

Internet-Draft              IODEF Data Model                   June 2006

-->

====================================================================
===  History class                                               ===
===    - HistoryItem                                             ===
====================================================================
<xs:element name="History">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:HistoryItem" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="restriction" type="iodef:restriction-type" default="default"/>
  </xs:complexType>
</xs:element>

<xs:element name="HistoryItem">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:DateTime"/>
      <xs:element ref="iodef:IncidentID" minOccurs="0"/>
      <xs:element ref="iodef:Description" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="restriction" type="iodef:restriction-type" default="default"/> 
    <xs:attribute name="action" type="iodef:action-type" default="other"/>
  </xs:complexType>
</xs:element>

-->

====================================================================
===  Expectation class                                           ===
====================================================================
<xs:element name="Expectation">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:Description" maxOccurs="unbounded"/>
      <xs:element ref="iodef:StartTime" minOccurs="0"/>
      <xs:element ref="iodef:EndTime" minOccurs="0"/>
      <xs:element ref="iodef:Contact" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="restriction" type="iodef:restriction-type" default="default"/>
    <xs:attribute name="severity" type="iodef:severity-type"/>
    <xs:attribute name="action" type="iodef:action-type" default="other"/>
  </xs:complexType>
</xs:element>

<xs:element name="Method">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:Classification" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Description" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="restriction" type="iodef:restriction-type"/>
  </xs:complexType>
</xs:element>

<xs:element name="Classification">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:name"/>
      <xs:element ref="iodef:url" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="origin" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="bugtraqid"/>
          <xs:enumeration value="cve"/>
          <xs:enumeration value="certcc"/>
          <xs:enumeration value="vendor"/>
          <xs:enumeration value="local"/>
          <xs:enumeration value="other"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>
<xs:element name="Assessment">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:Impact" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:TimeImpact" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:MonetaryImpact" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Confidence" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="restriction" type="iodef:restriction-type"/>
  </xs:complexType>
</xs:element>

<xs:element name="Impact">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="iodef:MLStringType">
        <xs:attribute name="severity" type="iodef:severity-type"/>
        <xs:attribute name="completion">
          <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="failed"/>
              <xs:enumeration value="succeeded"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="type" use="optional" default="unknown">
          <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="admin"/>
              <xs:enumeration value="dos"/>
              <xs:enumeration value="file"/>
              <xs:enumeration value="recon"/>
              <xs:enumeration value="user"/>
              <xs:enumeration value="unknown"/>
              <xs:enumeration value="other"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="TimeImpact">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:float">
        <xs:attribute name="severity" type="iodef:severity-type"/>
        <xs:attribute name="metric" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="labor"/>
              <xs:enumeration value="elapsed"/>
              <xs:enumeration value="downtime"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="duration" type="iodef:duration-type"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

<xs:element name="MonetaryImpact">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:float">
        <xs:attribute name="severity" type="iodef:severity-type"/>
        <xs:attribute name="currency" type="xs:string"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

<xs:element name="Confidence">
  <xs:complexType mixed="true">
    <xs:attribute name="rating" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="low"/>
          <xs:enumeration value="medium"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>
<xs:element name="EventData">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:Description" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:DetectTime" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:StartTime" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:EndTime" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Contact" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Assessment" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Method" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Flow" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Expectation" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Record" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:EventData" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:AdditionalData" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="restriction" type="iodef:restriction-type" default="default"/>
  </xs:complexType>
</xs:element>

---

== Flow class ==
---

<xs:element name="Flow">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:System" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

!--
<xs:element name="System">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:Node"/>
      <xs:element ref="iodef:Service" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:OperatingSystem" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Counter" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!--
====================================================================
===  System class                                                ===
====================================================================
-->
<xs:element name="Node">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:name" minOccurs="0"/>
      <xs:element ref="iodef:Address" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Location" minOccurs="0"/>
      <xs:element ref="iodef:DateTime" minOccurs="0"/>
      <xs:element ref="iodef:NodeRole" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Counter" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="Address">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="category" default="ipv4-addr"/>
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="asn"/>
            <xs:enumeration value="atm"/>
            <xs:enumeration value="e-mail"/>
            <xs:enumeration value="mac"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

<xs:element name="Location" type="iodef:MLStringType"/>
<xs:element name="NodeRole">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="iodef:MLStringType">
        <xs:attribute name="category" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="client"/>
              <xs:enumeration value="server-internal"/>
              <xs:enumeration value="server-public"/>
              <xs:enumeration value="www"/>
              <xs:enumeration value="mail"/>
              <xs:enumeration value="messaging"/>
              <xs:enumeration value="streaming"/>
              <xs:enumeration value="voice"/>
              <xs:enumeration value="file"/>
              <xs:enumeration value="ftp"/>
              <xs:enumeration value="p2p"/>
              <xs:enumeration value="name"/>
              <xs:enumeration value="directory"/>
              <xs:enumeration value="credential"/>
              <xs:enumeration value="print"/>
              <xs:enumeration value="application"/>
              <xs:enumeration value="database"/>
              <xs:enumeration value="infra"/>
              <xs:enumeration value="log"/>
              <xs:enumeration value="other"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
      <xs:extension base="iodef:MLStringType"></xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

Internet-Draft              IODEF Data Model                   June 2006

</xs:extension>
</xs:complexType>
</xs:element>
</xs:complexType>
</xs:element>
<!--
====================================================================
===  Service Class                                               ===
====================================================================
-->
<xs:element name="Service">
  <xs:complexType>
<xs:sequence>
  <xs:choice>
    <xs:element ref="iodef:port"/>
    <xs:element ref="iodef:portlist"/>
  </xs:choice>
  <xs:element name="ProtoType" type="xs:integer" minOccurs="0"/>
  <xs:element name="ProtoCode" type="xs:integer" minOccurs="0"/>
  <xs:element name="ProtoField" type="xs:integer" minOccurs="0"/>
  <xs:element ref="iodef:Application" minOccurs="0"/>
</xs:sequence>
<xs:attribute name="ip_version" type="xs:integer" default="4"/>
<xs:attribute name="ip_protocol" type="xs:integer"/>
</xs:complexType>

<xs:element name="port" type="xs:integer"/>
<xs:element name="portlist" type="xs:string"/>
<!--
====================================================================
===  Application and OperatingSystem class                       ===
====================================================================
-->
<xs:complexType>
  <xs:simpleContent>
    <xs:extension base="xs:double">
      <xs:attribute name="type" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="byte"/>
            <xs:enumeration value="packet"/>
            <xs:enumeration value="flow"/>
            <xs:enumeration value="session"/>
            <xs:enumeration value="event"/>
            <xs:enumeration value="alert"/>
            <xs:enumeration value="message"/>
            <xs:enumeration value="other"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="meaning" type="xs:string" use="optional"/>
      <xs:attribute name="duration" type="iodef:duration-type"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:element name="Record">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:RecordData" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="restriction" type="iodef:restriction-type"/>
  </xs:complexType>
</xs:element>

<xs:element name="RecordData">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:DateTime" minOccurs="0"/>
      <xs:element ref="iodef:Description" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Application" minOccurs="0"/>
      <xs:element ref="iodef:RecordPattern" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:RecordItem" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:sequence>
  <xs:attribute name="restriction" type="iodef:restriction-type"/>
</xs:complexType>
</xs:element>

<xs:element name="RecordPattern">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="type" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="regex"/>
              <xs:enumeration value="binary"/>
              <xs:enumeration value="xpath"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="offset" type="xs:integer" use="optional"/>
        <xs:attribute name="offsetunit" use="optional" default="line">
          <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="line"/>
              <xs:enumeration value="byte"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="instance" type="xs:integer" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

<xs:element name="RecordItem" type="iodef:ExtensionType"/>

<!--
====================================================================
=== Miscellaneous simple classes                                 ===
====================================================================
-->

<xs:element name="Description" type="iodef:MLStringType"/>
<xs:element name="name" type="xs:string"/>
<xs:element name="url" type="xs:string"/>

<!--
====================================================================
=== Complex Data Types                                           ===
====================================================================
-->
<xs:complexType name="MLStringType">
  <xs:simpleContent>
<xs:extension base="xs:string">


Internet-Draft IODEF Data Model June 2006

<xs:attribute name="lang" type="xs:string"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="ExtensionType" mixed="true">
<xs:sequence>
<xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="dtype" type="iodef:dtype-type" use="required"/>
<xs:attribute name="meaning" type="xs:string"/>
<xs:attribute name="formatid" type="xs:string"/>
<xs:attribute name="restriction" type="iodef:restriction-type"/>
</xs:complexType>

<!--
====================================================================
=== Global attribute type declarations.                             ===
====================================================================
-->
<!--
=== Global attribute type declarations.~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-->
<!--
| @restriction: defines restrictions on access to an element's content
-->
<xs:simpleType name="restriction-type">
<xs:restriction base="xs:NMTOKEN">
<xs:enumeration value="default"/>
<xs:enumeration value="public"/>
<xs:enumeration value="need-to-know"/>
<xs:enumeration value="private"/>
</xs:restriction>
</xs:simpleType>
<!--
| @severity: conveys the severity or priority of something
-->
<xs:simpleType name="severity-type">
<xs:restriction base="xs:NMTOKEN">
<xs:enumeration value="low"/>
<xs:enumeration value="medium"/>
<xs:enumeration value="high"/>
</xs:restriction>
</xs:simpleType>
<!--
@duration: units of time

```xml
<xs:simpleType name="duration-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="second"/>
    <xs:enumeration value="minute"/>
    <xs:enumeration value="hour"/>
    <xs:enumeration value="day"/>
    <xs:enumeration value="month"/>
    <xs:enumeration value="quarter"/>
    <xs:enumeration value="year"/>
  </xs:restriction>
</xs:simpleType>
```

<table>
<thead>
<tr>
<th>@action: actions taken</th>
</tr>
</thead>
</table>

```xml
<xs:simpleType name="action-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="nothing"/>
    <xs:enumeration value="contact-source-site"/>
    <xs:enumeration value="contact-target-site"/>
    <xs:enumeration value="contact-sender"/>
    <xs:enumeration value="investigate"/>
    <xs:enumeration value="block-host"/>
    <xs:enumeration value="block-network"/>
    <xs:enumeration value="block-port"/>
    <xs:enumeration value="rate-limit-host"/>
    <xs:enumeration value="rate-limit-network"/>
    <xs:enumeration value="rate-limit-port"/>
    <xs:enumeration value="remediate-other"/>
    <xs:enumeration value="status-triage"/>
    <xs:enumeration value="status-new-info"/>
    <xs:enumeration value="other"/>
  </xs:restriction>
</xs:simpleType>
```

<table>
<thead>
<tr>
<th>@dtype: data types for extensions</th>
</tr>
</thead>
</table>

```xml
<xs:simpleType name="dtype-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="boolean"/>
  </xs:restriction>
</xs:simpleType>
```
<xs:enumeration value="byte"/>
<xs:enumeration value="character"/>
<xs:enumeration value="date-time"/>
<xs:enumeration value="integer"/>
<xs:enumeration value="ntpstamp"/>
<xs:enumeration value="portlist"/>
<xs:enumeration value="real"/>
<xs:enumeration value="string"/>
<xs:enumeration value="file"/>
<xs:enumeration value="path"/>
<xs:enumeration value="frame"/>
<xs:enumeration value="packet"/>
<xs:enumeration value="ipv4-packet"/>
<xs:enumeration value="ipv6-packet"/>
<xs:enumeration value="url"/>

<xs:enumeration value="csv"/>
<xs:enumeration value="winreg"/>
<xs:enumeration value="xml"/>
<xs:enumeration value="other"/>
</xs:restriction>
</xs:simpleType>
</xs:schema>
9. Security considerations

This draft describes a data model for incident information. 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 the IODEF document itself is RECOMMENDED.

When used, the applied protective measures MUST use cryptographic techniques. XML Digital Signatures [13] MUST be used for ensuring integrity and non-repudiation, while XML Encryption [14] MUST be used to ensure the confidentiality of an IODEF document. Examples using signatures and encryption on an IODEF document can be found in Section 7:

- IODEF-Document with XML signature (Section 7.2)
- IODEF-Document encrypted using XML encryption (Section 7.3)
Additional information on applying XML Digital Signatures and XML Encryption to an IODEF document can be found in the IODEF Implementation Guide [20].

10. IANA considerations

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [16]

Registration request for the iodef namespace:

- URI: urn:ietf:params:xml:ns:iodef-1.0
- Registrant Contact: See the "Author's Address" section 10.2 of this document.
XML: None. Namespace URIs do not represent an XML specification.

Registration request for the iodef XML schema:

- URI: urn:ietf:params:xml:schema:iodef-1.0
- Registrant Contact: See the "Author's Address" section 10.2 of this document.
- XML: See the "IODEF Document Schema" of section 8 of this document.

FIXME: Many of the enumerated types in this document are subject to future extension, and should be defined by an IANA registry. The IANA Considerations section must explicitly list each of these enumerated types. Extensions to this registry may only be made by IETF Informational RFCs. We will request that IANA maintain this registry as an XML Schema defining a set of xs:simpleType elements for named types to be referenced by a future revision of the IODEF schema, further requiring that each value in each simpleType be annotated as to the RFC creating it.

11. Acknowledgments

The following groups contributed substantially to this document and should be recognized for their efforts.
o the Incident Object Description and Exchange Format Working-Group of the TERENA task-force (TF-CSIRT)

o the eCSIRT.net project
12. References

12.1. Normative References


March 2002.


12.2. Informative References


Authors' Addresses

Roman Danyliw
CERT Coordination Center
Pittsburgh
USA

Email: rdd@cert.org

Jan Meijer
SURFnet bv
Utrecht
Netherlands

Email: jan.meijer@surfnet.nl

Yuri Demchenko
University of Amsterdam
Netherlands

Email: demch@chello.nl
Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights 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; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat 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 implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an
"AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2006). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.