The Incident Object Description Exchange Format v2
draft-ietf-mile-rfc5070-bis-14

Abstract

   The Incident Object Description Exchange Format (IODEF) defines a
   data representation for sharing information commonly exchanged by
   Computer Security Incident Response Teams (CSIRTs) about computer
   security incidents. This document describes the information model
   for the IODEF and provides an associated data model specified with
   XML Schema.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   Organizations require help from other parties to mitigate malicious
   activity targeting their network and to gain insight into potential
   threats. This coordination might entail working with an ISP to
   filter attack traffic, contacting a remote site to take down a bot-
   network, or sharing watch-lists of known malicious IP addresses in a
   consortium.

   The Incident Object Description Exchange Format (IODEF) is a format
   for representing computer security information commonly exchanged
   between Computer Security Incident Response Teams (CSIRTs). It
   provides an XML representation for conveying:

   o cyber intelligence to characterize threats;

   o cyber incident reports to document particular cyber security
      events or relationships between events;

   o cyber event mitigation to request proactive and reactive
      mitigation approaches to cyber intelligence or incidents; and

   o cyber information sharing meta-data so that these various classes
      of information can be exchanged among parties.

   The data model encodes information about hosts, networks, and the
   services running on these systems; attack methodology and associated
   forensic evidence; impact of the activity; and limited approaches for
   documenting workflow.

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

   o increased automation in processing of incident data, since the
      resources of security analysts to parse free-form textual
      documents will be reduced;

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

o a common format on which to build interoperable tools for incident
   handling and subsequent analysis, specifically when data comes
   from multiple constituencies.

Coordinating with other CSIRTs is not strictly a technical problem.
There are numerous procedural, trust, and legal considerations that
might prevent an organization from sharing information. The IODEF
does not attempt to address them. However, operational
implementations of the IODEF will need to consider this broader
context.

Sections 3 and 8 specify the IODEF data model with text and an XML
schema. The types used by the data model are covered in Section 2.
Processing considerations, the handling of extensions, and
internationalization issues related to the data model are covered in
Sections 4, 5, and 6, respectively. Examples are listed in
Section 7. Section 1 provides the background for the IODEF, and
Section 9 documents the security considerations.

1.1. Changes from 5070

This document contains changes with respect to its predecessor
RFC5070.

o All of the RFC5070 Errata was implemented.

o Imported the xmlns:ds namespace to include digital signature hash
   classes.

o The following classes were added to IODEF-Document:
   AdditionalData.

o The following class and attribute was added to Incident:
   IndicatorData and @status.

o The following classes were added to Incident and EventData:
   Discovery.

o The following classes and attributes were added to the Service
   class: EmailData, DomainData, AssetID, ApplicationHeader @virtual,
   and @ownership. Service@ip_protocol was renamed to @ip-protocol.

o The following classes were added to the Record class: HashData and
   WindowsRegistryKeysModified.

o The following classes were added to the RelatedActivity class:
   ThreatActor, Campaign, Confidence, Description, and
   AdditionalData.

o The following classes were added to Assessment: IncidentCategory,
   SystemImpact, BusinessImpact, IntendedImpact and MitigatingFactor.

o The following classes were added to Node: PostalAddress and
   DomainData. The following classes were removed from Node: Removed
   NodeName and DateTime.

o The following classes were added to the Contact class:
   ContactTitle.

o The following classes were added to Expectation and HistoryItem:
   DefinedCOA.

o The following classes were added to Service: ServiceName

o The following classes were added to Reference: ReferenceName
   (replaced Name).

o The following attributes were added to Counter: type and unit.

o Additional enumerated values were added to the following
   attributes: @restriction, {Expectation, HistoryItem}@action,
   NodeRole@category, Incident@purpose, Contact@role,
   AdditionalData@dtype, System@spoofed.

o Added option for public extension of enumerated attributes with an
   IANA registry and added @ext-restriction.

o Removed Impact class in favor of using SystemImpact and
   IncidentCategory.

o iodef:MLStringType uses xml:lang and @translation-id.

1.2. 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].

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

1.3. Notations

The normative IODEF data model is specified with the text in
Section 3 and the XML schema in Section 8. To help in the
understanding of the data elements, Section 3 also depicts the

underlying information model using Unified Modeling Language (UML).
This abstract presentation of the IODEF is not normative.

For clarity in this document, the term "XML document" will be used
when referring generically to any instance of an XML document. The
term "IODEF document" will be used to refer to specific elements and
attributes of the IODEF schema. The terms "class" and "element" will
be used interchangeably to reference either the corresponding data
element in the information or data models, respectively.

1.4. 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.

o 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.

o As 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 in
   the IODEF that is flexible enough to encompass most operators.

o Describing an incident for all definitions would require an
   extremely complex data model. Therefore, the IODEF only intends
   to be a framework to convey commonly exchanged incident
   information. 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.

o The domain of security analysis is not fully standardized and must
   rely on free-form textual descriptions. The IODEF attempts to
   strike a balance between supporting this free-form content, while
   still allowing automated processing of incident information.

o The IODEF is only one of several security relevant data
   representations being standardized. Attempts were made to ensure
   they were complementary. The data model of the Intrusion
   Detection Message Exchange Format [RFC4765] influenced the design
   of the IODEF.

Further discussion of the desirable properties for the IODEF can be
found in the Requirements for the Format for Incident Information
Exchange (FINE) [refs.requirements].

1.5. About the IODEF Implementation

   The IODEF implementation is specified as an Extensible Markup
   Language (XML) [W3C.XML] Schema [W3C.SCHEMA].

   Implementing the IODEF in XML provides numerous advantages. Its
   extensibility makes it ideal for specifying a data encoding framework
   that supports various character encodings. Likewise, the abundance
   of related technologies (e.g., XSL, XPath, XML-Signature) makes for
   simplified manipulation. However, XML is fundamentally a text
   representation, which makes it inherently inefficient when binary
   data must be embedded or large volumes of data must be exchanged.

2. IODEF Data Types

   The various data elements of the IODEF data model are typed. This
   section discusses these data types. When possible, native Schema
   data types were adopted, but for more complicated formats, regular
   expressions (see Appendix F of [W3C.SCHEMA.DTYPES]) or external
   standards were used.

2.1. Integers

   An integer is represented by the INTEGER data type. Integer data
   MUST be encoded in Base 10.

   The INTEGER data type is implemented as an "xs:integer" in
   [W3C.SCHEMA.DTYPES].

2.2. Real Numbers

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

   The REAL data type is implemented as an "xs:float" in
   [W3C.SCHEMA.DTYPES].

2.3. Characters and Strings

   A single character is represented by the CHARACTER data type. A
   character string is represented by the STRING data type. Special
   characters must be encoded using entity references. See Section 4.1.

   The CHARACTER and STRING data types are implement as an "xs:string"
   in [W3C.SCHEMA.DTYPES].

2.4. Multilingual Strings

A character string that needs to be represented in a language
different than the default encoding of the document is of the
ML_STRING data type.

ML_STRING data type is implemented as the "iodef:MLStringType" type
in the schema. This type extends the "xs:string" to include two
attributes. The body of any class that uses this type is the
multilingual string.

```
+------------------------+
| iodef:MLStringType     |
+------------------------+
| ENUM xml:lang          |
| STRING translation-id  |
|                        |
+------------------------+
```

Figure 1: The iodef:MLStringType Type

Classes of the iodef:MLStringType type have two attributes:

xml:lang
   Optional. ENUM. A language identifier per Section 2.12 of
   [W3C.XML] whose values and form are described in [RFC5646]. The
   interpretation of this code is described in Section 6.

translation-id
   Optional. STRING. An identifier to relate other instances of
   this class with the same parent as translations of this text. The
   scope of this identifier is limited to all of the direct, peer
   child classes of a given parent class.

Using this class enables representing translation of the same text in
multiple language. Each translation is a distinct instance of this
class with a common parent. This relationship between multiple
classes being translated instances of the same text is indicated by a
common identifier set in the translation-id attribute. The language
of a given class of this type is set by the xml:lang attribute.

2.5. Bytes

A binary octet is represented by the BYTE data type. A sequence of
binary octets is represented by the BYTE[] data type. These octets
are encoded using base64.

The BYTE data type is implemented as an "xs:base64Binary" in
[W3C.SCHEMA.DTYPES].

2.6. Hexadecimal Bytes

A binary octet is represented by the HEXBIN (and HEXBIN[]) data type.
This octet is encoded as a character tuple consisting of two
hexadecimal digits.

The HEXBIN data type is implemented as an "xs:hexBinary" in
[W3C.SCHEMA.DTYPES].

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 the IODEF schema, the enumerated type
keywords are used as attribute values.

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

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.

Date-time strings are formatted according to a subset of [ISO8601]
documented in [RFC3339].

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

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|[\+\-](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 [W3C.SCHEMA.DTYPES]. This regular
expression is identical to the timezone representation implemented in
an "xs:dateTime".

2.10. Port Lists

   A list of network ports are represented by the PORTLIST data type. A
   PORTLIST consists of a comma-separated list of numbers and ranges
   (N-M means ports N through M, inclusive). It is formatted according
   to the following regular expression: "\d+(\-\d+)?(,\d+(\-\d+)?)*".
   For example, "2,5-15,30,32,40-50,55-60".

   The PORTLIST data type is implemented as an "xs:string" with a
   regular expression constraint in the schema.

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 Section 2.23 of
   [RFC4519]. It defines a postal address as a free-form multi-line
   string separated by the "$" character.

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

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 2.3 of [RFC4519].

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

2.13. Telephone and Fax Numbers

   A telephone or fax number is represented by the PHONE data type. The
   format of the PHONE data type is documented in Section 2.35 of
   [RFC4519].

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

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 [RFC5322].

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

2.15. Uniform Resource Locator strings

   A uniform resource locator (URL) is represented by the URL data type.
   The format of the URL data type is documented in [RFC3986].

   The URL data type is implemented as an "xs:anyURI" in the schema.

2.16. Identifiers and Identifier References

   An identifier unique to the Document is represented by the ID data
   type. A reference to this identifier is represented by the IDREF
   data type. The acceptable format of ID and IDREF is documented in
   Section 3.3.8 and 3.3.9 of [W3C.SCHEMA.DTYPES].

   The ID and IDREF data types are implemented as "xs:ID" and "xs:IDREF"
   in the 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
   described and the relationship with other classes will be depicted
   with UML. When necessary, specific comments will be made about
   corresponding definition in the schema in Section 8

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 xml:lang            |<>--{0..*}--[ AdditionalData ]
| STRING format-id         |
| STRING private-enum-name |
| STRING private-enum-id   |
+--------------------------+
```

                    Figure 2: IODEF-Document Class

   The aggregate class that constitute IODEF-Document is:

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

   AdditionalData
      Zero or more. Mechanism by which to extend the data model. See
      Section 3.9

   The IODEF-Document class has three attributes:

version
   Required. STRING. The IODEF specification version number to
   which this IODEF document conforms. The value of this attribute
   MUST be "2.00"

xml:lang
   Optional. ENUM. A language identifier per Section 2.12 of
   [W3C.XML] whose values and form are described in [RFC5646]. The
   interpretation of this code is described in Section 6.

format-id
   Optional. STRING. A free-form string to convey processing
   instructions to the recipient of the document. Its semantics must
   be negotiated out-of-band.

private-enum-name
   Optional. STRING. A globally unique identifier for the CSIRT
   generating the document to deconflict private extensions used in
   the Document. The fully qualified domain name associated with the
   CSIRT MUST be used as the identifier.

private-enum-id
   Optional. STRING. An organizationally unique identifier for an
   extension used in the Document. If this attribute is set, the
   private-enum-name MUST also be set.

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.

```
   +-------------------------+
   | Incident                |
   +-------------------------+
   | ENUM purpose            |<>----------[ IncidentID     ]
   | STRING ext-purpose      |<>--{0..1}--[ AlternativeID   ]
   | ENUM status             |<>--{0..*}--[ RelatedActivity ]
   | STRING ext-status       |<>--{0..1}--[ DetectTime      ]
   | ENUM xml:lang           |<>--{0..1}--[ StartTime       ]
   | ENUM restriction        |<>--{0..1}--[ EndTime         ]
   | STRING ext-restriction  |<>--{0..1}--{ RecoveryTime    ]
   | STRING observable-id    |<>----------[ ReportTime      ]
   |                         |<>--{0..1}--[ GenerationTime  ]
   |                         |<>--{0..*}--[ Description      ]
   |                         |<>--{0..*}  [ Discovery       ]
   |                         |<>--{1..*}--[ Assessment      ]
   |                         |<>--{0..*}--[ Method          ]
   |                         |<>--{1..*}--[ Contact         ]
   |                         |<>--{0..*}--[ EventData       ]
   |                         |<>--{0..*}--[ IndicatorData   ]
   |                         |<>--{0..1}--[ History         ]
   |                         |<>--{0..*}--[ AdditionalData  ]
   +-------------------------+
```

                     Figure 3: 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. The incident tracking numbers used by other CSIRTs
      to refer to the incident described in the document.

   RelatedActivity
      Zero or more. Related activity and attribution of this activity.

   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.

   RecoveryTime

      Zero or one. The time the site recovered from the incident.

   ReportTime
      One. The time the incident was reported.

   GenerationTime
      Zero or one. The time the content in this Incident class was
      generated.

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

   Discovery
      Zero or more. The means by which this incident was detected.

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

   Method
      Zero or more. The techniques used by the intruder in the
      incident.

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

   EventData
      Zero or more. Description of the events comprising the incident.

   IndicatorData
      Zero or more. Description of indicators.

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

   AdditionalData
      Zero or more. Mechanism by which to extend the data model.

   The Incident class has eight 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 3.17). These values are maintained in
      the "Incident-purpose" IANA registry per Table 1. 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. watch. The document was sent to convey indicators to watch
    for particular activity.

5. other. The document was sent for purposes specified in the
    Expectation class.

6. ext-value. An escape value used to extend this attribute.
    See Section 5.1.1.

ext-purpose
   Optional. STRING. A means by which to extend the purpose
   attribute. See Section 5.1.1.

status
   Optional. ENUM. The status attribute conveys the state in a
   workflow where the incident is currently found. These values are
   maintained in the "Incident-status" IANA registry per Table 1.
   This attribute is defined as an enumerated list:

1. new. The document is newly reported and has not been
    actioned.

2. in-progress. The contents of this document are under
    investigation.

3. forwarded. The document has been forwarded to another party
    for handling.

4. resolved. The investigation into the activity in this
    document has concluded.

5. future. The described activity is suspected to occur in the
    future.

6. ext-value. An escape value used to extend this attribute.
    See Section 5.1.1.

ext-status
   Optional. STRING. A means by which to extend the status
   attribute. See Section 5.1.1.

xml:lang
    Optional. ENUM. A language identifier per Section 2.12 of
    [W3C.XML] whose values and form are described in [RFC5646]. The
    interpretation of this code is described in Section 6.

restriction
    Optional. ENUM. See Section 3.3.1.

ext-restriction
    Optional. STRING. A means by which to extend the restriction
    attribute. See Section 5.1.1.

observable-id
    Optional. ID. See Section 3.3.2.

3.3. Common Attributes

There are a number of recurring attributes used by the data model.
They are documented in this section.

3.3.1. restriction Attribute

The restriction attribute indicates the disclosure guidelines to
which the sender expects the recipient to adhere for the information
represented in this class and its children. This guideline provides
no security since there are no specified technical means to ensure
that the recipient of the document handles the information as the
sender requested.

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 (e.g., 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.

These values are maintained in the "Restriction" IANA registry per
Table 1.

1.   public. The information can be freely distributed without
      restriction.

2.   partner. The information may be shared within a closed
      community of peers, partners, or affected parties, but cannot be
      openly published.

3.   need-to-know. The information may be shared only within the
      organization with individuals that have a need to know.

4.   private. The information may not be shared.

5.   default. The information can be shared according to an
      information disclosure policy pre-arranged by the communicating
      parties.

6.   white. Same as 'public'.

7.   green. Same as 'partner'.

8.   amber. Same as 'need-to-know'.

9.   red. Same as 'private'.

10.   ext-value. An escape value used to extend this attribute. See
      Section 5.1.1.

3.3.2. observable-id Attribute

Information included in an incident report may be an observable
relevant to an indicator. The observable-id attribute provides a
unique identifier in the scope of the document for this observable.
This identifier can then used to reference the observable with an
ObservableReference class to define an indicator in the IndicatorData
class.

3.4. IncidentID Class

The IncidentID class represents an incident tracking number that is
unique in the context of the CSIRT and identifies the activity
characterized in an IODEF Document. This identifier would serve as
an index into the CSIRT incident handling system. The combination of
the name attribute and the string in the element content MUST be a
globally unique identifier describing the activity. Documents

generated by a given CSIRT MUST NOT reuse the same value unless they
are referencing the same incident.

```
+------------------------+
| IncidentID             |
+------------------------+
| STRING                 |
|                        |
| STRING name            |
| STRING instance        |
| ENUM restriction       |
| STRING ext-restriction |
+------------------------+
```

Figure 4: The IncidentID Class

The IncidentID class has four attributes:

name
   Required. STRING. An identifier describing the CSIRT that
   created the document. In order to have a globally unique CSIRT
   name, the fully qualified domain name associated with the CSIRT
   MUST be used.

instance
   Optional. STRING. An identifier referencing a subset of the
   named incident.

restriction
   Optional. ENUM. See Section 3.3.1. The default value is
   "public".

ext-restriction
   Optional. STRING. A means by which to extend the restriction
   attribute. See Section 5.1.1.

3.5. AlternativeID Class

The AlternativeID class lists the incident tracking numbers used by
CSIRTs, other than the one generating the document, to refer to the
identical activity described in the IODEF document. 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 must never be considered an
AlternativeID.

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

                    Figure 5: 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 two attributes:

   restriction
      Optional. ENUM. See [Section 3.3.1](#).

   ext-restriction
      Optional. STRING. A means by which to extend the restriction
      attribute. See [Section 5.1.1](#).

3.6. RelatedActivity Class

   The RelatedActivity class relates the information described in the
   rest of the IODEF document to previously observed incidents or
   activity; and allows attribution to a specific actor or campaign.

```
+-----------------------+
| RelatedActivity       |
+-----------------------+
| ENUM restriction      |<>--{0..*}--[ IncidentID     ]
| STRING ext-restriction |<>--{0..*}--[ URL            ]
|                       |<>--{0..*}--[ ThreatActor    ]
|                       |<>--{0..*}--[ Campaign       ]
|                       |<>--{0..1}--[ Confidence     ]
|                       |<>--{0..*}--[ Description    ]
|                       |<>--{0..*}--[ AdditionalData ]
+-----------------------+
```

                    Figure 6: RelatedActivity Class

   The aggregate classes that constitutes RelatedActivity are:

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

   URL
      One or more. URL. A URL to activity related to this incident.

   ThreatActor
      One or more. The threat actor to whom the described activity is
      attributed.

   Campaign
      One or more. The campaign of a given threat actor to whom the
      described activity is attributed.

   Confidence
      Zero or one. An estimate of the confidence in attributing this
      RelatedActivity to the event described in the document.

   Description
      Zero or more. ML_STRING. A description of how these
      relationships were derived.

   AdditionalData
      Zero or more. A mechanism by which to extend the data model.

   RelatedActivity MUST at least have one instance of IncidentID, URL,
   ThreatActor, or Campaign.

   The RelatedActivity class has two attributes:

   restriction
      Optional. ENUM. See Section 3.3.1.

   ext-restriction
      Optional. STRING. A means by which to extend the restriction
      attribute. See Section 5.1.1.

3.7. ThreatActor Class

   The ThreatActor class describes a given actor.

```
   +------------------------+
   | Actor                  |
   +------------------------+
   | ENUM restriction       |<>--{0..1}--[ ThreatActorID  ]
   | STRING ext-restriction |<>--{0..*}--[ Description     ]
   |                        |<>--{0..*}--[ AdditionalData ]
   +------------------------+
```

                        Figure 7: ThreatActor Class

The aggregate classes that constitutes ThreatActor are:

ThreatActorID
   One or more. STRING. An identifier for the ThreatActor.

Description
   One or more. ML_STRING. A description of the ThreatActor.

AdditionalData
   Zero or more. A mechanism by which to extend the data model.

ThreatActor MUST have at least one instance of a ThreatActorID or
Description.

The ThreatActor class has two attributes:

restriction
   Optional. ENUM. See Section 3.3.1.

ext-restriction
   Optional. STRING. A means by which to extend the restriction
   attribute. See Section 5.1.1.

3.8. Campaign Class

The Campaign class describes a campaign of attacks by a threat actor.

```
+------------------------+
| Campaign               |
+------------------------+
| ENUM restriction       |<>--{0..1}--[ CampaignID     ]
| STRING ext-restriction |<>--{0..*}--[ Description    ]
|                        |<>--{0..*}--[ AdditionalData ]
+------------------------+
```

                    Figure 8: Campaign Class

The aggregate classes that constitutes Campaign are:

CampaignID
   One or more. STRING. An identifier for the Campaign.

Description
   One or more. ML_STRING. A description of the Campaign.

AdditionalData
   Zero or more. A mechanism by which to extend the data model.

   Campaign MUST have at least one instance of a Campaign or
   Description.

   The Campaign class has two attributes:

   restriction
      Optional. ENUM. See Section 3.3.1.

   ext-restriction
      Optional. STRING. A means by which to extend the restriction
      attribute. See Section 5.1.1.

3.9. AdditionalData Class

   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. A detailed
   discussion for extending the data model and the schema can be found
   in Section 5.

   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.

```
   +------------------------+
   | AdditionalData         |
   +------------------------+
   | ANY                    |
   |                        |
   | ENUM dtype             |
   | STRING ext-dtype       |
   | STRING meaning         |
   | STRING formatid        |
   | ENUM restriction       |
   | STRING ext-restriction |
   +------------------------+
```

                    Figure 9: The AdditionalData Class

   The AdditionalData class has six 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". These values are maintained in the
     "AdditionalData-dtype" IANA registry per Table 1.

     1.   boolean. The element content is of type BOOLEAN.

     2.   byte. The element content is of type BYTE.

     3.   bytes. The element content is of type HEXBIN.

     4.   character. The element content is of type CHARACTER.

     5.   date-time. The element content is of type DATETIME.

     6.   ntpstamp. Same as date-time.

     7.   integer. The element content is of type INTEGER.

     8.   portlist. The element content is of type PORTLIST.

     9.   real. The element content is of type REAL.

     10. string. The element content is of type STRING.

     11. file. The element content is a base64 encoded binary file
          encoded as a BYTE[] type.

     12. path. The element content is a file-system path encoded as a
          STRING type.

     13. frame. The element content is a layer-2 frame encoded as a
          HEXBIN type.

     14. packet. The element content is a layer-3 packet encoded as a
          HEXBIN type.

     15. ipv4-packet. The element content is an IPv4 packet encoded
          as a HEXBIN type.

     16. ipv6-packet. The element content is an IPv6 packet encoded
          as a HEXBIN type.

     17. url. The element content is of type URL.

     18. csv. The element content is a common separated value (CSV)
          list per Section 2 of [RFC4180] encoded as a STRING type.

19. winreg. The element content is a Windows registry key
     encoded as a STRING type.

20. xml. The element content is XML. See Section 5.

21. ext-value. An escape value used to extend this attribute.
     See Section 5.1.1.

ext-dtype
   Optional. STRING. A means by which to extend the dtype
   attribute. See Section 5.1.1.

meaning
   Optional. STRING. A free-form description of the element
   content.

formatid
   Optional. STRING. An identifier referencing the format and
   semantics of the element content.

restriction
   Optional. ENUM. See Section 3.3.1.

ext-restriction
   Optional. STRING. A means by which to extend the restriction
   attribute. See Section 5.1.1.

3.10. 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.

The inheriting definition of Contact provides a way to relate
information without requiring the explicit use of identifiers in the
classes or duplication of data. A complete point of contact is
derived by a particular traversal from the root Contact class to the
leaf Contact class. As such, multiple points of contact might be
specified in a single instance of a Contact class. Each child
Contact class logically inherits contact information from its
ancestors.

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

                     Figure 10: The Contact Class

   The aggregate classes that constitute the Contact class are:

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

   ContactTitle
      Zero or more. ML_STRING. The title for the individual named in
      the ContactName.

   Description
      Zero or more. 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 more. A handle name into the registry of the contact.

   PostalAddress
      Zero or one. The postal address of the contact.

   Email
      Zero or more. The email address of the contact.

   Telephone
      Zero or more. The telephone number of the contact.

   Fax
      Zero or one. The facsimile telephone number of the contact.

Timezone
   Zero or one. TIMEZONE. The timezone in which the contact resides
   formatted according to [Section 2.9](#).

Contact
   Zero or more. A Contact instance contained within another Contact
   instance inherits 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
   contacts at the same organization.

AdditionalData
   Zero or more. A mechanism by which to extend the data model.

At least one of the aggregate classes MUST be present in an instance
of the Contact class. This is not enforced in the IODEF schema as
there is no simple way to accomplish it.

The Contact class has six attributes:

role
   Required. ENUM. Indicates the role the contact fulfills. This
   attribute is defined as an enumerated list. These values are
   maintained in the "Contact-role" IANA registry per Table 1.

   1.   creator. The entity that generate the document.

   2.   reporter. The entity that reported the information.

   3.   admin. An administrative contact or business owner for an
        asset or organization.

   4.   tech. An entity responsible for the day-to-day management of
        technical issues for an asset or organization.

   5.   provider. An external hosting provider for an asset.

   6.   zone. An entity with authority over a DNS zone.

   7.   user. An end-user of an asset or part of an organization.

   8.   billing. An entity responsible for billing issues for an
        asset or organization.

   9.   legal. An entity responsible for legal issue related to an
        asset or organization.

10. irt. An entity responsible for handling security issues for
     an asset or organization.

11. abuse. An entity responsible for handling abuse originating
     from an asset or organization.

12. cc. An entity that is to be kept informed about the events
     related to an asset or organization.

13. cc-irt. A CSIRT or information sharing organization
     coordinating activity related to an asset or organization.

14. leo. A law enforcement organization supporting the
     investigation of activity affecting an asset or organization.

15. vendor. The vendor that produces an asset.

16. vendor-support. A vendor that provides services.

17. victim. A victim in the incident.

18. victim-notified. A victim in the incident who has been
     notified.

19. ext-value. An escape value used to extend this attribute.
     See Section 5.1.1.

ext-role
   Optional. STRING. A means by which to extend the role attribute.
   See Section 5.1.1.

type
   Required. ENUM. Indicates the type of contact being described.
   This attribute is defined as an enumerated list. These values are
   maintained in the "Contact-type" IANA registry per Table 1.

   1. person. The information for this contact references an
       individual.

   2. organization. The information for this contact references an
       organization.

   3. ext-value. An escape value used to extend this attribute.
       See Section 5.1.1.

ext-type
   Optional. STRING. A means by which to extend the type attribute.
   See Section 5.1.1.

   restriction
      Optional. ENUM. See Section 3.3.1.

   ext-restriction
      Optional. STRING. A means by which to extend the restriction
      attribute. See Section 5.1.1.

3.10.1. RegistryHandle Class

   The RegistryHandle class represents a handle into an Internet
   registry or community-specific database. The handle is specified in
   the element content and the type attribute specifies the database.

```
+--------------------+
| RegistryHandle     |
+--------------------+
| STRING             |
|                    |
| ENUM registry      |
| STRING ext-registry |
+--------------------+
```

                   Figure 11: The RegistryHandle Class

   The RegistryHandle class has two attributes:

   registry
      Required. ENUM. The database to which the handle belongs. These
      values are maintained in the "RegistryHandle-registry" IANA
      registry per Table 1. 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. 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

      8. ext-value. An escape value used to extend this attribute.
          See Section 5.1.1.

   ext-registry
      Optional. STRING. A means by which to extend the registry
      attribute. See Section 5.1.1.

3.10.2. PostalAddress Class

   The PostalAddress class specifies a postal address formatted
   according to the POSTAL data type (Section 2.11).

```
+--------------------+
| PostalAddress      |
+--------------------+
| POSTAL             |
|                    |
| STRING meaning     |
| ENUM xml:lang      |
+--------------------+
```

                   Figure 12: The PostalAddress Class

   The PostalAddress class has two attributes:

   meaning
      Optional. STRING. A free-form description of the element
      content.

   xml:lang
      Optional. ENUM. A language identifier per Section 2.12 of
      [W3C.XML] whose values and form are described in [RFC5646]. The
      interpretation of this code is described in Section 6.

3.10.3. Email Class

   The Email class specifies an email address formatted according to
   EMAIL data type (Section 2.14).

```
+--------------+
| Email        |
+--------------+
| EMAIL        |
|              |
| ENUM meaning |
+--------------+
```

                      Figure 13: The Email Class

   The Email class has one attribute:

meaning
      Optional. ENUM. A free-form description of the element content.

3.10.4. Telephone and Fax Classes

The Telephone and Fax classes specify a voice or fax telephone number
respectively, and are formatted according to PHONE data type
(Section 2.13).

```
+--------------------+
| {Telephone | Fax } |
+--------------------+
| PHONE              |
|                    |
| ENUM meaning       |
+--------------------+
```

Figure 14: The Telephone and Fax Classes

The Telephone class has one attribute:

meaning
      Optional. ENUM. A free-form description of the element content
      (e.g., hours of coverage for a given number).

3.11. Time Classes

The data model uses six 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 2.8).

```
+-----------------+
| StartTime       |
| EndTime         |
| ReportTime      |
| DetectTime      |
| GenerationTime  |
| DateTime        |
+-----------------+
| DATETIME        |
+-----------------+
```

Figure 15: The Time Classes

3.11.1. StartTime Class

   The StartTime class represents the time the incident began.

3.11.2. EndTime Class

   The EndTime class represents the time the incident ended.

3.11.3. DetectTime Class

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

3.11.4. ReportTime Class

   The ReportTime class represents the time the incident was reported.

3.11.5. GenerationTime Class

   The GenerationTime class represents the time when the IODEF document
   was produced. This timestamp MUST be the time at which the IODEF
   document was generated.

3.11.6. DateTime

   The DateTime class is a generic representation of a timestamp. Infer
   its semantics from the parent class in which it is aggregated.

3.12. Discovery Class

   The Discovery class describes how an incident was detected.

```
   +------------------------+
   | Discovery              |
   +------------------------+
   | ENUM source            |<>--{0..*}--[ Description      ]
   | STRING ext-source      |<>--{0..*}--[ Contact          ]
   | ENUM restriction       |<>--{0..*}--[ DetectionPattern ]
   | STRING ext-restriction |
   +------------------------+
```

                   Figure 16: The Discovery Class

   The Discovery class is composed of three aggregate classes.

   Description
      Zero or more. ML_STRING. A free-form text description of how
      this incident was detected.

Contact
   Zero or more. Contact information for the party that discovered
   the incident.

DetectionPattern
   Zero or more. Describes an application-specific configuration
   that detected the incident.

The Discovery class has four attribute:

source
   Optional. ENUM. Categorizes the techniques used to discover the
   incident. These values are partially derived from Table 3-1 of
   [NIST800.61rev2]. These values are maintained in the "Discovery-
   source" IANA registry per Table 1.

   1.   nidps. Network Intrusion Detection or Prevention system.

   2.   hips. Host-based Intrusion Prevention system.

   3.   siem. Security Information and Event Management System.

   4.   av. Antivirus or and antispam software.

   5.   third-party-monitoring. Contracted third-party monitoring
        service.

   6.   incident. The activity was discovered while investigating an
        unrelated incident.

   7.   os-log. Operating system logs.

   8.   application-log. Application logs.

   9.   device-log. Network device logs.

   10. network-flow. Network flow analysis.

   11. passive-dns. Passive DNS analysis.

   12. investigation. Manual investigation initiated based on
        notification of a new vulnerability or exploit.

   13. audit. Security audit.

   14. internal-notification. A party within the organization
        reported the activity

   15. external-notification. A party outside of the organization
        reported the activity.

   16. leo. A law enforcement organization notified the victim
        organization.

   17. partner. A customer or business partner reported the
        activity to the victim organization.

   18. actor. The threat actor directly or indirectly reported this
        activity to the victim organization.

   19. unknown. Unknown detection approach.

   20. ext-value. An escape value used to extend this attribute.
        See Section 5.1.1.

   ext-source
      Optional. STRING. A means by which to extend the source
      attribute. See Section 5.1.1.

   restriction
      Optional. ENUM. See Section 3.3.1.

   ext-restriction
      Optional. STRING. A means by which to extend the restriction
      attribute. See Section 5.1.1.

3.12.1. DetectionPattern Class

   The DetectionPattern class describes a configuration or signature
   that can be used by an IDS/IPS, SIEM, anti-virus, end-point
   protection, network analysis, malware analysis, or host forensics
   tool to identify a particular phenomenon. This class requires the
   identification of the target application and allows the configuration
   to be describes in either free-form or machine readable form.

```
   +------------------------+
   | DetectionPattern       |
   +------------------------+
   | ENUM restriction       |<>----------[ Application            ]
   | STRING ext-restriction |<>--{0..*}--[ Description            ]
   |                        |<>--{0..*}--[ DetectionConfiguration ]
   +------------------------+
```

                  Figure 17: The DetectionPattern Class

   The DetectionPattern class is composed of three aggregate classes.

   Application
      . One. The application for which the DetectionConfiguration or
      Description is being provided.

   Description
      Zero or more. ML_STRING. A free-form text description of how to
      use the Application or provided DetectionConfiguration.

   DetectionConfiguration
      Zero or more. STRING. A machine consumable configuration to find
      a pattern of activity.

   Either an instance of the Description or DetectionConfiguration class
   MUST be present.

   The DetectionPattern class has two attributes:

   restriction
      Optional. ENUM. See Section 3.3.1.

   ext-restriction
      Optional. STRING. A means by which to extend the restriction
      attribute. See Section 5.1.1.

3.13. Method Class

   The Method class describes the tactics, techniques, or procedures
   used by the intruder in the incident. This class consists of both a
   list of references describing the attack method and a free form
   description.

```
+-----------------------+
| Method                |
+-----------------------+
| ENUM restriction      |<>--{0..*}--[ Reference      ]
| STRING ext-restriction |<>--{0..*}--[ Description    ]
|                       |<>--{0..*}--[ AdditionalData ]
+-----------------------+
```

                      Figure 18: The Method Class

   The Method class is composed of three aggregate classes.

   enum:Reference
      Zero or more. A reference to a vulnerability, malware sample,
      advisory, or analysis of an attack technique.

   Description

      Zero or more. ML_STRING. A free-form text description of
      techniques, tactics, or procedures used by the intruder.

   AdditionalData
      Zero or more. A mechanism by which to extend the data model.

   Either an instance of the Reference or Description class MUST be
   present.

   The Method class has two attributes:

   restriction
      Optional. ENUM. See Section 3.3.1.

   ext-restriction
      Optional. STRING. A means by which to extend the restriction
      attribute. See Section 5.1.1.

3.13.1. Reference Class

   The Reference class is an external reference to relevant information
   such a vulnerability, IDS alert, malware sample, advisory, or attack
   technique. A reference consists of a name, a URL to this reference,
   and an optional description.

```
      +-------------------------+
      | Reference               |
      +-------------------------+
      | ID observable-id        |<>--{0..1}--[ enum:ReferenceName ]
      |                         |<>--{0..*}--[ URL                ]
      |                         |<>--{0..*}--[ Description         ]
      +-------------------------+
```

                     Figure 19: The Reference Class

   The aggregate classes that constitute Reference:

   ReferenceName
      Zero or one. Reference identifier per [RFC-ENUM].

   URL
      Zero or more. URL. A URL associated with the reference.

   Description
      Zero or more. ML_STRING. A free-form text description of this
      reference.

      At least one of these classes MUST be present.

The Reference class has one attribute.

observable-id
    Optional. ID. See [Section 3.3.2](#).

[3.14](#). Assessment Class

The Assessment class describes the repercussions of the incident to
the victim.

```
+-------------------------+
| Assessment              |
+-------------------------+
| ENUM occurrence         |<>--{0..*}--[ IncidentCategory ]
| ENUM restriction        |<>--{0..*}--[ SystemImpact     ]
| STRING ext-restriction  |<>--{0..*}--[ BusinessImpact   ]
| ID observable-id        |<>--{0..*}--[ TimeImpact       ]
|                         |<>--{0..*}--[ MonetaryImpact   ]
|                         |<>--{0..*}--[ IntendedImpact   ]
|                         |<>--{0..*}--[ Counter          ]
|                         |<>--{0..*}--[ MitigatingFactor ]
|                         |<>--{0..1}--[ Confidence       ]
|                         |<>--{0..*}--[ AdditionalData   ]
+-------------------------+
```

                    Figure 20: Assessment Class

The aggregate classes that constitute Assessment are:

IncidentCategory
    Zero or more. ML_STRING. A free-form text description
    categorizing the type of Incident.

SystemImpact
    Zero or more. Technical characterization of the impact of the
    activity on the victim's enterprise.

BusinessImpact
    Zero or more. Impact of the activity on the business functions of
    the victim organization.

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

MonetaryImpact
    Zero or more. Impact of the activity measured with respect to
    financial loss.

   IntendedImpact
      Zero or more. Intended impact to the victim by the attacker.
      Identically defined as [Section 3.14.2](#) but describes intent rather
      than the realized impact.

   Counter
      Zero or more. A counter with which to summarize the magnitude of
      the activity.

   MitigatingFactor
      Zero or one. ML_STRING. A description of a mitigating factor an
      impact.

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

   AdditionalData
      Zero or more. A mechanism by which to extend the data model.

   A least one instance of the possible three impact classes (i.e.,
   Impact, TimeImpact, or MonetaryImpact) MUST be present.

   The Assessment class has four attributes:

   occurrence
      Optional. ENUM. Specifies whether the assessment is describing
      actual or potential outcomes.

      1. actual. This assessment describes activity that has occurred.

      2. potential. This assessment describes potential activity that
          might occur.

   restriction
      Optional. ENUM. See [Section 3.3.1](#).

   ext-restriction
      Optional. STRING. A means by which to extend the restriction
      attribute. See [Section 5.1.1](#).

   observable-id
      Optional. ID. See [Section 3.3.2](#).

3.14.1. SystemImpact Class

   The SystemImpact class describes the technical impact of the incident
   to the systems on the network.

This class is based on [RFC4765].


```
+-----------------------+
| SystemImpact          |
+-----------------------+
| ML_STRING             |
|                       |
| ENUM xml:lang         |
| STRING translation-id |
| ENUM severity         |
| ENUM completion       |
| ENUM type             |
| STRING ext-type       |
+-----------------------+
```

Figure 21: SystemImpact Class

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

The SystemImpact class has six attributes:

xml:lang
   Optional. ENUM. A language identifier. See Section 6.

translation-id
   Optional. STRING. An identifier to relate other instances of
   this class as translations of this text. See Section 6.

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 impact. The permitted values are
      shown below. The default value is "unknown". These values are
      maintained in the "SystemImpact-type" IANA registry per Table 1.

      1.   takeover-account. Control was taken of a given account
           (e.g., a social media account).

      2.   takeover-service. Control was taken of a given service.

      3.   takeover-system. Control was taken of a given system.

      4.   cps-manipulation. A cyber physical system was manipulated.

      5.   cps-damage. A cyber physical system was damaged.

      6.   availability-data. Access to particular data was degraded or
           denied.

      7.   availability-account. Access to an account was degraded or
           denied.

      8.   availability-service. Access to a service was degraded or
           denied.

      9.   availability-system. Access to a system was degraded or
           denied.

      10. damaged-system. Hardware on a system was irreparably
           damaged.

      11. damaged-data. Data on a system was deleted.

      12. breach-proprietary. Sensitive or proprietary information was
           accessed or exfiltrated.

      13. breach-privacy. Personally identifiable information was
           accessed or exfiltrated.

      14. breach-credential. Credential information was accessed or
           exfiltrated.

      15. breach-configuration. System configuration or data inventory
           was access or exfiltrated.

      16. integrity-data. Data on the system was modified.

17. integrity-configuration. Application or system configuration was modified.

18. integrity-hardware. Firmware of a hardware component was modified.

19. traffic-redirection. Network traffic on the system was redirected

20. monitoring-traffic. Network traffic emerging from a host was monitored.

21. monitoring-host. System activity (e.g., running processes, keystrokes) were monitored.

22. policy. Activity violated the system owner's acceptable use policy.

23. unknown. The impact is unknown.

24. ext-value. An escape value used to extend this attribute. See Section 5.1.1.

ext-type
    Optional. STRING. A means by which to extend the type attribute. See Section 5.1.1.

3.14.2. BusinessImpact Class

The BusinessImpact class describes and characterizes the degree to which the function of the organization was impacted by the Incident.

The element body describes the impact to the organization as a free-form text string. The two attributes characterize the impact.

```
+------------------------+
| BusinessImpact         |
+------------------------+
| ML_STRING              |
|                        |
| ENUM xml:lang          |
| STRING translation-id  |
| ENUM severity          |
| STRING ext-severity    |
| ENUM type              |
| STRING ext-type        |
+------------------------+
```

Figure 22: BusinessImpact Class

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

The BusinessImpact class has four attributes:

xml:lang
   Optional. ENUM. A language identifier. See Section 6.

translation-id
   Optional. STRING. An identifier to relate other instances of this class as translations of this text. See Section 6.

severity
   Optional. ENUM. Characterizes the severity of the incident on business functions. The permitted values are shown below. They were derived from Table 3-2 of [NIST800.61rev2]. The default value is "unknown". These values are maintained in the "BusinessImpact-severity" IANA registry per Table 1.

   1. none. No effect to the organization's ability to provide all services to all users.

   2. low. Minimal effect as the organization can still provide all critical services to all users but has lost efficiency.

   3. medium. The organization has lost the ability to provide a critical service to a subset of system users.

   4. high. The organization is no longer able to provide some critical services to any users.

   5. unknown. The impact is not known.

6. ext-value. An escape value used to extend this attribute.
    See Section 5.1.1.

ext-severity
   Optional. STRING. A means by which to extend the severity
   attribute. See Section 5.1.1.

type
   Required. ENUM. Characterizes the effect this incident had on
   the business. The permitted values are shown below. There is no
   default value. These values are maintained in the
   "BusinessImpact-type" IANA registry per Table 1.

   1. breach-proprietary. Sensitive or proprietary information was
       accessed or exfiltrated.

   2. breach-privacy. Personally identifiable information was
       accessed or exfiltrated.

   3. breach-credential. Credential information was accessed or
       exfiltrated.

   4. loss-of-integrity. Sensitive or proprietary information was
       changed or deleted.

   5. loss-of-service. Service delivery was disrupted.

   6. theft-financial. Money was stolen.

   7. theft-service. Services were misappropriated.

   8. degraded-reputation. The reputation of the organization's
       brand was diminished.

   9. asset-damage. A cyber-physical system was damaged.

   10. asset-manipulation. A cyber-physical system was manipulated.

   11. legal. The incident resulted in legal or regulatory action.

   12. extortion. The incident resulted in actors extorting the
       victim organization.

   13. ext-value. An escape value used to extend this attribute.
       See Section 5.1.1.

ext-type

      Optional. STRING. A means by which to extend the type attribute.
      See Section 5.1.1.

3.14.3. 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.

```
+---------------------+
| TimeImpact          |
+---------------------+
| REAL                |
|                     |
| ENUM severity       |
| ENUM metric         |
| STRING ext-metrics  |
| ENUM duration       |
| STRING ext-duration |
+---------------------+
```

                     Figure 23: TimeImpact Class

   The element content is a positive, floating point (REAL) number
   specifying a unit of time. The duration and metric attributes will
   imply the semantics of the element content.

   The TimeImpact class has five 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. These values are maintained in the "TimeImpact-
      metric" IANA registry per Table 1.

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 (i.e., wall-clock time).

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

4. ext-value. An escape value used to extend this attribute.
    See Section 5.1.1.

ext-metric
    Optional. STRING. A means by which to extend the metric
    attribute. See Section 5.1.1.

duration
    Optional. 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 "hour". These values are
    maintained in the "TimeImpact-duration" IANA registry per Table 1.

1. second. The unit of the element content is seconds.

2. minute. The unit of the element content is minutes.

3. hour. The unit of the element content is hours.

4. day. The unit of the element content is days.

5. month. The unit of the element content is months.

6. quarter. The unit of the element content is quarters.

7. year. The unit of the element content is years.

8. ext-value. An escape value used to extend this attribute.
    See Section 5.1.1.

ext-duration
    Optional. STRING. A means by which to extend the duration
    attribute. See Section 5.1.1.

3.14.4. **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 24: MonetaryImpact Class

   The element content is a positive, floating point number (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
      Optional. STRING. Defines the currency in which the monetary
      impact is expressed. The permitted values are defined in "Codes
      for the representation of currencies and funds" of [ISO4217].
      There is no default value.

3.14.5. **Confidence Class**

   The Confidence class represents a best estimate of the validity and
   accuracy of the described impact (see Section 3.14) of the incident
   activity. This estimate can be expressed as a category or a numeric
   calculation.

This class if based upon [RFC4765].

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

Figure 25: 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 MUST 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.

   5. unknown. The confidence rating value is not known.

3.15. 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 ]
| STRING ext-restriction |
+------------------------+
```

                      Figure 26: 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 two attributes:

   restriction
      Optional. ENUM. See [Section 3.3.1](#). The default value is
      "default".

   ext-restriction
      Optional. STRING. A means by which to extend the restriction
      attribute. See [Section 5.1.1](#).

3.15.1. HistoryItem Class

   The HistoryItem class is an entry in the History ([Section 3.15](#)) 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.

```
+------------------------+
| HistoryItem            |
+------------------------+
| ENUM restriction       |<>----------[ DateTime       ]
| STRING ext-restriction |<>--{0..1}--[ IncidentId     ]
| ENUM action            |<>--{0..1}--[ Contact        ]
| STRING ext-action      |<>--{0..*}--[ Description    ]
| ID observable-id       |<>--{0..*}--[ DefinedCOA     ]
|                        |<>--{0..*}--[ AdditionalData ]
+------------------------+
```

                      Figure 27: HistoryItem Class

   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.

Contact
    Zero or One. Provides contact information for the person that
    performed the action documented in this class.

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

DefinedCOA
    Zero or more. ML_STRING. A unique identifier meaningful to the
    sender and recipient of this document that references a course of
    action. This class MUST be present if the action attribute is set
    to "defined-coa".

AdditionalData
    Zero or more. A mechanism by which to extend the data model.

The HistoryItem class has five attributes:

restriction
    Optional. ENUM. See Section 3.3.1.

ext-restriction
    Optional. STRING. A means by which to extend the restriction
    attribute. See Section 5.1.1.

action
    Required. 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 the action attribute of the Expectation class. The difference
    is only one of tense. When an action is in this class, it has
    been completed. See Section 3.17.

ext-action

      Optional. STRING. A means by which to extend the action
      attribute. See Section 5.1.1.

   observable-id
      Optional. ID. See Section 3.3.2.

3.16. EventData Class

   The EventData class describes a particular event of the incident for
   a given 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    ]
| STRING ext-restriction  |<>--{0..1}--[ DetectTime     ]
| ID observable-id        |<>--{0..1}--[ StartTime      ]
|                         |<>--{0..1}--[ EndTime        ]
|                         |<>--{0..1}--[ RecoveryTime   ]
|                         |<>--{0..1}--[ ReportTime     ]
|                         |<>--{0..*}--[ Contact        ]
|                         |<>--{0..*}--[ Discovery      ]
|                         |<>--{0..1}--[ Assessment     ]
|                         |<>--{0..*}--[ Method         ]
|                         |<>--{0..*}--[ Flow           ]
|                         |<>--{0..*}--[ Expectation    ]
|                         |<>--{0..1}--[ Record         ]
|                         |<>--{0..*}--[ EventData      ]
|                         |<>--{0..*}--[ AdditionalData ]
+-------------------------+
```

                    Figure 28: 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.

RecoveryTime
   Zero or one. The time the site recovered from the event.

ReportTime
   One. The time the event was reported.

Contact
   Zero or more. Contact information for the parties involved in the
   event.

Discovery
   Zero or more. The means by which the event was detected.

Assessment
   Zero or one. The impact of the event on the target and the
   actions taken.

Method
   Zero or more. The technique used by the intruder in the event.

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

Expectation
   Zero or more. The expected action to be performed by the
   recipient for the described event.

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 more. An extension mechanism for data not explicitly
   represented in the data model.

At least one of the aggregate classes MUST be present in an instance
of the EventData class. This is not enforced in the IODEF schema as
there is no simple way to accomplish it.

The EventData class has three attributes:

restriction
   Optional. ENUM. See Section 3.3.1. The default value is
   "default".

ext-restriction
   Optional. STRING. A means by which to extend the restriction
   attribute. See Section 5.1.1.

observable-id
   Optional. ID. See Section 3.3.2.

3.16.1. 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 Incident.

3.16.2. 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 relate 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 29.

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

                Figure 29: Recursion in the EventData Class

3.17. Expectation Class

The Expectation class conveys to the recipient of the IODEF document
the actions the sender is requesting. The scope of the requested
action is limited to purview of the EventData class in which this
class is aggregated.

```
+-------------------------+
| Expectation             |
+-------------------------+
| ENUM restriction        |<>--{0..*}--[ Description ]
| STRING ext-restriction  |<>--{0..*}--[ DefinedCOA  ]
| ENUM severity           |<>--{0..1}--[ StartTime   ]
| ENUM action             |<>--{0..1}--[ EndTime      ]
| STRING ext-action       |<>--{0..1}--[ Contact      ]
| ID observable-id        |
|                         |
+-------------------------+
```

                   Figure 30: The Expectation Class

The aggregate classes that constitute Expectation are:

Description
   Zero or more. ML_STRING. A free-form description of the desired
   action(s).

DefinedCOA
   Zero or more. ML_STRING. A unique identifier meaningful to the
   sender and recipient of this document that references a course of
   action. This class MUST be present if the action attribute is set
   to "defined-coa".

StartTime
   Zero or one. The time at which the sender would like the action
   performed. A timestamp that is earlier than the ReportTime
   specified in the Incident class denotes that the sender would like
   the action performed as soon as possible. The absence of this
   element indicates no expectations of when the recipient would like
   the action performed.

EndTime
   Zero or one. The time by which the sender expects the recipient
   to complete the action. If the recipient cannot complete the
   action before EndTime, the recipient MUST NOT carry out the
   action. Because of transit delays, clock drift, and so on, the
   sender MUST be prepared for the recipient to have carried out the
   action, even if it completes past EndTime.

Contact
   Zero or one. The expected actor for the action.

The Expectations class has six attributes:

restriction
   Optional. ENUM. See Section 3.3.1. The default value is
   "default".

ext-restriction
   Optional. STRING. A means by which to extend the restriction
   attribute. See Section 5.1.1.

severity
   Optional. ENUM. Indicates the desired priority of the action.
   This attribute is an enumerated list with no default value, and
   the semantics of these relative measures are context dependent.

   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 a default value of "other".
These values are maintained in the "Expectation-action" IANA
registry per Table 1.

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 systems(s) listed in the event.

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

7.   block-network. Block traffic from the network(s) lists as
     sources in the event.

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

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

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

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

12. redirect-traffic. Redirect traffic from intended recipient
     for further analysis.

13. honeypot. Redirect traffic to a honeypot for further
     analysis.

14. upgrade-software. Upgrade or patch the software or firmware
     on an asset.

15. rebuild-asset. Reinstall the operating system or
     applications on an asset.

16. harden-asset. Change the configuration an asset (e.g.,
     reduce the number of services or user accounts) to reduce the
     attack surface.

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

18. status-triage. Conveys receipts and the triaging of an
     incident.

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

20. watch-and-report. Watch for the described activity and share
     if seen.

21. training. Train user to identify or mitigate a threat.

22. defined-coa. Perform a predefined course of action (COA).
     The COA is named in the DefinedCOA class.

23. other. Perform some custom action described in the
     Description class.

24. ext-value. An escape value used to extend this attribute.
     See Section 5.1.1.

ext-action
   Optional. STRING. A means by which to extend the action
   attribute. See Section 5.1.1.

observable-id
   Optional. ID. See Section 3.3.2.

3.18. Flow Class

The Flow class groups related the source and target hosts.

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

Figure 31: The Flow Class

The aggregate class that constitutes Flow is:

   System
      One or More. A host or network involved in an event.

   The Flow class has no attributes.

[3.19](#). **System Class**

   The System class describes a system or network involved in an event.
   The systems or networks 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. A value of "sensor" dictates that this System was part
   of an instrumentation to monitor the network.

```
   +------------------------+
   | System                 |
   +------------------------+
   | ENUM restriction       |<>----------[ Node            ]
   | STRING ext-restriction |<>--{0..*}--[ NodeRole        ]
   | ENUM category          |<>--{0..*}--[ Service         ]
   | STRING ext-category    |<>--{0..*}--[ OperatingSystem ]
   | STRING interface       |<>--{0..*}--[ Counter         ]
   | ENUM spoofed           |<>--{0..*}--[ AssetID         ]
   | ENUM virtual           |<>--{0..*}--[ Description     ]
   | ENUM ownership         |<>--{0..*}--[ AdditionalData  ]
   | STRING ext-ownership   |
   |                        |
   +------------------------+
```

                        Figure 32: The System Class

   The aggregate classes that constitute System are:

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

   NodeRole
      Zero or more. The intended purpose of the system.

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

   OperatingSystem

Zero or more. The operating system running on the system.

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

AssetID
   Zero or more. An asset identifier for the System.

Description
   Zero or more. ML_STRING. A free-form text description of the
   System.

AdditionalData
   Zero or more. A mechanism by which to extend the data model.

The System class has nine attributes:

restriction
   Optional. ENUM. See Section 3.3.1.

ext-restriction
   Optional. STRING. A means by which to extend the restriction
   attribute. See Section 5.1.1.

category
   Optional. ENUM. Classifies the role the host or network played
   in the incident. These values are maintained in the "System-
   category" IANA registry per Table 1. The possible values are:

   1. source. The System was the source of the event.

   2. target. The System was the target of the event.

   3. intermediate. The System was an intermediary in the event.

   4. sensor. The System was a sensor monitoring the event.

   5. infrastructure. The System was an infrastructure node of
       IODEF document exchange.

   6. ext-value. An escape value used to extend this attribute.
       See Section 5.1.1.

ext-category
   Optional. STRING. A means by which to extend the category
   attribute. See Section 5.1.1.

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.

virtual
   Optional. ENUM. Indicates whether this System is a virtual or
   physical device. The default value is "unknown". The possible
   values are:

   1. yes. The System is a virtual device.

   2. no. The System is a physical device.

   3. unknown. It is not known if the System is virtual.

ownership
   Optional. ENUM. Describes the ownership of this System relative
   to the sender of the IODEF document. These values are maintained
   in the "System-ownership" IANA registry per Table 1. The possible
   values are:

   1. organization. The System is owned by the organization.

   2. personal. The System is owned by employee or affiliate of the
       organization.

   3. partner. The System is owned by a partner of the
       organization.

   4. customer. The System is owned by a customer of the
       organization.

    5. no-relationship. The System is owned by an entity that has no
        known relationship with the organization.

    6. unknown. The ownership of the System is unknown.

    7. ext-value. An escape value used to extend this attribute.
        See Section 5.1.1.

  ext-ownership
     Optional. STRING. A means by which to extend the ownership
     attribute. See Section 5.1.1.

3.20. Node Class

   The Node class names an asset or network.

   This class was derived from [RFC4765].

```
    +---------------+
    | Node          |
    +---------------+
    |               |<>--{0..*}--[ DomainData    ]
    |               |<>--{0..*}--[ Address       ]
    |               |<>--{0..1}--[ PostalAddress ]
    |               |<>--{0..*}--[ Location      ]
    |               |<>--{0..1}--[ DateTime      ]
    |               |<>--{0..*}--[ Counter       ]
    +---------------+
```

                       Figure 33: The Node Class

   The aggregate classes that constitute Node are:

   DomainData
      Zero or more. The detailed domain (DNS) information associated
      with this Node. If an Address is not provided, at least one
      DomainData MUST be specified.

   Address
      Zero or more. The hardware, network, or application address of
      the Node. If a DomainData is not provided, at least one Address
      MUST be specified.

   PostalAddress
      Zero or one. The postal address of the asset.

   Location

      Zero or more. ML_STRING. A free-from description of the physical
      location of the Node. This description may provide a more
      detailed description of where in the PostalAddress this Node is
      found (e.g., room number, rack number, slot number in a chassis).

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

   The Node class has no attributes.

3.20.1. Address Class

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

   This class was derived from [RFC4765].

```
   +-------------------------+
   | Address                 |
   +-------------------------+
   | ENUM category           |
   | STRING ext-category     |
   | STRING vlan-name        |
   | INTEGER vlan-num        |
   | ID observable-id        |
   +-------------------------+
```

                     Figure 34: The Address Class

   The Address class has five attributes:

   category
      Optional. ENUM. The type of address represented. The permitted
      values for this attribute are shown below. The default value is
      "ipv4-addr". These values are maintained in the "Address-
      category" IANA registry per Table 1.

      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 (i.e., a.b.c.d/nn)

   6.   ipv4-net-mask. IPv4 network address in dotted-decimal
        notation, slash, network mask in dotted-decimal notation
        (i.e., 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 (i.e., a:b:c:d:e:f)

   11. site-uri. A URL or URI for a resource.

   12. ext-value. An escape value used to extend this attribute.
        See Section 5.1.1.

ext-category
   Optional. STRING. A means by which to extend the category
   attribute. See Section 5.1.1.

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.

observable-id
   Optional. ID. See Section 3.3.2.

3.20.2. NodeRole Class

   The NodeRole class describes the function performed by a particular .

```
+---------------------+
| NodeRole            |
+---------------------+
| ENUM category       |
| STRING ext-category |
| ENUM xml:lang       |
+---------------------+
```

                  Figure 35: The NodeRole Class

The NodeRole class has three attributes:

category
    Required. ENUM. Functionality provided by a node. These values
    are maintained in the "NodeRole-category" IANA registry per
    Table 1.

    1.   client. Client computer

    2.   client-enterprise. Client computer on the enterprise network

    3.   client-partner. Client computer on network of a partner

    4.   client-remote. Client computer remotely connected to the
         enterprise network

    5.   client-kiosk. Client computer is serves as a kiosk

    6.   client-mobile. Client is a mobile device

    7.   server-internal. Server with internal services

    8.   server-public. Server with public services

    9.   www. WWW server

    10. mail. Mail server

    11. webmail. Web mail server

    12. messaging. Messaging server (e.g., NNTP, IRC, IM)

    13. streaming. Streaming-media server

    14. voice. Voice server (e.g., SIP, H.323)

    15. file. File server (e.g., SMB, CVS, AFS)

    16. ftp. FTP server

    17. p2p. Peer-to-peer node

    18. name. Name server (e.g., DNS, WINS)

    19. directory. Directory server (e.g., LDAP, finger, whois)

    20. credential. Credential server (e.g., domain controller,
         Kerberos)

21. print. Print server

22. application. Application server

23. database. Database server

24. backup. Backup server

25. dhcp. DHCP server

26. assessment. Assessment server (e.g., vulnerability scanner,
     end-point assessment)

27. source-control. Source code control server

28. config-management. Configuration management server

29. monitoring. Security monitoring server (e.g., IDS)

30. infra. Infrastructure server (e.g., router, firewall, DHCP)

31. infra-firewall. Firewall

32. infra-router. Router

33. infra-switch. Switch

34. camera. Camera and video system

35. proxy. Proxy server

36. remote-access. Remote access server

37. log. Log server (e.g., syslog)

38. virtualization. Server running virtual machines

39. pos. Point-of-sale device

40. scada. Supervisory control and data acquisition system

41. scada-supervisory. Supervisory system for a SCADA

42. sinkhole. Traffic sinkhole destination

43. honeypot. Honeypot server

44. anonymization. Anonymization server (e.g., Tor node)

45. c2. Malicious command and control server

46. malware-distribution. Server that distributes malware

47. drop-server. Server to which exfiltrated content is
    uploaded.

48. hop-point. Intermediary server used to get to a victim.

49. reflector. A system used in a reflector attacker.

50. phishing-site. Site hosting phishing content

51. spear-phishing-site. Site hosting spear-phishing content

52. recruiting-site. Site to recruit

53. fraudulent-site. Fraudulent site.

54. ext-value. An escape value used to extend this attribute.
    See Section 5.1.1.

ext-category
   Optional. STRING. A means by which to extend the category
   attribute. See Section 5.1.1.

xml:lang
   Optional. ENUM. A language identifier per Section 2.12 of
   [W3C.XML] whose values and form are described in [RFC5646]. The
   interpretation of this code is described in Section 6.

3.20.3. 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 dependent based on the class in which the
Counter is aggregated.

```
+--------------------+
| Counter            |
+--------------------+
| REAL               |
|                    |
| ENUM type          |
| STRING ext-type    |
| ENUM unit          |
| STRING ext-unit    |
| STRING meaning     |
| ENUM duration      |
| STRING ext-duration |
+--------------------+
```

                    Figure 36: The Counter Class

   The Counter class has seven attribute:

   type
      Required. ENUM. Specifies the type of counter specified in the
      element content. These values are maintained in the "Counter-
      type" IANA registry per Table 1.

      1. count. The Counter class value is a counter.

      2. peak. The Counter class value is a peak value.

      3. average. The Counter class value is an average.

      4. ext-value. An escape value used to extend this attribute.
          See [Section 5.1.1](#).

   ext-type
      Optional. STRING. A means by which to extend the type attribute.
      See [Section 5.1.1](#).

   unit
      Required. ENUM. Specifies the units of the element content.
      These values are maintained in the "Counter-unit" IANA registry
      per Table 1.

      1.   byte. Bytes transferred.

      2.   mbit. Megabits (Mbits) transfered.

      3.   packet. Packets.

      4.   flow. Network flow records.

   5. session. Sessions.

   6. alert. Notifications generated by another system (e.g., IDS
       or SIM).

   7. message. Messages (e.g., mail messages).

   8. event. Events.

   9. host. Hosts.

   10. site. Site.

   11. organization. Organizations.

   12. ext-value. An escape value used to extend this attribute.
       See Section 5.1.1.

   ext-unit
      Optional. STRING. A means by which to extend the unit attribute.
      See Section 5.1.1.

   meaning
      Optional. STRING. A free-form description of the metric
      represented by the Counter.

   duration
      Optional. ENUM. If present, the Counter class represents a rate.
      This attribute specifies unit of time over which the rate whose
      units are specified in the unit attribute is being conveyed. This
      attribute is the the denominator of the rate (where the unit
      attribute specified the nominator). The possible values of this
      attribute are defined in Section 3.14.3

   ext-duration
      Optional. STRING. A means by which to extend the duration
      attribute. See Section 5.1.1.

3.21. DomainData Class

   The DomainData class describes a domain name and meta-data associated
   with this domain.

```
+-------------------------+
| DomainData              |
+-------------------------+
| ENUM system-status      |<>----------[ Name                  ]
| STRING ext-system-status|<>--{0..1}--[ DateDomainWasChecked ]
| ENUM domain-status      |<>--{0..1}--[ RegistrationDate     ]
| STRING ext-domain-status|<>--{0..1}--[ ExpirationDate       ]
| ID observable-id        |<>--{0..*}--[ RelatedDNS           ]
|                         |<>--{0..*}--[ Nameservers          ]
|                         |<>--{0..1}--[ DomainContacts       ]
|                         |
+-------------------------+
```

                    Figure 37: The DomainData Class

   The aggregate classes that constitute DomainData are:

   Name
      One. STRING. The domain name of the Node (e.g., fully qualified
      domain name).

   DateDomainWasChecked
      Zero or one. DATETIME. A timestamp of when the Name was
      resolved.

   RegistrationDate
      Zero or one. DATETIME. A timestamp of when domain listed in Name
      was registered.

   ExpirationDate
      Zero or one. DATETIME. A timestamp of when the domain listed in
      Name is set to expire.

   RelatedDNS
      Zero or more. Additional DNS records associated with this domain.

   Nameservers
      Zero or more. The name servers identified for the domain listed
      in Name.

   DomainContacts
      Zero or one. Contact information for the domain listed in Name
      supplied by the registrar or through a whois query.

   The DomainData class has five attribute:

   system-status

Required. ENUM. Assesses the domain's involvement in the event.
These values are maintained in the "DomainData-system-status" IANA
registry per Table 1.

1. spoofed. This domain was spoofed.

2. fraudulent. This domain was operated with fraudulent
    intentions.

3. innocent-hacked. This domain was compromised by a third
    party.

4. innocent-hijacked. This domain was deliberately hijacked.

5. unknown. No categorization for this domain known.

6. ext-value. An escape value used to extend this attribute.
    See Section 5.1.1.

ext-system-status
    Optional. STRING. A means by which to extend the system-status
    attribute. See Section 5.1.1.

domain-status
    Required. ENUM. Categorizes the registry status of the domain at
    the time the document was generated. These values and their
    associated descriptions are derived from Section 3.2.2 of
    [RFC3982]. These values are maintained in the "DomainData-domain-
    status" IANA registry per Table 1.

1. reservedDelegation. The domain is permanently inactive.

2. assignedAndActive. The domain is in a normal state.

3. assignedAndInactive. The domain has an assigned registration
    but the delegation is inactive.

4. assignedAndOnHold. The domain is under dispute.

5. revoked. The domain is in the process of being purged from
    the database.

6. transferPending. The domain is pending a change in
    authority.

7. registryLock. The domain is on hold by the registry.

8. registrarLock. Same as "registryLock".

9. other. The domain has a known status but it is not one of
the redefined enumerated values.

10. unknown. The domain has an unknown status.

11. ext-value. An escape value used to extend this attribute.
See Section 5.1.1.

ext-domain-status
Optional. STRING. A means by which to extend the domain-status
attribute. See Section 5.1.1.

observable-id
Optional. ID. See Section 3.3.2.

3.21.1. RelatedDNS

The RelatedDNS class describes additional record types associated
with a given domain name. The record type is described in the
record-type attribute and the value of the record is the element
content. ... TODO Issue #39 ...

```
+----------------------+
| RelatedDNS           |
+----------------------+
| STRING               |
|                      |
| ENUM record-type     |
+----------------------+
```

Figure 38: The RelatedDNS Class

The RelatedDNS class has one attribute:

record-type
Required. ENUM. The DNS record type. ... TODO values need to be
listed ...

3.21.2. Nameservers Class

The Nameservers class describes the name servers associated with a
given domain.

```
+-------------------+
| Nameservers       |
+-------------------+
|                   |<>----------[ Server  ]
|                   |<>--{1..*}--[ Address ]
+-------------------+
```

                    Figure 39: The Nameservers Class

   The aggregate classes that constitute Nameservers are:

   Server
      One. STRING. The domain name of the name server.

   Address
      One or more. The address of the name server. See Section 3.20.1.

3.21.3. DomainContacts Class

   The DomainContacts class describes the contact information for a
   given domain provided either by the registrar or through a whois
   query.

   This contact information can be explicitly described through a
   Contact class or a reference can be provided to a domain with
   identical contact information. Either a single SameDomainContact
   MUST be present or one or many Contact classes.

```
+-------------------+
| DomainContacts    |
+-------------------+
|                   |<>--{0..1}--[ SameDomainContact ]
|                   |<>--{1..*}--[ Contact ]
+-------------------+
```

                    Figure 40: The DomainContacts Class

   The aggregate classes that constitute DomainContacts are:

   SameDomainContact
      Zero or one. STRING. A domain name already cited in this
      document or through previous exchange that contains the identical
      contact information as the domain name in question. The domain
      contact information associated with this domain should be used in
      lieu of explicit definition with the Contact class.

   Contact

      One or more. Contact information for the domain. See
      Section 3.10.

3.22. 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 [RFC4765].

```
+-------------------------+
| Service                 |
+-------------------------+
| INTEGER ip-protocol     |<>--{0..1}--[ ServiceName       ]
| ID observable-id        |<>--{0..1}--[ Port              ]
|                         |<>--{0..1}--[ Portlist          ]
|                         |<>--{0..1}--[ ProtoCode         ]
|                         |<>--{0..1}--[ ProtoType         ]
|                         |<>--{0..1}--[ ProtoField        ]
|                         |<>--{0..*}--[ ApplicationHeader ]
|                         |<>--{0..1}--[ EmailData         ]
|                         |<>--{0..1}--[ Application       ]
+-------------------------+
```

                     Figure 41: The Service Class

   The aggregate classes that constitute Service are:

   ServiceName
      Zero or one. Identifies the the observed service.

   Port
      Zero or one. INTEGER. A port number.

   Portlist
      Zero or one. PORTLIST. A list of port numbers formatted
      according to Section 2.10.

   ProtoCode
      Zero or one. INTEGER. A transport layer (layer 4) protocol-
      specific code field (e.g., ICMP code field).

ProtoType
   Zero or one. INTEGER. A transport layer (layer 4) protocol
   specific type field (e.g., ICMP type field).

ProtoField
   Zero or one. INTEGER. A transport layer (layer 4) protocol
   specific flag field (e.g., TCP flag field).

ApplicationHeader
   Zero or more. An application layer (layer 7) protocol header.
   See Section 3.22.2.

EmailData
   Zero or one. Headers associated with an email. See Section 3.24.

Application
   Zero or one. The application bound to the specified Port or
   Portlist. See Section 3.22.3.

Either a Port or Portlist class MUST be specified for a given
instance of a Service class.

When a given System classes with category="source" and another with
category="target" are aggregated into a single Flow class, and each
of these System classes has a Service and Portlist class, an implicit
relationship between these Portlists exists. If N ports are listed
for a System@category="source", and M ports are listed for
System@category="target", the number of ports in N must be equal to
M. Likewise, the ports MUST be listed in an identical sequence such
that the n-th port in the source corresponds to the n-th port of the
target. If N is greater than 1, a given instance of a Flow class
MUST only have a single instance of a System@category="source" and
System@category="target".

The Service class has two attributes:

ip-protocol
   Required. INTEGER. The IANA assigned IP protocol number per
   [IANA.Protocols].

observable-id
   Optional. ID. See Section 3.3.2.

3.22.1. ServiceName Class

The ServiceName class names an application protocol. It can be
described by referencing an IANA registered protocol, a URL or with
free-form text.

```
   +--------------------+
   | ServiceName        |
   +--------------------+
   |                    |<>--{0..1}--[ IANAService        ]
   |                    |<>--{0..*}--[ URL                ]
   |                    |<>--{0..*}--[ Description        ]
   +--------------------+
```

                    Figure 42: The ServiceName Class

   The aggregate classes that constitute ServiceName:

   IANAService
      Zero or one. The name of the service per the "Service Name" field
      of the [IANA.Ports] registry.

   URL
      Zero or more. URL. A URL describing the service.

   Description
      Zero or more. ML_STRING. A free-form text description of the
      service.

   At least one of these classes MUST be present.

   The ServiceName class has no attributes.

3.22.2. ApplicationHeader Class

   The ApplicationHeader class allows the representation of arbitrary
   fields from an application layer protocol header and its
   corresponding value.

```
   +------------------------+
   | ApplicationHeader      |
   +------------------------+
   | ANY                    |
   |                        |
   | INTEGER proto          |
   | STRING proto-name      |
   | STRING field           |
   | ENUM dtype             |
   | STRING ext-dtype       |
   | ID observable-id       |
   +------------------------+
```

                 Figure 43: The ApplicationHeader Class

The ApplicationHeader class has six attributes:

proto
    Optional. INTEGER. The IANA assigned port number per the
    "Protocol Number" field of the [IANA.Ports] registry corresponding
    to the application layer protocol whose field will be represented.

proto-name
    Optional. STRING. The IANA assigned service name per the
    "Service Name" field of the the [IANA.Ports] registry
    corresponding to the application layer protocol whose field will
    be represented.

field
    Required. STRING. The name of the protocol field whose value
    will be found in the element body.

dtype
    Required. ENUM. The data type of the element content. The
    permitted values for this attribute are shown below. The default
    value is "string". These values are maintained in the
    "ApplicationHeader-proto-dtype" IANA registry per Table 1.

    1.   boolean. The element content is of type BOOLEAN.

    2.   byte. The element content is of type BYTE.

    3.   bytes. The element content is of type HEXBIN.

    4.   character. The element content is of type CHARACTER.

    5.   date-time. The element content is of type DATETIME.

    6.   integer. The element content is of type INTEGER.

    7.   portlist. The element content is of type PORTLIST.

    8.   real. The element content is of type REAL.

    9.   string. The element content is of type STRING.

    10. file. The element content is a base64 encoded binary file
         encoded as a BYTE[] type.

    11. path. The element content is a file-system path encoded as a
         STRING type.

    12. xml. The element content is XML. See Section 5.

        13. ext-value. An escape value used to extend this attribute.
             See Section 5.1.1.

   ext-dtype
      Optional. STRING. A means by which to extend the dtype
      attribute. See Section 5.1.1.

   observable-id
      Optional. ID. See Section 3.3.2.

   Either the proto or proto-name attribute MUST be set. If both are
   set, they MUST correspond to the same entry in the registry.

3.22.3. Application Class

   The Application class describes a software application. It can be
   described by using formal reference, a URL or with free-form text.

```
   +-------------------+
   | Application       |
   +-------------------+
   |                   |<>--{0..1}--[ SoftwareReference ]
   |                   |<>--{0..*}--[ URL               ]
   |                   |<>--{0..*}--[ Description       ]
   +-------------------+
```

                   Figure 44: The Application Class

   The aggregate classes that constitute Application:

   SoftwareReference
      Zero or one. Reference to a software application.

   URL
      Zero or more. URL. A URL associated with the application.

   Description
      Zero or more. ML_STRING. A free-form text description of this
      application.

   At least one of these classes MUST be present.

   The Application class has no attributes.

3.22.4. **SoftwareReference Class**

The Application class describes a software application. It can be
described by using formal reference, a URL or with free-form text.

```
+---------------------+
| SoftwareReference   |
+---------------------+
| ANY                 |
|                     |
| ENUM spec-name      |
| STRING ext-spec-name |
| ENUM dtype          |
| STRING enum-dtype   |
+---------------------+
```

Figure 45: The SoftwareReference Class

The element body of this class varies according to the value of the
spec-name attribute.

The SoftwareReference class has four attributes:

spec-name
   Required. ENUM. Identifies the format and semantics of the the
   element body of this class. Formal standards and specifications
   can be referenced as well as free-form description with user-
   provided data-types. These values are maintained in the
   "SoftwareReference-spec-id" IANA registry per Table 1

   1. custom. The element content is free-form and of the data type
       specified by the dtype attribute. If this value is selected,
       then the dtype attribute MUST be set.

   2. cpe. The element content describes a Common Platform
       Enumeration (CPE) entry [fix me. reference].

   3. swid. The element content describes a software identification
       (SWID) tag per ISO/IEC 19770-2:2009 [fix me. reference].

   4. ext-value. An escape value used to extend this attribute.
       See Section 5.1.1.

ext-spec-name
   Optional. STRING. A means by which to extend the spec-name
   attribute. See Section 5.1.1.

dtype

Optional. ENUM. The data type of the element content. The
permitted values for this attribute are shown below. The default
value is "string". These values are maintained in the
"SoftwareReference-dtype" IANA registry per Table 1.

1. bytes. The element content is of type HEXBIN.

2. integer. The element content is of type INTEGER.

3. real. The element content is of type REAL.

4. string. The element content is of type STRING.

5. xml. The element content is XML. See Section 5.

6. ext-value. An escape value used to extend this attribute.
    See Section 5.1.1.

ext-dtype
   Optional. STRING. A means by which to extend the dtype
   attribute. See Section 5.1.1.

3.23. OperatingSystem Class

The OperatingSystem class describes the operating system running on a
System. The definition is identical to the Application class
(Section 3.22.3).

3.24. EmailData Class

The EmailData class describes headers from an email message. Common
headers have dedicated classes, but arbitrary headers can also be
described.

```
+-------------------------+
| EmailData               |
+-------------------------+
| ID observable-id        |<>--{0..1}--[ EmailFrom        ]
|                         |<>--{0..1}--[ EmailSubject     ]
|                         |<>--{0..1}--[ EmailX-Mailer    ]
|                         |<>--{0..*}--[ EmailHeaderField ]
|                         |<>--{0..*}--[ HashData         ]
|                         |<>--{0..*}--[ SignatureData    ]
+-------------------------+
```

                     Figure 46: EmailData Class

The aggregate class that constitutes EmailData are:

EmailFrom
   Zero or one. The value of the "From:" header field in an email.
   See Section 3.6.2 of [RFC5322].

EmailSubject
   Zero or one. The value of the "Subject:" header field in an
   email. See Section 3.6.4 of [RFC5322].

EmailX-Mailer
   Zero or one. The value of the "X-Mailer:" header field in an
   email.

EmailHeaderField
   Zero or one. The value of an arbitrary header field in the email.
   See Section 3.22.2. The attributes of EmailHeaderField MUST be
   set as follows: proto="25" or proto-name="smtp", or both can be
   set; and dtype="string". The name of the email header field MUST
   be set in the field attribute.

HashData
   Zero or One. Hash(es) associated with this email.

SignatureData
   Zero or One. Signature(s) associated with this email.

The EmailData class has one attribute:

observable-id
   Optional. ID. See Section 3.3.2.

3.25. 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. These logs
substantiate the activity described in the document.

```
+------------------------+
| Record                 |
+------------------------+
| ENUM restriction       |<>--{1..*}--[ RecordData ]
| STRING ext-restriction |
+------------------------+
```

                        Figure 47: Record Class

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 two attributes:

   restriction
      Optional. ENUM. See Section 3.3.1.

   ext-restriction
      Optional. STRING. A means by which to extend the restriction
      attribute. See Section 5.1.1.

3.25.1. 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             |
   +------------------------+
   | ENUM restriction       |<>--{0..1}--[ DateTime             ]
   | STRING ext-restriction |<>--{0..*}--[ Description          ]
   | ID observable-id       |<>--{0..1}--[ Application          ]
   |                        |<>--{0..*}--[ RecordPattern        ]
   |                        |<>--{0..*}--[ RecordItem           ]
   |                        |<>--{0..*}--[ FileData             ]
   |                        |<>--{0..*}--[ CertificateData      ]
   |                        |<>--{0..*}--
   |                        |      [ WindowsRegistryKeysModified ]
   |                        |<>--{0..*}--[ AdditionalData       ]
   +------------------------+
```

                   Figure 48: 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.

   RecordPattern
      Zero or more. A search string to precisely find the relevant data
      in a RecordItem.

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

   FileData
      Zero or one. The file name and hash of a file indicator.

   WindowsRegistryKeysModified
      Zero or more. The registry keys that were modified that are
      indicator(s).

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

   The RecordData class has three attributes:

   restriction
      Optional. ENUM. See Section 3.3.1.

   ext-restriction
      Optional. STRING. A means by which to extend the restriction
      attribute. See Section 5.1.1.

   observable-id
      Optional. ID. See Section 3.3.2.

3.25.2. 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            |
| STRING ext-type      |
| INTEGER offset       |
| ENUM offsetunit      |
| STRING ext-offsetunit |
| INTEGER instance     |
+----------------------+
```

                   Figure 49: 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 six attributes:

   type
      Required. ENUM. Describes the type of pattern being specified in
      the element content. The default is "regex". These values are
      maintained in the "RecordPattern-type" IANA registry per Table 1.

      1. regex. regular expression as defined by POSIX Extended
          Regular Expressions (ERE) in Chapter 9 of [IEEE.POSIX].

      2. binary. Binhex encoded binary pattern, per the HEXBIN data
          type.

      3. xpath. XML Path (XPath) [W3C.XPATH]

      4. ext-value. An escape value used to extend this attribute.
          See Section 5.1.1.

   ext-type
      Optional. STRING. A means by which to extend the type attribute.
      See Section 5.1.1.

   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". These values are maintained in the
      "RecordPattern-offsetunit" IANA registry per Table 1.

1. line. Offset is a count of lines.

2. byte. Offset is a count of bytes.

3. ext-value. An escape value used to extend this attribute.
       See Section 5.1.1.

ext-offsetunit
   Optional. STRING. A means by which to extend the offsetunit
   attribute. See Section 5.1.1.

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

3.25.3. 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 3.9).

3.26. WindowsRegistryKeysModified Class

The WindowsRegistryKeysModified class describes Windows operating
system registry keys and the operations that were performed on them.
This class was derived from [RFC5901].

```
+-----------------------------+
| WindowsRegistryKeysModified |
+-----------------------------+
| ID observable-id            |<>--{1..*}--[ Key ]
+-----------------------------+
```

             Figure 50: The WindowsRegistryKeysModified Class

The aggregate class that constitutes the WindowsRegistryKeysModified
class is:

Key
   One or many. The Window registry key.

The WindowsRegistryKeysModified class has one attribute:

observable-id

Optional. ID. See Section 3.3.2.

3.26.1. **Key Class**

The Key class describes a particular Windows operating system
registry key name and value pair, and the operation performed on it.

```
+---------------------------+
| Key                       |
+---------------------------+
| ENUM registryaction       |<>----------[ KeyName ]
| STRING ext-registryaction |<>--{0..1}--[ KeyValue ]
| ID observable-id          |
+---------------------------+
```

Figure 51: The Key Class

The aggregate classes that constitutes Key are:

KeyName
    One. STRING. The name of the Windows operating system registry
    key (e.g.,[HKEY_LOCAL_MACHINE\Software\Test\KeyName])

KeyValue
    Zero or one. STRING. The value of the associated registry key
    encoded as in Microsoft .reg files [KB310516].

The Key class has three attributes:

registryaction
    Optional. ENUM. The type of action taken on the registry key.
    These values are maintained in the "Key-registryaction" IANA
    registry per Table 1.

    1. add-key. Registry key added.

    2. add-value. Value added to registry key.

    3. delete-key. Registry key deleted.

    4. delete-value. Value deleted from registry key.

    5. modify-key. Registry key modified.

    6. modify-value. Value modified for registry key.

    7. ext-value. An escape value used to extend this attribute.
        See Section 5.1.1.

   ext-registryaction
      Optional. STRING. A means by which to extend the registryaction
      attribute. See Section 5.1.1.

   observable-id
      Optional. ID. See Section 3.3.2.

3.27. CertificateData Class

   The CertificateData class describes X.509 certificates.

```
+------------------------+
| CertificateData        |
+------------------------+
| ID observable-id       |<>--{1..*}--[ Certificate    ]
| ENUM restriction       |
| STRING ext-restriction |
+------------------------+
```

                   Figure 52: The CertificateData Class

   The aggregate classes that constitutes CertificateData are:

   Certificate
      One or more. A certificate.

   The CertificateData class has three attributes:

   observable-id
      Optional. ID. See Section 3.3.2.

   restriction
      Optional. ENUM. See Section 3.3.1.

   ext-restriction
      Optional. STRING. A means by which to extend the restriction
      attribute. See Section 5.1.1.

3.27.1. Certificate Class

   The Certificate class describes a given X.509 certificate or
   certificate chain.

```
   +-------------------------+
   | Certificate             |
   +-------------------------+
   | ENUM valid              |<>----------[ ds: X509Data   ]
   | ID observable-id        |
   +-------------------------+
```

Figure 53: The Certificate Class

The aggregate classes that constitutes Certificate are:

ds:X509Data
   One. A given X.509 certificate or chain. See Section 4.4.4 of
   [W3C.XMLSIG].

The Certificate class has one attribute:

valid
   Optional. Indicates whether a given certificate has a valid
   signature. An invalid signature may be due to an invalid
   certificate chain, a signature not decoding properly, or a
   certificate contents not matching the hash.

   1. yes. The certificate is valid.

   2. no. The certificate is not valid.

observable-id
   Optional. ID. See Section 3.3.2.

3.28. FileData Class

The FileData class describes files of interest identified during the
analysis of an incident.

```
   +------------------------+
   | FileData               |
   +------------------------+
   | ID observable-id       |<>--{1..*}--[ File      ]
   | ENUM restriction       |
   | STRING ext-restriction |
   +------------------------+
```

Figure 54: The FileData Class

The aggregate class that constitutes FileData is:

File

One or more. A description of a file.

The FileData class has three attributes:

observable-id
    Optional. ID. See Section 3.3.2.

restriction
    Optional. ENUM. See Section 3.3.1.

ext-restriction
    Optional. STRING. A means by which to extend the restriction
    attribute. See Section 5.1.1.

3.28.1. File Class

The File class describes a file and its associated meta data.

```
+-----------------------+
| File                  |
+-----------------------+
| ID observable-id      |<>--{0..1}--[ FileName         ]
|                       |<>--{0..1}--[ FileSize         ]
|                       |<>--{0..1}--[ FileType         ]
|                       |<>--{0..*}--[ URL              ]
|                       |<>--{0..1}--[ HashData         ]
|                       |<>--{0..1}--[ SignatureData    ]
|                       |<>--{0..1}--[ AssociatedSoftware ]
|                       |<>--{0..*}--[ FileProperties   ]
+-----------------------+
```

                    Figure 55: The File Class

The aggregate classes that constitutes File are:

FileName
    Zero or One. STRING. The name of the file.

FileSize
    Zero or One. INTEGER. The size of the file in bytes.

FileType
    Zero or One. STRING. The type of file per the IANA Media Types
    Registry [IANA.Media]. Valid values correspond to the text in the
    "Template" column (e.g., "application/pdf").

URL
    Zero or more. A URL reference to the file.

   HashData
      Zero or One. Hash(es) associated with this file.

   SignatureData
      Zero or One. Signature(s) associated with this file.

   AssociatedSoftware
      Zero or One. The software application or operating system to
      which this file belongs. See Section 3.22.3 for the definition.

   FileProperties
      Zero or more. Mechanism by which to extend the data model to
      describe properties of the file. See Section 3.9.

   The File class has one attribute:

   observable-id
      Optional. ID. See Section 3.3.2.

3.29. HashData Class

   The HashData class describes different types of hashes on an given
   object (e.g., file, part of a file, email).

```
   +-------------------------+
   | HashData                |
   +-------------------------+
   | ENUM scope              |<>--{0..1}--[ HashTarget   ]
   |                         |<>--{0..*}--[ Hash         ]
   |                         |<>--{0..*}--[ FuzzyHash    ]
   +-------------------------+
```

                      Figure 56: The HashData Class

   The aggregate classes that constitutes HashData are:

   HashTarget
      Zero or One. An identifier that references a a subset of the
      object per the @scope attribute.

   Hash
      Zero or more. The hash generated on the object.

   FuzzyHash
      Zero or more. The fuzzy hash of the object.

   A single instance of Hash or FuzzyHash MUST be present.

The HashData class has one attribute:

scope
    Required. ENUM. Describes the scope of the hash on a type of
    object. These values are maintained in the "HashData-scope" IANA
    registry per Table 1.

    1. file-contents. A hash computed over the entire contents of a
        file.

    2. file-pe-section. A hash computed on a given section of a
        Windows Portable Executable (PE) file. If set to this value,
        the HashTargetId class MUST identify the section being hashed.
        This section is identified by an ordinal number (starting at
        1) corresponding to the the order in which the given section
        header was defined in the Section Table of the PE file header.

    3. file-pe-iat. A hash computed on the Import Address
        Table (IAT) of a PE file. As IAT hashes are often tool
        dependent, if this value is set, the HashTargetId class MUST
        specify the tool used to generate the hash.

    4. file-pe-resource. A hash computed on a given resource in a PE
        file. If set to this value, the HashTargetId class MUST
        identify the resource being hashed. This resource is
        identified by an ordinal number (starting at 1) corresponding
        to the oder in which the given resource is declared in the
        Resource Directory of the Data Dictionary in the PE file
        header.

    5. file-pdf-object. A hash computed on a given object in a
        Portable Document Format (PDF) file. If set to this value,
        the HashTargetId class MUST identify the object being hashed.
        This object is identified by its offset in the PDF file.

    6. email-hash. A hash computed over the headers and body of an
        email message.

    7. email-headers-hash. A hash computed over all of the headers
        of an email message.

    8. email-body-hash. A hash computed over the body of an email
        message.

    9. ext-value. An escape value used to extend this attribute.
        See Section 5.1.1.

    ext-scope

      Optional. STRING. A means by which to extend the scope
      attribute. See Section 5.1.1.

3.29.1. Hash Class

   The Hash class describes a specific hash value, algorithm, and an
   application used to generate it.

```
   +----------------+
   | Hash           |
   +----------------+
   |                |<>----------[ ds:DigestMethod            ]
   |                |<>----------[ ds:DigestValue             ]
   |                |<>--{0..1}--[ ds:CannonicalizationMethod ]
   |                |<>--{0..1}--[ Application                ]
   +----------------+
```

                      Figure 57: The Hash Class

   The aggregate classes that constitutes Hash are:

   ds:DigestMethod
      One. The hash algorithm used to generate the hash. See
      Section 4.3.3.5 of [W3C.XMLSIG]

   ds:DigestValue
      One. The computed hash value. See Section 4.3.3.6 of
      [W3C.XMLSIG].

   ds:CannonicalizationMethod
      Zero or one. The canonicalization method used for the has. See
      Section 4.3.1 of [W3C.XMLSIG].

   Application
      Zero or One. The application used to calculate the hash.

   The HashData class has no attribute:

3.29.2. FuzzyHash Class

   The FuzzyHash class describes a fuzzy hash (in an extensible way) and
   the application used to generate it.

```
+-------------------------+
| FuzzyHash               |
+-------------------------+
|                         |<>--{0..*}--[ AdditionalData ]
|                         |<>--{0..1}--[ Application    ]
+-------------------------+
```
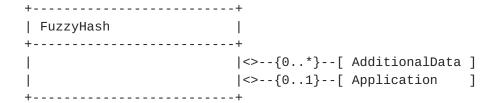
                      Figure 58: The FuzzyHash Class

   The aggregate classes that constitutes FuzzyHash are:

   AdditionalData
      Zero or more. Mechanism by which to extend the data model. See
      Section 3.9.

   Application
      Zero or One. The application used to calculate the hash.

   The FuzzyData class has no attribute:

3.30. SignatureData Class

   The SignatureData class describes different signatures on an given
   object.

```
+-------------------------+
| SignatureData           |
+-------------------------+
|                         |<>--{1..*}--[ ds:Signature ]
+-------------------------+
```

                    Figure 59: The SignatureData Class

   The aggregate classes that constitutes SignatureData are:

   Signature
      One or more. An given signature. See Section 4.2 of [W3C.XMLSIG]

   The SignatureData class has no attribute:

3.31. IndicatorData Class

   The IndicatorData class describes the indicators identified from
   analysis of an incident.

```
    +-------------------------+
    | IndicatorData           |
    +-------------------------+
    |                         |<>--{1..*}--[ Indicator    ]
    +-------------------------+
```

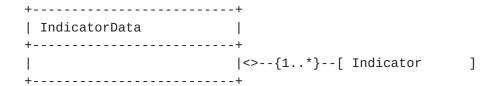   Figure 60: The IndicatorData Class

    The aggregate class that constitutes IndicatorData is:

    Indicator
        One or more. An indicator from the incident.

    The IndicatorData class has no attributes.

3.32. Indicator Class

    The Indicator class describes a cyber indicator. An indicator
    consists of observable features and phenomenon that aid in the
    forensic or proactive detection of malicious activity, and associated
    meta-data. This indicator can be described outright or reference
    observable features and phenomenon described elsewhere in the
    incident information. Portions of an incident description can be
    composed to define an indicator, as can the indicators themselves.

```
    +-----------------------+
    | Indicator             |
    +-----------------------+
    | ENUM restriction      |<>----------[ IndicatorID          ]
    | STRING ext-restriction|<>--{0..1}--[ AlternativeIndicatorID ]
    |                       |<>--{0..*}--[ Description           ]
    |                       |<>--{0..1}--[ StartTime             ]
    |                       |<>--{0..1}--[ EndTime               ]
    |                       |<>--{0..1}--[ Confidence            ]
    |                       |<>--{0..*}--[ Contact               ]
    |                       |<>--{0..1}--[ Observable            ]
    |                       |<>--{0..1}--[ ObservableReference   ]
    |                       |<>--{0..1}--[ IndicatorExpression   ]
    |                       |<>--{0..1}--[ IndicatorReference    ]
    |                       |<>--{0..*}--[ AdditionalData        ]
    +-----------------------+
```

                    Figure 61: The Indicator Class

    The aggregate classes that constitute Indicator are:

    IndicatorID
        One. An identifier for this indicator. See Section 3.32.1

AlternativeIndicatorID
   Zero or one. An alternative identifier for this indicator. See
   Section 3.32.2

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

StartTime
   Zero or one. DATETIME. A timestamp of the start of the time
   period during which this indicator is valid.

EndTime
   Zero or one. DATETIME. A timestamp of the end of the time period
   during which this indicator is valid.

Confidence
   Zero or one. An estimate of the confidence in the quality of the
   indicator. See Section 3.14.5.

Contact
   Zero or more. Contact information for this indicator. See
   Section 3.10.

Observable
   Zero or one. An observable feature or phenomenon of this
   indicator. See Section 3.32.3.

ObservableReference
   Zero or one. A reference to a feature or phenomenon defined
   elsewhere in the document. See Section 3.32.5.

IndicatorExpression
   Zero or one. A composition of observables. See Section 3.32.4.

IndicatorReference
   Zero or one. A reference to an indicator.

AdditionalData
   Zero or more. Mechanism by which to extend the data model. See
   Section 3.9

The Indicator class MUST have exactly one instance of an Observable,
IndicatorExpression, ObservableReference, or IndicatorReference
class.

The StartTime and EndTime classes can be used to define an interval
during which the indicator is valid. If both classes are present,

the indicator is consider valid only during the described interval.
If neither class is provided, the indicator is considered valid
during any time interval. If only a StartTime is provided, the
indicator is valid anytime after this timestamp. If only an EndTime
is provided, the indicator is valid anytime prior to this timestamp.

The Indicator class has two attributes:

restriction
   Optional. ENUM. See Section 3.3.1.

ext-restriction
   Optional. STRING. A means by which to extend the restriction
   attribute. See Section 5.1.1.

3.32.1. IndicatorID Class

The IndicatorID class identifies an indicator with a globally unique
identifier. The combination of the name and version attributes, and
the element content form this identifier. Indicators generated by
given CSIRT MUST NOT reuse the same value unless they are referencing
the same indicator.

```
+------------------+
| IndicatorID      |
+------------------+
| ID               |
|                  |
| STRING name      |
| STRING version   |
+------------------+
```

Figure 62: The IndicatorID Class

The IndicatorID class has two attributes:

name
   Required. STRING. An identifier describing the CSIRT that
   created the indicator. In order to have a globally unique CSIRT
   name, the fully qualified domain name associated with the CSIRT
   MUST be used. This format is identical to the IncidentID@name
   attribute in Section 3.4.

version
   Required. STRING. A version number of an indicator.

3.32.2. AlternativeIndicatorID Class

The AlternativeIndicatorID class lists alternative identifiers for an
indicator.

```
+-------------------------+
| AlternativeIndicatorID  |
+-------------------------+
| ENUM restriction        |<>--{1..*}--[ IndicatorReference ]
| STRING ext-restriction  |
+-------------------------+
```

            Figure 63: The AlternativeIndicatorID Class

The aggregate class that constitutes AlternativeIndicatorID is:

IndicatorReference
   One or more. A reference to an indicator.

The AlternativeIndicatorID class has two attributes:

restriction
   Optional. ENUM. See Section 3.3.1.

ext-restriction
   Optional. STRING. A means by which to extend the restriction
   attribute. See Section 5.1.1.

3.32.3. Observable Class

The Observable class describes a feature and phenomenon that can be
observed or measured for the purposes of detecting malicious
behavior.

```
    +-------------------+
    | Observable        |
    +-------------------+
    |                   |<>--{0..1}--[ Address                     ]
    |                   |<>--{0..1}--[ DomainData                  ]
    |                   |<>--{0..1}--[ Service                     ]
    |                   |<>--{0..1}--[ EmailData                   ]
    |                   |<>--{0..1}--[ ApplicationHeader           ]
    |                   |<>--{0..1}--[ WindowsRegistryKeysModified ]
    |                   |<>--{0..1}--[ FileData                    ]
    |                   |<>--{0..1}--[ CertificateData             ]
    |                   |<>--{0..1]--[ RegistryHandle              ]
    |                   |<>--{0..1}--[ RecordData                  ]
    |                   |<>--{0..1}--[ EventData                   ]
    |                   |<>--{0..1}--[ Incident                    ]
    |                   |<>--{0..*}--[ Expectation                 ]
    |                   |<>--{0..*}--[ Reference                   ]
    |                   |<>--{0..1}--[ Assessment                  ]
    |                   |<>--{0..1}--[ HistoryItem                 ]
    |                   |<>--{0..1}--[ BulkObservable              ]
    |                   |<>--{0..*}--[ AdditionalData              ]
    +-------------------+
```

                    Figure 64: The Observable Class

   The aggregate classes that constitute Observable are:

   Address
      Zero or One. An Address observable. See Section 3.20.1.

   DomainData
      Zero or One. A DomainData observable. See Section 3.21.

   Service
      Zero or One. A Service observable. See Section 3.22.

   EmailData
      Zero or One. A EmailData observable. See Section 3.24.

   ApplicationHeader
      Zero or One. An ApplicationHeader observable. See
      Section 3.22.2.

   WindowsRegistryKeysModified
      Zero or One. A WindowsRegistryKeysModified observable. See
      Section 3.26.

   FileData

      Zero or One. A FileData observable. See Section 3.28.

   CertificateData
      Zero or One. A CertificateData observable. See Section 3.27.

   RegistryHandle
      Zero or One. A RegistryHandle observable. See Section 3.10.1.

   RecordData
      Zero or One. A RecordData observable. See Section 3.25.1.

   EventData
      Zero or One. An EventData observable. See Section 3.16.

   Incident
      Zero or One. An Incident observable. See Section 3.2.

   EventData
      Zero or One. An EventData observable. See Section 3.16.

   Expectation
      Zero or One. An Expectation observable. See Section 3.17.

   Reference
      Zero or One. A Reference observable. See Section 3.13.1.

   Assessment
      Zero or One. An Assessment observable. See Section 3.14.

   HistoryItem
      Zero or One. A HistoryItem observable. See Section 3.15.1.

   BulkObservable
      Zero or One. A bulk list of observables. See Section 3.32.3.1.

   AdditionalData
      Zero or more. Mechanism by which to extend the data model. See
      Section 3.9.

   The Observable class MUST have exactly one of the possible child
   classes.

   The Observable class has no attributes.

3.32.3.1. **BulkObservable Class**

   The BulkObservable class allows the bulk enumeration of single type
   of observables without requiring each one to be encoded individually
   in multiple instances of the same class. The type attribute
   describes the type observable listed in the child BulkObservableList
   class. The BulkObservableFormat class optionally provides additional
   meta-data.

```
+---------------------------+
| BulkObservable            |
+---------------------------+
| ENUM type                 |<>--{0..1}--[ BulkObservableFormat ]
| STRING ext-type           |<>----------[ BulkObservableList   ]
|                           |<>--{0..*}--[ AdditionalData       ]
+---------------------------+
```

                  Figure 65: The BulkObservable Class

   The aggregate classes that constitutes BulkObservable are:

   BulkObservableFormat
      Zero or one. Provides additional meta-data about the observables
      enumerated in the BulkObservableList class.

   BulkObservableList
      One. STRING. A list of observables, one per line. Each line is
      separated with either a CR character or CR-and-LF characters. The
      type attribute will specify the which observables will be listed.

   AdditionalData
      Zero or more. Mechanism by which to extend the data model. See
      Section 3.9.

   The BulkObservable class has two attributes:

   type
      Optional. ENUM. The type of the observable listed in the child
      ObservableList class. These values are maintained in the
      "BulkObservable-type" IANA registry per Table 1.

      1. asn. Autonomous System Number (per the Address@category
          attribute).

      2. atm. Asynchronous Transfer Mode (ATM) address (per the
          Address@category attribute).

3.   e-mail. Electronic mail address ([RFC 822](#)) (per the
     Address@category attribute).

4.   ipv4-addr. IPv4 host address in dotted-decimal notation
     (e.g., 192.0.2.1) (per the Address@category attribute).

5.   ipv4-net. IPv4 network address in dotted-decimal notation,
     slash, significant bits (e.g., 192.0.2.0/24) (per the
     Address@category attribute).

6.   ipv4-net-mask. IPv4 network address in dotted-decimal
     notation, slash, network mask in dotted-decimal notation
     (i.e., 192.0.2.0/255.255.255.0) (per the Address@category
     attribute).

7.   ipv6-addr. IPv6 host address (e.g., 2001:DB8::3) (per the
     Address@category attribute).

8.   ipv6-net. IPv6 network address, slash, significant bits
     (e.g., 2001:DB8::/32) (per the Address@category attribute).

9.   ipv6-net-mask. IPv6 network address, slash, network mask
     (per the Address@category attribute).

10. mac. Media Access Control (MAC) address (i.e., a:b:c:d:e:f)
     (per the Address@category attribute).

11. site-uri. A URL or URI for a resource (per the
     Address@category attribute).

12. fqdn. Fully qualified domain name.

13. domain-name. A fully qualified domain name or part of a
     name. (e.g., fqdn.example.com, example.com).

14. domain-to-ipv4. A fqdn-to-IPv4 address mapping specified as
     a comma separated list (e.g., "fqdn.example.com, 192.0.2.1").

15. domain-to-ipv6. A fqdn-to-IPv6 address mapping specified as
     a comma separated list (e.g., "fqdn.example.com,
     2001:DB8::3").

16. domain-to-ipv4-timestamp. Same as domain-to-ipv4 but with a
     timestamp (in the DATETIME format) of the resolution (e.g.,
     "fqdn.example.com, 192.0.2.1, 2015-06-11T00:38:31-06:00").

17. domain-to-ipv6-timestamp. Same as domain-to-ipv6 but with a
     timestamp (in the DATETIME format) of the resolution (e.g.,
     "fqdn.example.com, 2001:DB8::3, 2015-06-11T00:38:31-06:00").

18. ipv4-port. An IPv4 address, port and protocol tuple (e.g.,
     192.0.2.1, 80, tcp). The protocol name corresponds to the
     "Keyword" column in the [IANA.Protocols] registry.

19. ipv6-port. An IPv6 address, port and protocol tuple (e.g.,
     2001:DB8::3, 80, tcp). The protocol name corresponds to the
     "Keyword" column in the [IANA.Protocols] registry.

20. windows-reg-key. A Microsoft Windows Registry key.

21. file-hash. A file hash. The format of this hash is
     described in the Hashclass that MUST be present in a sibling
     BulkObservableFormat class.

22. email-x-mailer. An X-Mailer field from an email.

23. email-subject. An email subject line.

24. http-user-agent. A User Agent field from an HTTP request
     header (e.g., "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:38.0)
     Gecko/20100101 Firefox/38.0").

25. http-request-uri. The Request URI from an HTTP request
     header.

26. mutex. The name of a system mutex.

27. file-path. A file path (e.g., "/tmp/local/file",
     "c:\windows\system32\file.sys")

28. user-name. A username.

29. ext-value. An escape value used to extend this attribute.
     See Section 5.1.1.

ext-type
   Optional. STRING. A means by which to extend the type attribute.
   See Section 5.1.1.

3.32.3.1.1. BulkObservableFormat Class

The ObservableFormat class specifies meta-data about the format of an
observable enumerated in a sibling BulkObservableList class.

```
   +--------------------------+
   | BulkObservableFormat     |
   +--------------------------+
   |                          |<>--{0..1}--[ Hash              ]
   |                          |<>--{0..*}--[ AdditionalData    ]
   +--------------------------+
```
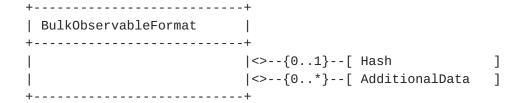
                Figure 66: The BulkObservableFormat Class

   The aggregate classes that constitutes BulkObservableFormat are:

   Hash
      Zero or one. Describes the format of a hash.

   AdditionalData
      Zero or more. Mechanism by which to extend the data model. See
      Section 3.9.

   The BulkObservableFormat class has no attributes.

   Either Hash or AdditionalData MUST be present.

3.32.4. IndicatorExpression Class

   The IndicatorExpression describes an expression composed of observed
   phenomenon or features, or indicators. Elements of the expression
   can be described directly, reference relevant data from other parts
   of a given IODEF document, or reference previously defined
   indicators.

   All child classes of a given instance of IndicatorExpression form a
   boolean algebraic expression where the operator between them is
   determined by the operator attribute. Nesting an IndicatorExpression
   in itself is akin to a parenthesis in the expression.

```
   +--------------------------+
   | IndicatorExpression      |
   +--------------------------+
   | ENUM operator            |<>--{0..*}--[ IndicatorExpression  ]
   |                          |<>--{0..*}--[ Observable           ]
   |                          |<>--{0..*}--[ ObservableReference  ]
   |                          |<>--{0..*}--[ IndicatorReference   ]
   |                          |<>--{0..*}--[ AdditionalData       ]
   +--------------------------+
```

                Figure 67: The IndicatorExpression Class

   The aggregate classes that constitute IndicatorExpression are:

   IndicatorExpression
      Zero or more. An expression composed of other observables or
      indicators.

   Observable
      Zero or more. A description of an observable.

   ObservableReference
      Zero or more. A reference to another observable.

   IndicatorReference
      Zero or more. A reference to another indicator.

   AdditionalData
      Zero or more. Mechanism by which to extend the data model. See
      Section 3.9

   ... TODO Additional text is required to describe the valid
   combinations of classes and how the operator class should be applied
   ...

   The IndicatorExpression class has one attribute:

   operator
      Optional. ENUM. The operator to be applied between the child
      elements.

      1. not. negation operator.

      2. and. conjunction operator.

      3. or. disjunction operator.

      4. xor. exclusive disjunction operator.

3.32.5. ObservableReference Class

   The ObservableReference describes a reference to an observable
   feature or phenomenon described elsewhere in the document.

   This class has no content.

```
+------------------------+
| ObservableReference    |
+------------------------+
| EMPTY                  |
|                        |
| IDREF uid-ref          |
+------------------------+
```
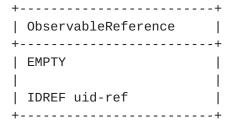
                Figure 68: The ObservableReference Class

   The ObservableReference class has one attributes:

   uid-ref
      Required. IDREF. An identifier that serves as a reference to a
      class in the IODEF document. The referenced class will have this
      identifier set in the observable-id attribute.

3.32.6. IndicatorReference Class

   The IndicatorReference describes a reference to an indicator. This
   reference may be to an indicator described in the IODEF document or
   in a previously exchanged IODEF document.

```
+------------------------+
| IndicatorReference     |
+------------------------+
| EMPTY                  |
|                        |
| IDREF uid-ref          |
| STRING euid-ref        |
| STRING version         |
+------------------------+
```
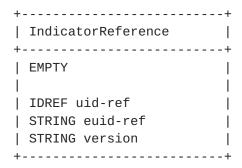
                 Figure 69: The IndicatorReference Class

   The IndicatorReference class has one attributes:

   uid-ref
      Optional. IDREF. An identifier that serves as a reference to an
      Indicator class in the IODEF document. The referenced Indicator
      class will have this identifier set in the IndicatorID class.

   euid-ref
      Optional. STRING. An identifier that references an IndicatorID
      not in this IODEF document.

   version
      Optional. STRING. A version number of an indicator.

Either the uid-ref or the euid-ref attribute MUST be set.

4. Processing Considerations

This section defines additional requirements on creating and parsing
IODEF documents.

4.1. Encoding

Every IODEF document MUST begin with an XML declaration, and MUST
specify the XML version used. The character encoding MUST also be
explicitly specified. UTF-8 [RFC3629] SHOULD be used unless UTF-16
[RFC2781] is necessary. Encodings other than UTF-8 and UTF-16 SHOULD
NOT be used. The IODEF conforms to all XML data encoding conventions
and constraints.

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

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

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

```
<?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 [RFC2978].

The following characters have special meaning in XML and MUST be
escaped with their entity reference equivalent: "&", "<", ">", "\""
(double quotation mark), and "'" (apostrophe). These entity
references are "&", "<", ">", """, and "'"
respectively.

4.2. IODEF Namespace

The IODEF schema declares a namespace of
"urn:ietf:params:xml:ns:iodef-2.0" and registers it per [W3C.XMLNS].
Each IODEF document MUST include a valid reference to the IODEF
schema using the "xsi:schemaLocation" attribute. An example of such
a declaration would look as follows:

```
<IODEF-Document
   version="2.00" lang="en-US"
   xmlns:iodef="urn:ietf:params:xml:ns:iodef-2.0"
   xsi:schemaLocation="urn:ietf:params:xmls:schema:iodef-2.0"
```

4.3. Validation

The IODEF documents MUST be well-formed XML. It is RECOMMENDED that
recipients validate the document against the schema described in
Section 8.   However, mere conformance to the schema is not sufficient
for a semantically valid IODEF document. There is additional
specification in the text of Section 3 that cannot be readily encoded
in the schema and it must also be considered by an IODEF parser. The
following is a list of discrepancies in what is more strictly
specified in the normative text (Section 3), but not enforced in the
IODEF schema:

o The elements or attributes that are defined as POSTAL, NAME,
   PHONE, and EMAIL data-types are implemented as "xs:string", but
   more rigid formatting requirements are specified in the text.

o The IODEF-Document@lang and MLStringType@lang attributes are
   declared as an "xml:language" that constrains values with a
   regular expression. However, the value of this attribute still
   needs to be validated against the list of possible enumerated
   values is defined in [RFC5646].

o The MonetaryImpact@currency attribute is declared as an
   "xs:string", but the list of valid values as defined in [ISO4217].

o All of the aggregated classes Contact and EventData are optional
   in the schema, but at least one of these aggregated classes MUST
   be present.

o There are multiple conventions that can be used to categorize a
   system using the NodeRole class or to specify software with the
   Application and OperatingSystem classes. IODEF parsers MUST
   accept incident reports that do not use these fields in accordance
   with local conventions.

o The Confidence@rating attribute determines whether the element
   content of Confidence should be empty.

o The Address@type attribute determines the format of the element
   content.

o The attributes AdditionalData@dtype and RecordItem@dtype derived
   from iodef:ExtensionType determine the semantics and formatting of
   the element content.

o Symmetry in the enumerated ports of a Portlist class is required
   between sources and targets. See Section 3.22.

o The enumerated values present in this document are a static list
   that will be incomplete over time as select attributes can be
   extended by a corresponded IANA registry. See Table 1. Hence,
   the schema to validate a given document MUST be dynamically
   generated from these registry values.

4.4. Incompatibilities with v1

Version 2 of the IODEF data model makes a number of changes to
[RFC5070]. Largely, these changes were additive in nature -- classes
and enumerated values were added. The following is a list of
incompatibilities where the data model has changed between versions:

o The IODEF-Document@version attribute is set to "2.0".

o The Service@ip_protocol attribute was renamed to @ip-protocol.

o The Node/NodeName class was removed in favor of representing
   domain names with Node/DomainData/Name class. The Node/DataTime
   class was also removed so that the Node/DomainData/
   DateDomainWasChecked class can represent the time at which the
   name to address resolution occurred.

o The Node/NodeRole class was moved to System/NodeRole.

o The Reference class is now defined by [RFC-ENUM].

o Extending enumerated values is now handled through collection of
   IANA registries. All attributes of with a name prefixed by "ext-"
   have been removed.

o The data previously represented in the Impact class is now in the
   SystemImpact and IncidentCategory classes. The Impact class has
   been removed.

o The Description class has been redefined to use xml:lang and
   @translation-id. IODEF-document also uses xml:lang.

o The semantics of Counter@type in v1 are now represented in
   Counter@unit.

o The IODEF-Document@formatid attribute has been renamed to @format-
   id.

5. Extending the IODEF

   In order to support the changing activity of CSIRTS, the IODEF data
   model will need to evolve along with them. This section discusses
   how new data elements that have no current representation in the data
   model can be incorporated into the IODEF. These techniques are
   designed so that adding new data will not require a change to the
   base IODEF schema. With proven value, well documented extensions can
   be incorporated into future versions of the specification. However,
   this approach also supports private extensions relevant only to a
   closed consortium.

5.1. Extending the Enumerated Values of Attributes

   Enumerated values of select attributes can be extended for private
   use through specially marked attributes with the "ext-" prefix.
   Likewise, each extensible attribute has a corresponding IANA registry
   to which to added public extensions.

5.1.1. Private Extension of Enumerated Values

   The data model supports a means by which to add new enumerated values
   to an attribute without public registration. For each attribute that
   supports this extension technique, there is a corresponding attribute
   in the same element whose name is identical but with a prefix of
   "ext-". This special attribute is referred to as the extension
   attribute, and the attribute being extended is referred to as an
   extensible attribute. For example, an extensible attribute named
   "foo" will have a corresponding extension attribute named "ext-foo".
   An element may have many extensible, and therefore many extension,
   attributes.

   In addition to a corresponding extension attribute, each extensible
   attribute has "ext-value" as one its possible enumerated values.
   This particular value serves as an escape sequence to the implementor
   to signal that the extension attribute value should be read.
   Otherwise, this value and has no valid meaning.

   In order to add a new enumerated value to an extensible attribute,
   the value of this attribute MUST be set to "ext-value", and the new
   desired value MUST be set in the corresponding extension attribute.
   For example, an extended instance of the type attribute of the
   SystemImpact class would look as follows:

     <SystemImpact type="ext-value" ext-type="new-attack-type">

   A given extension attribute MUST NOT be set unless the corresponding
   extensible attribute has been set to "ext-value".

5.1.2. **Public Extension of Enumerated Values**

   Select enumerated value of the attributes defined in the data model
   can be extended by adding entries to the corresponding IANA registry.
   Table 1 enumerates these registries. Section 4.3 discusses the XML
   Validation implications of these types of extensions.

5.2. **Extending Classes**

   The classes of the data model can be extended only through the use of
   the AdditionalData and RecordItem classes. These container classes,
   collectively referred to as the extensible classes, are implemented
   with the iodef:ExtensionType data type in the schema. They provide
   the ability to have new atomic or XML-encoded data elements in all of
   the top-level classes of the Incident class and a few of the more
   complicated subordinate classes. As there are multiple instances of
   the extensible classes in the data model, there is discretion on
   where to add a new data element. It is RECOMMENDED that the
   extension be placed in the most closely related class to the new
   information.

   Extensions using the atomic data types (i.e., all values of the dtype
   attributes other than "xml") MUST:

   1. Set the element content of extensible class to the desired value,
       and

   2. Set the dtype attribute to correspond to the data type of the
       element content.

   The following guidelines exist for extensions using XML:

   1. The element content of the extensible class MUST be set to the
       desired value and the dtype attribute MUST be set to "xml".

   2. The extension schema MUST declare a separate namespace. It is
       RECOMMENDED that these extensions have the prefix "iodef-". This
       recommendation makes readability of the document easier by
       allowing the reader to infer which namespaces relate to IODEF by
       inspection.

   3. It is RECOMMENDED that extension schemas follow the naming
       convention of the IODEF data model. This makes reading an
       extended IODEF document look like any other IODEF document. The
       names of all elements are capitalized. For elements with
       composed names, a capital letter is used for each word.
       Attribute names are lower case. Attributes with composed names
       are separated by a hyphen.

4. Parsers that encounter an unrecognized element in a namespace
    that they do support MUST reject the document as a syntax error.

5. There are security and performance implications in requiring
    implementations to dynamically download schemas at run time.
    Thus, implementations SHOULD NOT download schemas at runtime,
    unless implementations take appropriate precautions and are
    prepared for potentially significant network, processing, and
    time-out demands.

6. Some users of the IODEF may have private schema definitions that
    might not be available on the Internet. In this situation, if a
    IODEF document leaks out of the private use space, references to
    some of those document schemas may not be resolvable. This has
    two implications. First, references to private schemas may never
    resolve. As such, in addition to the suggestion that
    implementations do not download schemas at runtime mentioned
    above, recipients MUST be prepared for a schema definition in an
    IODEF document never to resolve.

The following schema and XML document excerpt provide a template for
an extension schema and its use in the IODEF document.

This example schema defines a namespace of "iodef-extension1" and a
single element named "newdata".

```
  <xs:schema
     targetNamespace="iodef-extension1.xsd"
     xmlns:iodef-extension1="iodef-extension1.xsd"
     xmlns:xs="http://www.w3.org/2001/XMLSchema">
     attributeFormDefault="unqualified"
     elementFormDefault="qualified">
   <xs:import
       namespace="urn:ietf:params:xml:ns:iodef-1.0"
       schemaLocation=" urn:ietf:params:xml:schema:iodef-1.0"/>

     <xs:element name="newdata" type="xs:string" />
   </xs:schema>
```

The following XML excerpt demonstrates the use of the above schema as
an extension to the IODEF.

```
    <IODEF-Document
        version="2.00" lang="en-US"
        xmlns="urn:ietf:params:xml:ns:iodef-1.0"
        xmlns:iodef=" urn:ietf:params:xml:ns:iodef-1.0"
        xmlns:iodef-extension1="iodef-extension1.xsd"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="iodef-extension1.xsd">
      <Incident purpose="reporting">
      ...
      <AdditionalData dtype="xml" meaning="xml">
        <iodef-extension1:newdata>
            Field that could not be represented elsewhere
        </iodef-extension1:newdata>
      </AdditionalData>
  </IODEF-Document
```

5.3. Deconflicting Private Extensions

   Private extensions used in a document can be labeled to attribute
   their original specifier using the private-enum-name and private-
   enum-id attributes. This allows a recipient of a document to
   disambiguate private extensions. Only a single private extension can
   be identified in a given IODEF-Document.

   If a CSIRT has only a single private extension, then only the
   private-enum-name attribute needs to be specified. Additional,
   multiple distinct private extensions or versioning of a single
   extension can be accomplished by also setting the corresponding
   private-num-id attribute.

   The following XML excerpt demonstrates the specification of a private
   extension from "example.com" with an identifier of "13".

```
    <IODEF-Document
        version="2.00" lang="en-US"
        private-enum-name="example.com"
        private-enum-id="13"
       ...
  </IODEF-Document>
```

   If an unrecognized private extension is encountered in processing,
   the recipient MAY reject the entire document as a syntax error.

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 and threat information

shared. The IODEF supports this goal by depending on XML constructs,
and through explicit design choices in the data model.

Since IODEF is implemented as an XML Schema, it implicitly supports
all the different character encodings, such as UTF-8 and UTF-16,
possible with XML. Additionally, each IODEF document MUST specify
the language in which their contents are encoded. The language can
be specified with the attribute "xml:lang" (per Section 2.12 of
[W3C.XML]) in the top-level element (i.e., IODEF-Document) and
letting all other elements inherit that definition. All IODEF
classes with a free-form text definition (i.e., all those defined of
type iodef:MLStringType) can also specify a language different from
the rest of the document. The valid language codes for the
"xml:lang" attribute are described in [RFC5646].

The data model supports multiple translations of free-form text. For
classes where free-text is used for descriptive purposes (e.g.,
classes of the iodef:MLStringType type such as the Description
class), the given class always has a one-to-many cardinality to its
parent. The intent is to allow the identical text to be encoded in
different instances of the same class, but each being in a different
language. This approach allows an IODEF document author to send
recipients speaking different languages an identical document. The
IODEF parser SHOULD extract the appropriate language relevant to the
recipient.

Related instances of a given iodef:MLStringType class that are
translations of each other are identified by a common identifier set
in the translation-id attribute. The example below shows three
instances of a Description class expressed in three difference
languages. The relationship between these three instances of the
Description class is conveyed by the common value of "1" in the
translation-id attribute.


```
<IODEF-Document version="2.00" xml:lang="en" ...
  <Incident purpose="reporting">
    ...
    <Description translation-id="1"
               xml:lang="en">English</Description>
    <Description translation-id="1"
               xml:lang="de">Englisch</Description>
    <Description translation-id="1"
               xml:lang="fr">Anglais</Description>
```

While the intent of the data model is to provide internationalization
and localization, the intent is not to do so at the detriment of
interoperability. While the 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 receives even if the text based data elements are
written in a language unfamiliar to the analyst.

7. Examples

This section provides examples of an incident encoded in the IODEF.
These examples do not necessarily represent the only way to encode a
particular incident.

7.1. Worm

An example of a CSIRT reporting an instance of the Code Red worm.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- This example demonstrates a report for a very
     old worm (Code Red) -->
<IODEF-Document version="2.00" lang="en"
  xmlns="urn:ietf:params:xml:ns:iodef-1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:ietf:params:xml:schema:iodef-1.0">
  <Incident purpose="reporting">
    <IncidentID name="csirt.example.com">189493</IncidentID>
    <ReportTime>2001-09-13T23:19:24+00:00</ReportTime>
    <Description>Host sending out Code Red probes</Description>
    <!-- An administrative privilege was attempted, but failed -->
    <Assessment>
      <Impact completion="failed" type="admin"/>
    </Assessment>
    <Contact role="creator" type="organization">
      <ContactName>Example.com CSIRT</ContactName>
      <RegistryHandle registry="arin">example-com</RegistryHandle>
      <Email>contact@csirt.example.com</Email>
    </Contact>
    <EventData>
      <Flow>
        <System category="source">
          <Node>
            <Address category="ipv4-addr">192.0.2.200</Address>
            <Counter type="event">57</Counter>
          </Node>
        </System>
        <System category="target">
          <Node>
            <Address category="ipv4-net">192.0.2.16/28</Address>
```

```
                </Node>
                <Service ip_protocol="6">
                  <Port>80</Port>
                </Service>
              </System>
            </Flow>
            <Expectation action="block-host" />
            <!-- <RecordItem> has an excerpt from a log -->
            <Record>
              <RecordData>
                <DateTime>2001-09-13T18:11:21+02:00</DateTime>
                <Description>Web-server logs</Description>
                <RecordItem dtype="string">
                192.0.2.1 - - [13/Sep/2001:18:11:21 +0200] "GET /default.ida?
                XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                </RecordItem>
                  <!-- Additional logs -->
                <RecordItem dtype="url">
                   http://mylogs.example.com/logs/httpd_access</RecordItem>
              </RecordData>
            </Record>
          </EventData>
          <History>
            <!-- Contact was previously made with the source network
                 owner -->
            <HistoryItem action="contact-source-site">
              <DateTime>2001-09-14T08:19:01+00:00</DateTime>
              <Description>Notification sent to
                         constituency-contact@192.0.2.200</Description>
            </HistoryItem>
          </History>
        </Incident>
      </IODEF-Document>
```

7.2. Reconnaissance

An example of a CSIRT reporting a scanning activity.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- This example describes reconnaissance activity: one-to-one
     and one-to-many scanning -->
<IODEF-Document version="2.00" lang="en"
  xmlns="urn:ietf:params:xml:ns:iodef-1.0"
```

```
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="urn:ietf:params:xml:schema:iodef-1.0">
    <Incident purpose="reporting">
      <IncidentID name="csirt.example.com">59334</IncidentID>
      <ReportTime>2006-08-02T05:54:02-05:00</ReportTime>
      <Assessment>
        <Impact type="recon" completion="succeeded" />
      </Assessment>
      <Method>
        <!-- Reference to the scanning tool "nmap" -->
        <Reference>
          <ReferenceName>nmap</ReferenceName>
          <URL>http://nmap.toolsite.example.com</URL>
        </Reference>
      </Method>
      <!-- Organizational contact and that for staff in that
           organization -->
      <Contact role="creator" type="organization">
        <ContactName>CSIRT for example.com</ContactName>
        <Email>contact@csirt.example.com</Email>
        <Telephone>+1 412 555 12345</Telephone>
        <!-- Since this <Contact> is nested, Joe Smith is part of
             the CSIRT for example.com -->
        <Contact role="tech" type="person" restriction="need-to-know">
          <ContactName>Joe Smith</ContactName>
          <Email>smith@csirt.example.com</Email>
        </Contact>
      </Contact>
      <EventData>
        <!-- Scanning activity as follows:
          192.0.2.1:60524 >> 192.0.2.3:137
                192.0.2.1:60526 >> 192.0.2.3:138
                192.0.2.1:60527 >> 192.0.2.3:139
                192.0.2.1:60531 >> 192.0.2.3:445
        -->
        <Flow>
          <System category="source">
            <Node>
              <Address category="ipv4-addr">192.0.2.200</Address>
            </Node>
            <Service ip_protocol="6">
              <Portlist>60524,60526,60527,60531</Portlist>
            </Service>
          </System>
          <System category="target">
            <Node>
              <Address category="ipv4-addr">192.0.2.201</Address>
            </Node>
```

```
          <Service ip_protocol="6">
            <Portlist>137-139,445</Portlist>
          </Service>
        </System>
      </Flow>
      <!-- Scanning activity as follows:
           192.0.2.2 >> 192.0.2.3/28:445 -->
      <Flow>
        <System category="source">
          <Node>
            <Address category="ipv4-addr">192.0.2.240</Address>
          </Node>
        </System>
        <System category="target">
          <Node>
            <Address category="ipv4-net">192.0.2.64/28</Address>
          </Node>
          <Service ip_protocol="6">
            <Port>445</Port>
          </Service>
        </System>
      </Flow>
    </EventData>
  </Incident>
</IODEF-Document>
```

7.3. Bot-Net Reporting

An example of a CSIRT reporting a bot-network.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- This example describes a compromise and subsequent installation
     of bots -->
<IODEF-Document version="2.00" lang="en"
  xmlns="urn:ietf:params:xml:ns:iodef-1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:ietf:params:xml:schema:iodef-1.0">
  <Incident purpose="mitigation">
    <IncidentID name="csirt.example.com">908711</IncidentID>
    <ReportTime>2006-06-08T05:44:53-05:00</ReportTime>
    <Description>Large bot-net</Description>
    <Assessment>
      <Impact type="dos" severity="high" completion="succeeded" />
    </Assessment>
    <Method>
      <!-- References a given piece of malware, "GT Bot" -->
```

```
      <Reference>
        <ReferenceName>GT Bot</ReferenceName>
      </Reference>
      <!-- References the vulnerability used to compromise the
           machines -->
      <Reference>
        <ReferenceName>CA-2003-22</ReferenceName>
        <URL>http://www.cert.org/advisories/CA-2003-22.html</URL>
        <Description>Root compromise via this IE vulnerability to
                     install the GT Bot</Description>
      </Reference>
    </Method>
    <!-- A member of the CSIRT that is coordinating this
         incident -->
    <Contact type="person" role="irt">
      <ContactName>Joe Smith</ContactName>
      <Email>jsmith@csirt.example.com</Email>
    </Contact>
    <EventData>
      <Description>These hosts are compromised and acting as bots
                   communicating with irc.example.com.</Description>
      <Flow>
        <!-- bot running on 192.0.2.1 and sending DoS traffic at
             10,000 bytes/second -->
        <System category="source">
          <Node>
            <Address category="ipv4-addr">192.0.2.1</Address>
          </Node>
          <Counter type="byte" duration="second">10000</Counter>
          <Description>bot</Description>
        </System>
        <!-- a second bot on 192.0.2.3 -->
        <System category="source">
          <Node>
            <Address category="ipv4-addr">192.0.2.3</Address>
          </Node>
          <Counter type="byte" duration="second">250000</Counter>
          <Description>bot</Description>
        </System>
        <!-- Command-and-control IRC server for these bots-->
        <System category="intermediate">
          <Node>
            <NodeName>irc.example.com</NodeName>
            <Address category="ipv4-addr">192.0.2.20</Address>
            <DateTime>2006-06-08T01:01:03-05:00</DateTime>
          </Node>
          <Description>
             IRC server on #give-me-cmd channel
```

```
              </Description>
            </System>
          </Flow>
          <!-- Request to take these machines offline -->
          <Expectation action="investigate">
            <Description>
              Confirm the source and take machines off-line and
              remediate
            </Description>
          </Expectation>
        </EventData>
      </Incident>
    </IODEF-Document>
```

7.4. Watch List

An example of a CSIRT conveying a watch-list.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- This example demonstrates a trivial IP watch-list -->
<!-- @formatid is set to "watch-list-043" to demonstrate how
     additional semantics about this document could be conveyed
     assuming both parties understood it-->
<IODEF-Document version="2.00" lang="en" formatid="watch-list-043"
  xmlns="urn:ietf:params:xml:ns:iodef-1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:ietf:params:xml:schema:iodef-1.0">
  <Incident purpose="reporting" restriction="private">
    <IncidentID name="csirt.example.com">908711</IncidentID>
    <ReportTime>2006-08-01T00:00:00-05:00</ReportTime>
    <Description>
      Watch-list of known bad IPs or networks
    </Description>
    <Assessment>
      <Impact type="admin" completion="succeeded" />
      <Impact type="recon" completion="succeeded" />
    </Assessment>
    <Contact type="organization" role="creator">
      <ContactName>CSIRT for example.com</ContactName>
      <Email>contact@csirt.example.com</Email>
    </Contact>
    <!-- Separate <EventData> is used to convey
         different <Expectation> -->
    <EventData>
      <Flow>
        <System category="source">
```

```
           <Node>
             <Address category="ipv4-addr">192.0.2.53</Address>
           </Node>
           <Description>Source of numerous attacks</Description>
         </System>
       </Flow>
       <!-- Expectation class indicating that sender of list would
            like to be notified if activity from the host is seen -->
       <Expectation action="contact-sender" />
     </EventData>
     <EventData>
       <Flow>
         <System category="source">
           <Node>
             <Address category="ipv4-net">192.0.2.16/28</Address>
           </Node>
           <Description>
             Source of heavy scanning over past 1-month
           </Description>
         </System>
       </Flow>
       <Flow>
         <System category="source">
           <Node>
             <Address category="ipv4-addr">192.0.2.241</Address>
           </Node>
           <Description>C2 IRC server</Description>
         </System>
       </Flow>
       <!-- Expectation class recommends that these networks
            be filtered -->
       <Expectation action="block-host" />
     </EventData>
   </Incident>
 </IODEF-Document>
```

8. The IODEF Schema

```
<xs:schema targetNamespace="urn:ietf:params:xml:ns:iodef-2.0"
         xmlns="urn:ietf:params:xml:ns:iodef-2.0"
         xmlns:iodef="urn:ietf:params:xml:ns:iodef-2.0"
         xmlns:enum="urn:ietf:params:xml:ns:iodef-enum-1.0"
         xmlns:xs="http://www.w3.org/2001/XMLSchema"
         xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
         elementFormDefault="qualified"
         attributeFormDefault="unqualified">
<xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
```

```
            schemaLocation="http://www.w3.org/TR/2002/
REC-xmldsig-core-20020212/xmldsig-core-schema.xsd"/>
<xs:import namespace="urn:ietf:params:xml:ns:iodef-enum-1.0"
            schemaLocation="http://www.iana.org/assignments/xml-registry/schema/
iodef-enum-1.0.xsd" />
<xs:import namespace="http://www.w3.org/XML/1998/namespace"
            schemaLocation="http://www.w3c.org/2001/xml.xsd" />
  <xs:annotation>
    <xs:documentation>
       Incident Object Description Exchange Format v2.0, RFC5070bis
    </xs:documentation>
  </xs:annotation>

  <!--
   ================================================================
   == IODEF-Document class                                       ==
   ================================================================
  -->
    <xs:element name="IODEF-Document">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="iodef:Incident"
                     maxOccurs="unbounded"/>
          <xs:element ref="iodef:AdditionalData"
                     minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="version"
                     type="xs:string" fixed="2.00"/>
        <xs:attribute ref="xml:lang" />
        <xs:attribute name="format-id"
                     type="xs:string" use="optional"/>
        <xs:attribute name="private-enum-name"
                     type="xs:string" use="optional"/>
        <xs:attribute name="private-enum-id"
                     type="xs:string" use="optional"/>
      </xs:complexType>
    </xs:element>
  <!--
   ================================================================
   ===  Incident class                                         ===
   ================================================================
  -->
    <xs:element name="Incident">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="iodef:IncidentID"/>
          <xs:element ref="iodef:AlternativeID"
                     minOccurs="0"/>
```

```
        <xs:element ref="iodef:RelatedActivity"
```

```
                        minOccurs="0" maxOccurs="unbounded"/>
           <xs:element ref="iodef:DetectTime"
                        minOccurs="0"/>
           <xs:element ref="iodef:StartTime"
                        minOccurs="0"/>
           <xs:element ref="iodef:EndTime"
                        minOccurs="0"/>
           <xs:element ref="iodef:RecoveryTime"
                        minOccurs="0"/>
           <xs:element ref="iodef:ReportTime"/>
           <xs:element ref="iodef:GenerationTime"
                        minOccurs="0"/>
           <xs:element ref="iodef:Description"
                        minOccurs="0" maxOccurs="unbounded"/>
           <xs:element ref="iodef:Discovery"
                        minOccurs="0" maxOccurs="unbounded"/>
           <xs:element ref="iodef:Assessment"
                        maxOccurs="unbounded"/>
           <xs:element ref="iodef:Method"
                        minOccurs="0" maxOccurs="unbounded"/>
           <xs:element ref="iodef:Contact"
                        maxOccurs="unbounded"/>
           <xs:element ref="iodef:EventData"
                        minOccurs="0" maxOccurs="unbounded"/>
           <xs:element ref="iodef:History"
                        minOccurs="0"/>
           <xs:element ref="iodef:AdditionalData"
                        minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="purpose" use="required"
                        type="incident-purpose-type"/>
        <xs:attribute name="ext-purpose"
                        type="xs:string" use="optional"/>
        <xs:attribute name="status" type="incident-status-type"/>
        <xs:attribute name="ext-status"
                        type="xs:string" use="optional"/>
        <xs:attribute ref="xml:lang" />
        <xs:attribute name="restriction"
                         type="iodef:restriction-type"
                         default="private"/>
        <xs:attribute name="ext-restriction"
                        type="xs:string" use="optional"/>
        <xs:attribute name="observable-id"
                         type="xs:ID" use="optional"/>
      </xs:complexType>
    </xs:element>

    <xs:simpleType name="incident-purpose-type">
```

```
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="traceback"/>
        <xs:enumeration value="mitigation"/>
        <xs:enumeration value="reporting"/>
        <xs:enumeration value="watch" />
        <xs:enumeration value="other"/>
        <xs:enumeration value="ext-value"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="incident-status-type">
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="new"/>
        <xs:enumeration value="in-progress"/>
        <xs:enumeration value="forwarded"/>
        <xs:enumeration value="resolved" />
        <xs:enumeration value="future"/>
        <xs:enumeration value="ext-value"/>
      </xs:restriction>
    </xs:simpleType>

  <!--
   ===================================================================
   ==  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:attribute name="instance"
                        type="xs:string" use="optional"/>
          <xs:attribute name="restriction"
                        type="iodef:restriction-type"
                        default="public"/>
          <xs:attribute name="ext-restriction"
                        type="xs:string" use="optional"/>
        </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:attribute name="ext-restriction"
                    type="xs:string" use="optional"/>
    </xs:complexType>
  </xs:element>
<!--
 ====================================================================
 ==  RelatedActivity class                                        ==
 ====================================================================
-->
  <xs:element name="RelatedActivity">
    <xs:complexType>
      <xs:sequence>
        <xs:choice maxOccurs="unbounded">
          <xs:element ref="iodef:IncidentID"
                      maxOccurs="unbounded"/>
          <xs:element ref="iodef:URL"
                      maxOccurs="unbounded"/>
          <xs:element ref="iodef:ThreatActor"
                      maxOccurs="unbounded"/>
          <xs:element ref="iodef:Campaign"
                      maxOccurs="unbounded"/>
        </xs:choice>
        <xs:element ref="iodef:Confidence"
                    minOccurs="0"/>
        <xs:element ref="iodef:Description"
                    minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="iodef:AdditionalData"
                    minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="restriction"
                    type="iodef:restriction-type"/>
      <xs:attribute name="ext-restriction"
                    type="xs:string" use="optional"/>
    </xs:complexType>
  </xs:element>

<!--
 ====================================================================
 ==  ThreatActor class                                            ==
 ====================================================================
```

```
   -->
     <xs:element name="ThreatActor">
       <xs:complexType>
         <xs:sequence>
           <xs:choice>
             <xs:sequence>
               <xs:element ref="iodef:ThreatActorID" />
               <xs:element ref="iodef:Description"
                          minOccurs="0" maxOccurs="unbounded"/>
             </xs:sequence>
             <xs:element ref="iodef:Description"
                        minOccurs="1" maxOccurs="unbounded"/>
           </xs:choice>
           <xs:element ref="iodef:AdditionalData"
                      minOccurs="0" maxOccurs="unbounded"/>
         </xs:sequence>
         <xs:attribute name="restriction"
                      type="iodef:restriction-type"/>
         <xs:attribute name="ext-restriction"
                      type="xs:string" use="optional"/>
       </xs:complexType>
     </xs:element>

     <xs:element name="ThreatActorID" type="xs:string"/>

   <!--
    ================================================================
    ==  Campaign class                                            ==
    ================================================================
   -->
     <xs:element name="Campaign">
       <xs:complexType>
         <xs:sequence>
           <xs:choice>
             <xs:sequence>
               <xs:element ref="iodef:CampaignID"/>
               <xs:element ref="iodef:Description"
                          minOccurs="0" maxOccurs="unbounded"/>
             </xs:sequence>
             <xs:element ref="iodef:Description"
                        minOccurs="1" maxOccurs="unbounded"/>
           </xs:choice>
           <xs:element ref="iodef:AdditionalData"
                      minOccurs="0" maxOccurs="unbounded"/>
         </xs:sequence>
         <xs:attribute name="restriction"
                      type="iodef:restriction-type"/>
         <xs:attribute name="ext-restriction"
```

```
                        type="xs:string" use="optional"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="CampaignID" type="xs:string"/>

  <!--
   ====================================================================
   ==  AdditionalData class                                          ==
   ====================================================================
  -->
    <xs:element name="AdditionalData"
              type="iodef:ExtensionType"/>
  <!--
   ====================================================================
   ==   Contact class                                                ==
   ====================================================================
  -->
    <xs:element name="Contact">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="iodef:ContactName"
                      minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:ContactTitle"
                      minOccurs="0" maxOccurs="unbounded"/>
          <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:element ref="iodef:AdditionalData"
                      minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="role" use="required"
                      type="contact-role-type"/>
        <xs:attribute name="ext-role"
                      type="xs:string" use="optional"/>
        <xs:attribute name="type" use="required"
```

```
                         type="contact-type-type"/>
        <xs:attribute name="ext-type"
                         type="xs:string" use="optional"/>
        <xs:attribute name="restriction"
                         type="iodef:restriction-type"/>
        <xs:attribute name="ext-restriction"
                         type="xs:string" use="optional"/>
    </xs:complexType>
  </xs:element>

  <xs:simpleType name="contact-role-type">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="creator"/>
      <xs:enumeration value="reporter"/>
      <xs:enumeration value="admin"/>
      <xs:enumeration value="tech"/>
      <xs:enumeration value="provider"/>
      <xs:enumeration value="zone"/>
      <xs:enumeration value="user"/>
      <xs:enumeration value="billing"/>
      <xs:enumeration value="legal"/>
      <xs:enumeration value="abuse"/>
      <xs:enumeration value="irt"/>
      <xs:enumeration value="cc"/>
      <xs:enumeration value="cc-irt"/>
      <xs:enumeration value="leo"/>
      <xs:enumeration value="vendor"/>
      <xs:enumeration value="vendor-services"/>
      <xs:enumeration value="victim"/>
      <xs:enumeration value="victim-notified"/>
      <xs:enumeration value="ext-value"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="contact-type-type">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="person"/>
      <xs:enumeration value="organization"/>
      <xs:enumeration value="ext-value"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:element name="ContactName"
              type="iodef:MLStringType"/>
  <xs:element name="ContactTitle"
              type="iodef:MLStringType"/>
  <xs:element name="RegistryHandle">
    <xs:complexType>
```

```
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="registry"
                        type="registryhandle-registry-type"/>
          <xs:attribute name="ext-registry"
                        type="xs:string" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

  <xs:simpleType name="registryhandle-registry-type">
    <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:enumeration value="ext-value"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:element name="PostalAddress">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="iodef:MLStringType">
          <xs:attribute name="meaning"
                        type="xs:string" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="Email" type="iodef:ContactMeansType"/>
  <xs:element name="Telephone" type="iodef:ContactMeansType"/>
  <xs:element name="Fax" type="iodef:ContactMeansType"/>

  <xs:complexType name="ContactMeansType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="meaning"
                      type="xs:string" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

<!--
```

```
      ================================================================
      ==   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="RecoveryTime"
                  type="xs:dateTime"/>
      <xs:element name="GenerationTime"
                  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>
    <!--
      ================================================================
      ==   History class                                            ==
      ================================================================
    -->
      <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:attribute name="ext-restriction"
                        type="xs:string" use="optional"/>
        </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:Contact"
                        minOccurs="0"/>
          <xs:element ref="iodef:Description"
                        minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:DefinedCOA"
                        minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:AdditionalData"
                        minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="restriction"
                        type="iodef:restriction-type"/>
        <xs:attribute name="ext-restriction"
                        type="xs:string" use="optional"/>
        <xs:attribute name="action"
                        type="iodef:action-type" use="required"/>
        <xs:attribute name="ext-action"
                        type="xs:string" use="optional"/>
        <xs:attribute name="observable-id"
                        type="xs:ID" use="optional"/>
     </xs:complexType>
   </xs:element>
   <xs:element name="DefinedCOA" type="iodef:MLStringType"/>

 <!--
  ================================================================
  ==  Expectation class                                        ==
  ================================================================
 -->
   <xs:element name="Expectation">
     <xs:complexType>
       <xs:sequence>
         <xs:element ref="iodef:Description"
                        minOccurs="0" maxOccurs="unbounded"/>
         <xs:element ref="iodef:DefinedCOA"
                        minOccurs="0" 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="ext-restriction"
                        type="xs:string" use="optional"/>
```

```
      <xs:attribute name="severity"
                    type="iodef:severity-type"/>
      <xs:attribute name="action"
                    type="iodef:action-type" default="other"/>
      <xs:attribute name="ext-action"
                    type="xs:string" use="optional"/>
      <xs:attribute name="observable-id"
                    type="xs:ID" use="optional"/>
    </xs:complexType>
  </xs:element>

<!--
  ================================================================
  ==   Discovery class                                        ==
  ================================================================
-->
  <xs:element name="Discovery">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="iodef:Description"
                    minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="iodef:Contact"
                    minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="iodef:DetectionPattern"
                    minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="source"
                    type="discovery-source-type"
                    use="optional" default="unknown"/>
      <xs:attribute name="ext-source"
                    type="xs:string" use="optional"/>
      <xs:attribute name="restriction"
                    type="iodef:restriction-type"/>
      <xs:attribute name="ext-restriction"
                    type="xs:string" use="optional"/>
    </xs:complexType>
  </xs:element>

  <xs:simpleType name="discovery-source-type">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="nidps"/>
      <xs:enumeration value="hips"/>
      <xs:enumeration value="siem"/>
      <xs:enumeration value="av"/>
      <xs:enumeration value="third-party-monitoring"/>
      <xs:enumeration value="incident"/>
      <xs:enumeration value="os-log"/>
      <xs:enumeration value="application-log"/>
```

```
        <xs:enumeration value="device-log"/>
        <xs:enumeration value="network-flow"/>
        <xs:enumeration value="passive-dns"/>
        <xs:enumeration value="investigation"/>
        <xs:enumeration value="audit"/>
        <xs:enumeration value="internal-notification"/>
        <xs:enumeration value="external-notification"/>
        <xs:enumeration value="leo"/>
        <xs:enumeration value="partner"/>
        <xs:enumeration value="actor"/>
        <xs:enumeration value="unknown"/>
        <xs:enumeration value="ext-value"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:element name="DetectionPattern">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="iodef:Application"/>
          <xs:element ref="iodef:Description"
                     minOccurs="0" maxOccurs="unbounded"/>
          <xs:element name="DetectionConfiguration"
                     type="xs:string"
                     minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="restriction"
                     type="iodef:restriction-type"/>
        <xs:attribute name="ext-restriction"
                     type="xs:string" use="optional"/>
      </xs:complexType>
    </xs:element>

  <!--
   ================================================================
   ==  Method class                                              ==
   ================================================================
  -->
    <xs:element name="Method">
      <xs:complexType>
        <xs:sequence>
          <xs:choice maxOccurs="unbounded">
            <xs:element ref="iodef:Reference"/>
            <xs:element ref="iodef:Description"/>
          </xs:choice>
          <xs:element ref="iodef:AdditionalData"
                     minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="restriction"
```

```
                          type="iodef:restriction-type"/>
             <xs:attribute name="ext-restriction"
                          type="xs:string" use="optional"/>
         </xs:complexType>
       </xs:element>

     <!--
       ==================================================================
       ==   Reference class                                          ==
       ==================================================================
     -->
       <xs:element name="Reference">
         <xs:complexType>
           <xs:sequence>
             <xs:element ref="enum:ReferenceName"
                          minOccurs="0" />
             <xs:element ref="iodef:URL"
                          minOccurs="0" maxOccurs="unbounded"/>
             <xs:element ref="iodef:Description"
                          minOccurs="0" maxOccurs="unbounded"/>
           </xs:sequence>
           <xs:attribute name="observable-id"
                          type="xs:ID" use="optional"/>
         </xs:complexType>
       </xs:element>

     <!--
       ==================================================================
       ==   Assessment class                                         ==
       ==================================================================
     -->
       <xs:element name="Assessment">
         <xs:complexType>
           <xs:sequence>
             <xs:element ref="iodef:IncidentCategory"
                          minOccurs="0" maxOccurs="unbounded"/>
             <xs:choice maxOccurs="unbounded">
               <xs:element ref="iodef:SystemImpact"/>
               <xs:element ref="iodef:BusinessImpact"/>
               <xs:element ref="iodef:TimeImpact"/>
               <xs:element ref="iodef:MonetaryImpact"/>
               <xs:element ref="iodef:IntendedImpact"/>
             </xs:choice>
             <xs:element ref="iodef:Counter"
                          minOccurs="0" maxOccurs="unbounded"/>
             <xs:element ref="MitigatingFactor"
                          minOccurs="0" maxOccurs="unbounded"/>
             <xs:element ref="iodef:Confidence" minOccurs="0"/>
```

```
            <xs:element ref="iodef:AdditionalData"
                    minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="occurrence">
          <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="actual"/>
              <xs:enumeration value="potential"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="restriction"
                    type="iodef:restriction-type"/>
        <xs:attribute name="ext-restriction"
                    type="xs:string" use="optional"/>
        <xs:attribute name="observable-id"
                    type="xs:ID" use="optional"/>
      </xs:complexType>
    </xs:element>

    <xs:element name="IncidentCategory" type="iodef:MLStringType"/>
    <xs:element name="BusinessImpact"
                type="iodef:BusinessImpactType"/>
    <xs:element name="IntendedImpact"
                type="iodef:BusinessImpactType"/>
    <xs:element name="MitigatingFactor"
                type="iodef:MLStringType"/>

    <xs:element name="SystemImpact">
      <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"
                        type="systemimpact-type-type"
                        use="optional"/>
            <xs:attribute name="ext-type"
                        type="xs:string" use="optional"/>
          </xs:extension>
```

```
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>

      <xs:simpleType name="systemimpact-type-type">
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="admin"/>
          <xs:enumeration value="takeover-account"/>
          <xs:enumeration value="takeover-service"/>
          <xs:enumeration value="takeover-system"/>
          <xs:enumeration value="cps-manipulation"/>
          <xs:enumeration value="cps-damage"/>
          <xs:enumeration value="availability-data"/>
          <xs:enumeration value="availibility-account"/>
          <xs:enumeration value="availibility-service"/>
          <xs:enumeration value="availibility-system"/>
          <xs:enumeration value="damaged-system"/>
          <xs:enumeration value="damaged-data"/>
          <xs:enumeration value="breach-proprietary"/>
          <xs:enumeration value="breach-privacy"/>
          <xs:enumeration value="breach-credential"/>
          <xs:enumeration value="breach-configuration"/>
          <xs:enumeration value="integrity-data"/>
          <xs:enumeration value="integrity-configuration"/>
          <xs:enumeration value="integrity-hardware"/>
          <xs:enumeration value="traffic-redirection"/>
          <xs:enumeration value="monitoring-traffic"/>
          <xs:enumeration value="monitoring-host"/>
          <xs:enumeration value="policy"/>
          <xs:enumeration value="ext-value"/>
        </xs:restriction>
      </xs:simpleType>

      <xs:complexType name="BusinessImpactType">
        <xs:simpleContent>
          <xs:extension base="iodef:MLStringType">
            <xs:attribute name="severity"
                          type="businessimpact-severity-type"
                          use="optional"/>
            <xs:attribute name="ext-severity"
                          type="xs:string" use="optional"/>
            <xs:attribute name="type"
                          type="businessimpact-type-type"
                          use="optional"/>
            <xs:attribute name="ext-type"
                          type="xs:string" use="optional"/>
          </xs:extension>
        </xs:simpleContent>
```

```
    </xs:complexType>

    <xs:simpleType name="businessimpact-severity-type">
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="none"/>
        <xs:enumeration value="low"/>
        <xs:enumeration value="medium"/>
        <xs:enumeration value="high"/>
        <xs:enumeration value="unknown"/>
        <xs:enumeration value="ext-value"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="businessimpact-type-type">
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="breach-proprietary"/>
        <xs:enumeration value="breach-privacy"/>
        <xs:enumeration value="breach-credential"/>
        <xs:enumeration value="loss-of-integrity"/>
        <xs:enumeration value="loss-of-service" />
        <xs:enumeration value="theft-financial"/>
        <xs:enumeration value="theft-service"/>
        <xs:enumeration value="degraded-reputation"/>
        <xs:enumeration value="asset-damage"/>
        <xs:enumeration value="asset-manipulation"/>
        <xs:enumeration value="legal"/>
        <xs:enumeration value="extortion"/>
        <xs:enumeration value="ext-value"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:element name="TimeImpact">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="iodef:PositiveFloatType">
            <xs:attribute name="severity"
                          type="iodef:severity-type"/>
            <xs:attribute name="metric" use="required"
                          type="timeimpact-metric-type"/>
            <xs:attribute name="duration"
                          type="iodef:duration-type"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>

    <xs:simpleType name="timeimpact-metric-type">
      <xs:restriction base="xs:NMTOKEN">
```

```
            <xs:enumeration value="labor"/>
            <xs:enumeration value="elapsed"/>
            <xs:enumeration value="downtime"/>
            <xs:enumeration value="ext-value"/>
          </xs:restriction>
      </xs:simpleType>

      <xs:element name="MonetaryImpact">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="iodef:PositiveFloatType">
              <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"
                        type="confidence-rating-type"/>
        </xs:complexType>
      </xs:element>

      <xs:simpleType name="confidence-rating-type">
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="low"/>
          <xs:enumeration value="medium"/>
          <xs:enumeration value="high"/>
          <xs:enumeration value="numeric"/>
          <xs:enumeration value="unknown"/>
        </xs:restriction>
      </xs:simpleType>

  <!--
    ================================================================
    == EventData class                                            ==
    ================================================================
  -->
      <xs:element name="EventData">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="iodef:Description"
                        minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="iodef:DetectTime"
```

```
                         minOccurs="0"/>
              <xs:element ref="iodef:StartTime"
                         minOccurs="0"/>
              <xs:element ref="iodef:EndTime"
                         minOccurs="0"/>
              <xs:element ref="iodef:RecoveryTime"
                         minOccurs="0"/>
              <xs:element ref="iodef:ReportTime"
                         minOccurs="0"/>
              <xs:element ref="iodef:Contact"
                         minOccurs="0" maxOccurs="unbounded"/>
              <xs:element ref="iodef:Discovery"
                         minOccurs="0" maxOccurs="unbounded"/>
              <xs:element ref="iodef:Assessment"
                         minOccurs="0"/>
              <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"/>
              <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:attribute name="ext-restriction"
                         type="xs:string" use="optional"/>
            <xs:attribute name="observable-id"
                         type="xs:ID" use="optional"/>
          </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>
      <!--
        ================================================================
        ==  System class                                              ==
        ================================================================
      -->
        <xs:element name="System">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="iodef:Node" maxOccurs="unbounded"/>
              <xs:element ref="iodef:NodeRole"
                          minOccurs="0" maxOccurs="unbounded" />
              <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:element name="AssetID" type="xs:string"
                          minOccurs="0" maxOccurs="unbounded"/>
              <xs:element ref="iodef:Description"
                          minOccurs="0" maxOccurs="unbounded"/>
              <xs:element ref="iodef:AdditionalData"
                          minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="restriction"
                          type="iodef:restriction-type"/>
            <xs:attribute name="ext-restriction"
                          type="xs:string" use="optional"/>
            <xs:attribute name="category" type="system-category-type"/>
            <xs:attribute name="ext-category"
                          type="xs:string" use="optional"/>
            <xs:attribute name="interface"
                          type="xs:string"/>
            <xs:attribute name="spoofed" type="yes-no-unknown-type"
                          default="unknown" />
            <xs:attribute name="virtual" type="yes-no-unknown-type"
                          use="optional" default="unknown"/>
            <xs:attribute name="ownership" type="system-ownership-type"
                          use="optional" />
            <xs:attribute name="ext-ownership"
                          type="xs:string" use="optional"/>
          </xs:complexType>
        </xs:element>

        <xs:simpleType name="system-category-type">
          <xs:restriction base="xs:NMTOKEN">
```

```
        <xs:enumeration value="source"/>
        <xs:enumeration value="target"/>
        <xs:enumeration value="intermediate"/>
        <xs:enumeration value="sensor"/>
        <xs:enumeration value="infrastructure"/>
        <xs:enumeration value="ext-value"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="system-ownership-type">
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="organization"/>
        <xs:enumeration value="personal"/>
        <xs:enumeration value="partner"/>
        <xs:enumeration value="customer"/>
        <xs:enumeration value="no-relationship"/>
        <xs:enumeration value="unknown"/>
        <xs:enumeration value="ext-value"/>
      </xs:restriction>
    </xs:simpleType>

  <!--
   =================================================================
   == Node class                                                  ==
   =================================================================
  -->
    <xs:element name="Node">
      <xs:complexType>
        <xs:sequence>
          <xs:choice maxOccurs="unbounded">
            <xs:element ref="iodef:DomainData" minOccurs="0"
                       maxOccurs="unbounded"/>
            <xs:element ref="iodef:Address"
                       minOccurs="0" maxOccurs="unbounded"/>
          </xs:choice>
          <xs:element ref="iodef:PostalAddress"
                     minOccurs="0"/>
          <xs:element ref="iodef:Location"
                     minOccurs="0" maxOccurs="unbounded"/>
          <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"
                        type="address-category-type"
                        default="ipv4-addr"/>
          <xs:attribute name="ext-category"
                        type="xs:string" use="optional"/>
          <xs:attribute name="vlan-name"
                        type="xs:string"/>
          <xs:attribute name="vlan-num"
                        type="xs:integer"/>
          <xs:attribute name="observable-id"
                        type="xs:ID" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

  <xs:simpleType name="address-category-type">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="asn"/>
      <xs:enumeration value="atm"/>
      <xs:enumeration value="e-mail"/>
      <xs:enumeration value="mac"/>
      <xs:enumeration value="ipv4-addr"/>
      <xs:enumeration value="ipv4-net"/>
      <xs:enumeration value="ipv4-net-mask"/>
      <xs:enumeration value="ipv6-addr"/>
      <xs:enumeration value="ipv6-net"/>
      <xs:enumeration value="ipv6-net-mask"/>
      <xs:enumeration value="site-uri"/>
      <xs:enumeration value="ext-value"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:element name="Location" type="iodef:MLStringType"/>

  <xs:element name="NodeRole">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="iodef:MLStringType">
          <xs:attribute name="category"
                        type="noderole-category-type"
                        use="required"/>
          <xs:attribute name="ext-category"
                        type="xs:string" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
```

```
        </xs:complexType>
      </xs:element>

      <xs:simpleType name="noderole-category-type">
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="client"/>
          <xs:enumeration value="client-enterprise"/>
          <xs:enumeration value="client-partner"/>
          <xs:enumeration value="client-remote"/>
          <xs:enumeration value="client-kiosk"/>
          <xs:enumeration value="client-mobile"/>
          <xs:enumeration value="server-internal"/>
          <xs:enumeration value="server-public"/>
          <xs:enumeration value="www"/>
          <xs:enumeration value="mail"/>
          <xs:enumeration value="webmail" />
          <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="backup"/>
          <xs:enumeration value="dhcp"/>
          <xs:enumeration value="assessment"/>
          <xs:enumeration value="source-control"/>
          <xs:enumeration value="config-management"/>
          <xs:enumeration value="monitoring"/>
          <xs:enumeration value="infra"/>
          <xs:enumeration value="infra-firewall"/>
          <xs:enumeration value="infra-router"/>
          <xs:enumeration value="infra-switch"/>
          <xs:enumeration value="camera"/>
          <xs:enumeration value="proxy"/>
          <xs:enumeration value="remote-access"/>
          <xs:enumeration value="log"/>
          <xs:enumeration value="virtualization"/>
          <xs:enumeration value="pos"/>
          <xs:enumeration value="scada"/>
          <xs:enumeration value="scada-supervisory"/>
          <xs:enumeration value="sinkhole"/>
          <xs:enumeration value="honeypot"/>
```

```
          <xs:enumeration value="anonymization"/>
          <xs:enumeration value="c2-server"/>
          <xs:enumeration value="malware-distribution"/>
          <xs:enumeration value="drop-server"/>
          <xs:enumeration value="hop-point"/>
          <xs:enumeration value="reflector"/>
          <xs:enumeration value="phishing-site"/>
          <xs:enumeration value="spear-phishing-site"/>
          <xs:enumeration value="recruiting-site"/>
          <xs:enumeration value="fraudulent-site"/>
          <xs:enumeration value="ext-value"/>
        </xs:restriction>
      </xs:simpleType>

   <!--
     ================================================================
     ==   Service Class                                            ==
     ================================================================
   -->
      <xs:element name="Service">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="iodef:ServiceName"
                             minOccurs="0"/>
            <xs:choice minOccurs="0">
              <xs:element ref="iodef:Port"/>
              <xs:element ref="iodef:Portlist"/>
            </xs:choice>
            <xs:element ref="iodef:ProtoType" minOccurs="0"/>
            <xs:element ref="iodef:ProtoCode" minOccurs="0"/>
            <xs:element ref="iodef:ProtoField" minOccurs="0"/>
            <xs:element ref="iodef:ApplicationHeader"
                        minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="iodef:EmailData" minOccurs="0"/>
            <xs:element ref="iodef:Application"
                        minOccurs="0"/>
          </xs:sequence>
          <xs:attribute name="ip-protocol"
                        type="xs:integer" use="required"/>
          <xs:attribute name="observable-id"
                        type="xs:ID" use="optional"/>
        </xs:complexType>
      </xs:element>

      <xs:element name="Port" type="xs:integer"/>
      <xs:element name="Portlist" type="iodef:PortlistType"/>
      <xs:element name="ProtoType" type="xs:integer"/>
      <xs:element name="ProtoCode" type="xs:integer"/>
```

```
   <xs:element name="ProtoField" type="xs:integer"/>
   <xs:element name="ApplicationHeader"
               type="iodef:ApplicationHeaderType"/>

   <xs:element name="ServiceName">
     <xs:complexType>
       <xs:sequence>
         <xs:element ref="iodef:IANAService"/>
         <xs:element ref="iodef:URL"
                        minOccurs="0" maxOccurs="unbounded"/>
         <xs:element ref="iodef:Description"
                        minOccurs="0" maxOccurs="unbounded"/>
       </xs:sequence>
     </xs:complexType>
   </xs:element>

   <xs:element name="IANAService" type="xs:string"/>

 <!--
  ================================================================
  ==  Counter class                                            ==
  ================================================================
 -->
   <xs:element name="Counter">
     <xs:complexType>
       <xs:simpleContent>
         <xs:extension base="xs:double">
           <xs:attribute name="type"
                        type="counter-type-type"
                        use="required"/>
           <xs:attribute name="unit"
                        type="counter-unit-type"
                        use="required"/>
           <xs:attribute name="ext-type"
                        type="xs:string" use="optional"/>
           <xs:attribute name="meaning"
                        type="xs:string" use="optional"/>
           <xs:attribute name="duration"
                        type="iodef:duration-type"/>
           <xs:attribute name="ext-duration"
                        type="xs:string" use="optional"/>
         </xs:extension>
       </xs:simpleContent>
     </xs:complexType>
   </xs:element>

   <xs:simpleType name="counter-type-type">
     <xs:restriction base="xs:NMTOKEN">
```

```
        <xs:enumeration value="counter"/>
        <xs:enumeration value="rate"/>
        <xs:enumeration value="average"/>
        <xs:enumeration value="ext-value"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="counter-unit-type">
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="byte"/>
        <xs:enumeration value="mbit"/>
        <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="host"/>
        <xs:enumeration value="site"/>
        <xs:enumeration value="organization"/>
        <xs:enumeration value="ext-value"/>
      </xs:restriction>
    </xs:simpleType>

  <!--
   ================================================================
   ==   EmailData class                                        ==
   ================================================================
  -->
  <xs:element name="EmailData">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="iodef:EmailFrom" minOccurs="0"/>
        <xs:element ref="iodef:EmailSubject" minOccurs="0"/>
        <xs:element ref="iodef:EmailX-Mailer" minOccurs="0"/>
        <xs:element ref="iodef:EmailHeaderField" minOccurs="0"/>
        <xs:element ref="iodef:HashData"
                   minOccurs="0" />
        <xs:element ref="SignatureData"
                   minOccurs="0" />
      </xs:sequence>
      <xs:attribute name="observable-id"
                   type="xs:ID" use="optional"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="EmailFrom" type="iodef:MLStringType"/>
  <xs:element name="EmailSubject" type="iodef:MLStringType"/>
```

```
    <xs:element name="EmailX-Mailer" type="iodef:MLStringType"/>
    <xs:element name="EmailHeaderField"
               type="iodef:ApplicationHeaderType"/>

  <!--
   ================================================================
   ==    DomainData class - from RFC5901                        ==
   ================================================================
  -->
  <xs:element name="DomainData">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="iodef:Name" maxOccurs="1" />
        <xs:element ref="iodef:DateDomainWasChecked"
                   minOccurs="0" maxOccurs="1" />
        <xs:element ref="iodef:RegistrationDate"
                   minOccurs="0" maxOccurs="1" />
        <xs:element ref="iodef:ExpirationDate"
                   minOccurs="0" maxOccurs="1" />
        <xs:element ref="iodef:RelatedDNS"
                   minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="iodef:Nameservers"
                   minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="iodef:DomainContacts"
                   minOccurs="0" maxOccurs="1" />
      </xs:sequence>
      <xs:attribute name="system-status"
                   type="domaindata-system-status-type"/>
      <xs:attribute name="ext-system-status"
                   type="xs:string" use="optional"/>
      <xs:attribute name="domain-status"
                   type="domaindata-domain-status-type"/>
      <xs:attribute name="ext-domain-status"
                   type="xs:string" use="optional"/>
      <xs:attribute name="observable-id"
                   type="xs:ID" use="optional"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="Name" type="xs:string"/>
  <xs:element name="DateDomainWasChecked" type="xs:dateTime"/>
  <xs:element name="RegistrationDate" type="xs:dateTime"/>
  <xs:element name="ExpirationDate" type="xs:dateTime"/>

  <xs:simpleType name="domaindata-system-status-type">
    <xs:restriction base="xs:string">
      <xs:enumeration value="spoofed"/>
      <xs:enumeration value="fraudulent"/>
```

```
      <xs:enumeration value="innocent-hacked"/>
      <xs:enumeration value="innocent-hijacked"/>
      <xs:enumeration value="unknown"/>
      <xs:enumeration value="ext-value"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="domaindata-domain-status-type">
    <xs:restriction base="xs:string">
      <xs:enumeration value="reservedDelegation"/>
      <xs:enumeration value="assignedAndActive"/>
      <xs:enumeration value="assignedAndInactive"/>
      <xs:enumeration value="assignedAndOnHold"/>
      <xs:enumeration value="revoked"/>
      <xs:enumeration value="transferPending"/>
      <xs:enumeration value="registryLock"/>
      <xs:enumeration value="registrarLock"/>
      <xs:enumeration value="other"/>
      <xs:enumeration value="unknown"/>
      <xs:enumeration value="ext-value"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:element name="RelatedDNS">
    <xs:complexType>
     <xs:simpleContent>
      <xs:extension base="xs:string">
      <xs:attribute name="record-type" use="optional">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="A"/>
            <xs:enumeration value="AAAA"/>
            <xs:enumeration value="AFSDB"/>
            <xs:enumeration value="APL"/>
            <xs:enumeration value="AXFR"/>
            <xs:enumeration value="CAA"/>
            <xs:enumeration value="CERT"/>
            <xs:enumeration value="CNAME"/>
            <xs:enumeration value="DHCID"/>
            <xs:enumeration value="DLV"/>
            <xs:enumeration value="DNAME"/>
            <xs:enumeration value="DNSKEY"/>
            <xs:enumeration value="DS"/>
            <xs:enumeration value="HIP"/>
            <xs:enumeration value="IXFR"/>
            <xs:enumeration value="IPSECKEY"/>
            <xs:enumeration value="LOC"/>
            <xs:enumeration value="MX"/>
```

```
            <xs:enumeration value="NAPTR"/>
            <xs:enumeration value="NS"/>
            <xs:enumeration value="NSEC"/>
            <xs:enumeration value="NSEC3"/>
            <xs:enumeration value="NSEC3PARAM"/>
            <xs:enumeration value="OPT"/>
            <xs:enumeration value="PTR"/>
            <xs:enumeration value="RRSIG"/>
            <xs:enumeration value="RP"/>
            <xs:enumeration value="SIG"/>
            <xs:enumeration value="SOA"/>
            <xs:enumeration value="SPF"/>
            <xs:enumeration value="SRV"/>
            <xs:enumeration value="SSHFP"/>
            <xs:enumeration value="TA"/>
            <xs:enumeration value="TKEY"/>
            <xs:enumeration value="TLSA"/>
            <xs:enumeration value="TSIG"/>
            <xs:enumeration value="TXT"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
     </xs:extension>
    </xs:simpleContent>
   </xs:complexType>
  </xs:element>

  <xs:element name="Nameservers">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="iodef:Server"/>
        <xs:element ref="iodef:Address" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="Server" type="xs:string"/>

  <xs:element name="DomainContacts">
    <xs:complexType>
       <xs:choice>
         <xs:element ref="iodef:SameDomainContact"/>
         <xs:element ref="iodef:Contact"
                    maxOccurs="unbounded" minOccurs="1"/>
       </xs:choice>
    </xs:complexType>
  </xs:element>
```

```
  <xs:element name="SameDomainContact" type="xs:string"/>

<!--
 ===================================================================
 ==  Record class                                               ==
 ===================================================================
-->
  <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:attribute name="ext-restriction"
                    type="xs:string" use="optional"/>
    </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:element ref="iodef:FileData"
                    minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="iodef:WindowsRegistryKeysModified"
                    minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="iodef:CertificateData"
                    minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="iodef:AdditionalData"
                    minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="restriction"
                    type="iodef:restriction-type"/>
      <xs:attribute name="ext-restriction"
                    type="xs:string" use="optional"/>
      <xs:attribute name="observable-id"
                    type="xs:ID" use="optional"/>
    </xs:complexType>
```

```
      </xs:element>

      <xs:element name="RecordPattern">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="type"
                            type="recordpattern-type-type"
                            use="required"/>
              <xs:attribute name="ext-type"
                            type="xs:string" use="optional"/>
              <xs:attribute name="offset"
                            type="xs:integer" use="optional"/>
              <xs:attribute name="offsetunit"
                            type="recordpattern-offsetunit-type"
                            use="optional" default="line"/>
              <xs:attribute name="ext-offsetunit"
                            type="xs:string" use="optional"/>
              <xs:attribute name="instance"
                            type="xs:integer" use="optional"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>

      <xs:simpleType name="recordpattern-type-type">
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="regex"/>
          <xs:enumeration value="binary"/>
          <xs:enumeration value="xpath"/>
          <xs:enumeration value="ext-value"/>
        </xs:restriction>
      </xs:simpleType>

      <xs:simpleType name="recordpattern-offsetunit-type">
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="line"/>
          <xs:enumeration value="byte"/>
          <xs:enumeration value="ext-value"/>
        </xs:restriction>
      </xs:simpleType>

      <xs:element name="RecordItem"
                  type="iodef:ExtensionType"/>
<!--
  ================================================================
  ==  Class to describe Windows Registry Keys                  ==
  ================================================================
```

```
-->
      <xs:element name="WindowsRegistryKeysModified">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="iodef:Key"
                        maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="observable-id"
                        type="xs:ID" use="optional"/>
        </xs:complexType>
      </xs:element>

      <xs:element name="Key">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="iodef:KeyName"/>
            <xs:element ref="iodef:Value" minOccurs="0"/>
          </xs:sequence>
          <xs:attribute name="registryaction"
                        type="key-registryaction-type"/>
          <xs:attribute name="ext-registryaction"
                        type="xs:string" use="optional"/>
        </xs:complexType>
      </xs:element>

      <xs:element name="KeyName" type="xs:string"/>
      <xs:element name="Value" type="xs:string"/>

      <xs:simpleType name="key-registryaction-type">
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="add-key"/>
          <xs:enumeration value="add-value"/>
          <xs:enumeration value="delete-key"/>
          <xs:enumeration value="delete-value"/>
          <xs:enumeration value="modify-key"/>
          <xs:enumeration value="modify-value"/>
          <xs:enumeration value="ext-value"/>
        </xs:restriction>
      </xs:simpleType>

<!--
=================================================================
==  Classes to describe a file                                 ==
=================================================================
-->

   <xs:element name="FileData">
     <xs:complexType>
```

```
         <xs:sequence>
           <xs:element ref="iodef:File"
                       minOccurs="1" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="observable-id"
                        type="xs:ID" use="optional"/>
          <xs:attribute name="restriction"
                        type="iodef:restriction-type"/>
          <xs:attribute name="ext-restriction"
                        type="xs:string" use="optional"/>
         </xs:complexType>
    </xs:element>

    <xs:element name="File">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="iodef:FileName" minOccurs="0" />
          <xs:element ref="iodef:FileSize" minOccurs="0" />
          <xs:element ref="FileType" minOccurs="0" />
          <xs:element ref="iodef:URL"
                      minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:HashData"
                      minOccurs="0" />
          <xs:element ref="ds:Signature"
                      minOccurs="0" />
          <xs:element name="Application"
                      type="iodef:SoftwareType" minOccurs="0"/>
          <xs:element ref="iodef:FileProperties"
                      minOccurs="0" maxOccurs="unbounded"/>
         </xs:sequence>
         <xs:attribute name="observable-id"
                       type="xs:ID" use="optional"/>
         </xs:complexType>
    </xs:element>

    <xs:element name="FileName" type="xs:string"/>
    <xs:element name="FileSize" type="xs:integer"/>
    <xs:element name="FileType" type="xs:integer"/>
    <xs:element name="FileProperties" type="iodef:ExtensionType"/>

<!--
=============================================================
==  Classes to describe a hash                             ==
=============================================================
-->

  <xs:element name="HashData">
  <xs:complexType>
```

```
   <xs:sequence>
      <xs:element ref="iodef:HashTarget" minOccurs="0"/>
      <xs:element ref="iodef:Hash"
                  minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:FuzzyHash"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="scope"
                  type="hashdata-scope-type"
                  use="required"/>
    <xs:attribute name="ext-scope"
                  type="xs:string" use="optional"/>
   </xs:complexType>
  </xs:element>

  <xs:element name="HashTarget" type="iodef:MLStringType"/>

  <xs:simpleType name="hashdata-scope-type">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="file-contents"/>
      <xs:enumeration value="file-pe-section"/>
      <xs:enumeration value="file-pe-iat"/>
      <xs:enumeration value="file-pe-resource"/>
      <xs:enumeration value="file-pdf-object"/>
      <xs:enumeration value="email-hash"/>
      <xs:enumeration value="email-headers-hash"/>
      <xs:enumeration value="email-body-hash"/>
      <xs:enumeration value="ext-value"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:element name="Hash">
   <xs:complexType>
    <xs:sequence>
      <xs:element ref="ds:DigestMethod" />
      <xs:element ref="ds:DigestValue" />
      <xs:element ref="ds:CanonicalizationMethod" />
      <xs:element ref="iodef:Application"
                  minOccurs="0"/>
    </xs:sequence>
   </xs:complexType>
  </xs:element>

  <xs:element name="FuzzyHash">
   <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:AdditionalData" />
      <xs:element ref="iodef:Application"
```

```
                        minOccurs="0"/>
        </xs:sequence>
     </xs:complexType>
    </xs:element>

<!--
   ==============================================================
   ==  Classes to describe a signature                        ==
   ==============================================================
-->

   <xs:element name="SignatureData">
    <xs:complexType>
     <xs:sequence>
         <xs:element ref="ds:Signature"
                     maxOccurs="unbounded" />
       </xs:sequence>
     </xs:complexType>
    </xs:element>

<!--
   ==============================================================
   ==  Classes to describe a certficate                       ==
   ==============================================================
-->

   <xs:element name="CertificateData">
     <xs:complexType>
       <xs:sequence>
         <xs:element ref="iodef:Certificate"
                     maxOccurs="unbounded"/>
       </xs:sequence>
       <xs:attribute name="observable-id"
                     type="xs:ID" use="optional"/>
       <xs:attribute name="restriction"
                     type="iodef:restriction-type"/>
       <xs:attribute name="ext-restriction"
                     type="xs:string" use="optional"/>
     </xs:complexType>
   </xs:element>

   <xs:element name="Certificate">
     <xs:complexType>
       <xs:sequence>
         <xs:element ref="ds:X509Data" />
       </xs:sequence>
      <xs:attribute name="observable-id"
                    type="xs:ID" use="optional"/>
```

```
      </xs:complexType>
    </xs:element>

    <!--
      ================================================================
      ==  Classes that describe software                          ==
      ================================================================
    -->
      <xs:complexType name="SoftwareType">
        <xs:sequence>
          <xs:element ref="iodef:SoftwareReference"
                      minOccurs="0"/>
          <xs:element ref="iodef:URL"
                      minOccurs="0" maxOccurs="unbounded" />
          <xs:element ref="iodef:Description"
                      minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
      </xs:complexType>

      <xs:element name="SoftwareReference">
        <xs:complexType>
          <xs:sequence>
            <xs:any namespace="##any" processContents="lax"
                    minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="spec-name"
                        type="softwarereference-spec-name-type"
                        use="required"/>
          <xs:attribute name="ext-spec-name"
                        type="xs:string" use="optional"/>
          <xs:attribute name="dtype"
                        type="softwarereference-dtype-type"
                        use="optional"/>
          <xs:attribute name="ext-dtype"
                        type="xs:string" use="optional"/>
        </xs:complexType>
      </xs:element>

      <xs:simpleType name="softwarereference-spec-name-type">
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="custom"/>
          <xs:enumeration value="cpe"/>
          <xs:enumeration value="swid"/>
          <xs:enumeration value="ext-value"/>
        </xs:restriction>
      </xs:simpleType>

      <xs:simpleType name="softwarereference-dtype-type">
```

```
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="bytes"/>
          <xs:enumeration value="integer"/>
          <xs:enumeration value="real"/>
          <xs:enumeration value="string"/>
          <xs:enumeration value="xml"/>
          <xs:enumeration value="ext-value"/>
        </xs:restriction>
      </xs:simpleType>

      <xs:element name="Application" type="iodef:SoftwareType"/>
      <xs:element name="OperatingSystem" type="iodef:SoftwareType"/>

    <!--
      ================================================================
      == IndicatorData classes                                    ==
      ================================================================
    -->
      <xs:element name="IndicatorData">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="iodef:Indicator"
                        minOccurs="1" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>

      <xs:element name="Indicator">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="iodef:IndicatorID" />
            <xs:element ref="iodef:AlternativeIndicatorID"
                        minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="iodef:Description"
                        minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="iodef:StartTime"
                        minOccurs="0" />
            <xs:element ref="iodef:EndTime"
                        minOccurs="0" />
            <xs:element ref="iodef:Confidence"
                        minOccurs="0" />
            <xs:element ref="iodef:Contact"
                        minOccurs="0" maxOccurs="unbounded"/>
            <xs:choice>
              <xs:element ref="iodef:Observable" />
              <xs:element ref="iodef:ObservableReference" />
              <xs:element ref="iodef:IndicatorExpression" />
              <xs:element ref="iodef:IndicatorReference" />
```

```
          </xs:choice>
        </xs:sequence>
        <xs:attribute name="restriction"
                      type="iodef:restriction-type"/>
        <xs:attribute name="ext-restriction"
                      type="xs:string" use="optional"/>
      </xs:complexType>
    </xs:element>

    <xs:element name="IndicatorID">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:ID">
            <xs:attribute name="name"
                          type="xs:string" use="required"/>
            <xs:attribute name="version"
                          type="xs:string" use="required"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>

    <xs:element name="AlternativeIndicatorID">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="iodef:IndicatorID"
                      maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="restriction"
                      type="iodef:restriction-type"/>
        <xs:attribute name="ext-restriction"
                      type="xs:string" use="optional"/>
      </xs:complexType>
    </xs:element>

    <xs:element name="Observable">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="iodef:Address"
                      minOccurs="0"/>
          <xs:element ref="iodef:DomainData"
                      minOccurs="0"/>
          <xs:element ref="iodef:EmailData"
                      minOccurs="0"/>
          <xs:element name="ApplicationHeader"
                      type="iodef:ApplicationHeaderType"
                      minOccurs="0"/>
          <xs:element ref="iodef:WindowsRegistryKeysModified"
```

```
                         minOccurs="0"/>
          <xs:element ref="iodef:FileData"
                         minOccurs="0"/>
          <xs:element ref="iodef:CertificateData"
                         minOccurs="0"/>
          <xs:element ref="iodef:RegistryHandle"
                         minOccurs="0"/>
          <xs:element ref="iodef:RecordData"
                         minOccurs="0"/>
          <xs:element ref="iodef:EventData"
                         minOccurs="0"/>
          <xs:element ref="iodef:Incident"
                         minOccurs="0"/>
          <xs:element ref="iodef:Expectation"
                         minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="Reference"
                         minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:Assessment"
                         minOccurs="0"/>
          <xs:element ref="iodef:HistoryItem"
                         minOccurs="0"/>
          <xs:element ref="iodef:BulkObservable"
                         minOccurs="0"/>
          <xs:element ref="iodef:AdditionalData"
                         minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="restriction"
                         type="iodef:restriction-type"/>
        <xs:attribute name="ext-restriction"
                         type="xs:string" use="optional"/>
      </xs:complexType>
   </xs:element>

   <xs:element name="BulkObservable">
     <xs:complexType>
       <xs:sequence>
         <xs:element ref="iodef:BulkObservableFormat"
                         minOccurs="0"/>
         <xs:element name="BulkObservableList"
                         type="xs:string" minOccurs="0"/>
       </xs:sequence>
       <xs:attribute name="type"
                         type="observable-type-type"
                         use="required"/>
       <xs:attribute name="ext-type"
                         type="xs:string" use="optional"/>
     </xs:complexType>
   </xs:element>
```

```
<xs:simpleType name="observable-type-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="asn"/>
    <xs:enumeration value="atm"/>
    <xs:enumeration value="e-mail"/>
    <xs:enumeration value="ipv4-addr"/>
    <xs:enumeration value="ipv4-net"/>
    <xs:enumeration value="ipv4-net-mask"/>
    <xs:enumeration value="ipv6-addr"/>
    <xs:enumeration value="ipv6-net"/>
    <xs:enumeration value="ipv6-net-mask"/>
    <xs:enumeration value="mac"/>
    <xs:enumeration value="site-uri"/>
    <xs:enumeration value="fqdn"/>
    <xs:enumeration value="doman-name"/>
    <xs:enumeration value="domain-to-ipv4"/>
    <xs:enumeration value="domain-to-ipv6"/>
    <xs:enumeration value="domain-to-ipv4-timestamp"/>
    <xs:enumeration value="domain-to-ipv6-timestamp"/>
    <xs:enumeration value="ipv4-port"/>
    <xs:enumeration value="ipv6-port"/>
    <xs:enumeration value="windows-reg-key"/>
    <xs:enumeration value="file-hash"/>
    <xs:enumeration value="email-x-mailer"/>
    <xs:enumeration value="email-subject"/>
    <xs:enumeration value="http-user-agent"/>
    <xs:enumeration value="http-request-uri"/>
    <xs:enumeration value="mutex"/>
    <xs:enumeration value="file-path"/>
    <xs:enumeration value="user-name"/>
  </xs:restriction>
</xs:simpleType>

<xs:element name="BulkObservableFormat">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:Hash"
                  minOccurs="0"/>
      <xs:element ref="iodef:AdditionalData"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="IndicatorExpression">
  <xs:complexType>
    <xs:sequence>
      <xs:choice>
```

```
          <xs:element ref="iodef:IndicatorExpression"
                      minOccurs="0"/>
          <xs:element ref="iodef:Observable"
                      minOccurs="0" />
          <xs:element ref="iodef:ObservableReference"
                      minOccurs="0"/>
          <xs:element ref="iodef:IndicatorReference"
                      minOccurs="0"/>
       </xs:choice>
       <xs:element ref="iodef:AlternativeIndicatorID"
                   minOccurs="0" maxOccurs="unbounded"/>
     </xs:sequence>
     <xs:attribute name="operator"
                   type="indicatorexpression-operator-type"
                   use="required">
    </xs:attribute>
    </xs:complexType>
  </xs:element>

  <xs:simpleType name="indicatorexpression-operator-type">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="not"/>
      <xs:enumeration value="and"/>
      <xs:enumeration value="or"/>
      <xs:enumeration value="xor"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:element name="ObservableReference">
    <xs:complexType>
      <xs:attribute name="uid-ref"
                    type="xs:IDREF" use="required"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="IndicatorReference">
    <xs:complexType>
      <xs:attribute name="uid-ref"
                    type="xs:IDREF" use="optional"/>
      <xs:attribute name="euid-ref"
                    type="xs:string" use="optional"/>
      <xs:attribute name="version"
                    type="xs:string" use="optional"/>
    </xs:complexType>
  </xs:element>
<!--
  ================================================================
  == Miscellaneous simple classes                             ==
```

```
     ================================================================
   -->
     <xs:element name="Description" type="iodef:MLStringType"/>
     <xs:element name="URL" type="xs:anyURI"/>
   <!--
     ================================================================
     == IODEF Basic Data Types
==
     ================================================================
   -->
     <xs:simpleType name="PositiveFloatType">
       <xs:restriction base="xs:float">
         <xs:minExclusive value="0"/>
       </xs:restriction>
     </xs:simpleType>

     <xs:complexType name="MLStringType">
       <xs:simpleContent>
         <xs:extension base="xs:string">
           <xs:attribute name="translation-id"
                         type="xs:string" use="optional"/>
           <xs:attribute ref="xml:lang" />
         </xs:extension>
       </xs:simpleContent>
     </xs:complexType>

     <xs:simpleType name="PortlistType">
       <xs:restriction base="xs:string">
         <xs:pattern value="\d+(\-\d+)?(,\d+(\-\d+)?)*"/>
       </xs:restriction>
     </xs:simpleType>

     <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:attribute name="ext-restriction"
                     type="xs:string" use="optional"/>
     </xs:complexType>
```

```
   <xs:complexType name="ApplicationHeaderType" mixed="true">
     <xs:sequence>
        <xs:any namespace="##any" processContents="lax"
                minOccurs="0" maxOccurs="unbounded"/>
     </xs:sequence>
     <xs:attribute name="proto"
                   type="xs:integer" use="optional"/>
     <xs:attribute name="proto-name"
                   type="xs:integer" use="optional"/>
     <xs:attribute name="field"
                   type="xs:string" use="required"/>
     <xs:attribute name="dtype"
                   type="iodef:proto-dtype-type"
                   use="required"/>
     <xs:attribute name="observable-id"
                   type="xs:ID" use="optional"/>
   </xs:complexType>

 <!--
  ================================================================
  == Global attribute type declarations                        ==
  ================================================================
 -->
   <xs:simpleType name="yes-no-type">
     <xs:restriction base="xs:NMTOKEN">
       <xs:enumeration value="yes"/>
       <xs:enumeration value="no"/>
     </xs:restriction>
   </xs:simpleType>

   <xs:simpleType name="yes-no-unknown-type">
     <xs:restriction base="xs:NMTOKEN">
       <xs:enumeration value="yes"/>
       <xs:enumeration value="no"/>
       <xs:enumeration value="unknown"/>
     </xs:restriction>
   </xs:simpleType>

   <xs:simpleType name="restriction-type">
     <xs:restriction base="xs:NMTOKEN">
       <xs:enumeration value="default"/>
       <xs:enumeration value="public"/>
       <xs:enumeration value="partner"/>
       <xs:enumeration value="need-to-know"/>
       <xs:enumeration value="private"/>
       <xs:enumeration value="white"/>
       <xs:enumeration value="green"/>
       <xs:enumeration value="amber"/>
```

```
          <xs:enumeration value="red"/>
          <xs:enumeration value="ext-value"/>
        </xs:restriction>
      </xs:simpleType>

      <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>

      <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:enumeration value="ext-value"/>
        </xs:restriction>
      </xs:simpleType>

      <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="redirect-traffic"/>
          <xs:enumeration value="honeypot"/>
          <xs:enumeration value="upgrade-software"/>
          <xs:enumeration value="rebuild-asset"/>
          <xs:enumeration value="harden-asset"/>
          <xs:enumeration value="remediate-other"/>
          <xs:enumeration value="status-triage"/>
          <xs:enumeration value="status-new-info"/>
          <xs:enumeration value="watch-and-report"/>
```

```
      <xs:enumeration value="defined-coa"/>
      <xs:enumeration value="other"/>
      <xs:enumeration value="ext-value"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="dtype-type">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="boolean"/>
      <xs:enumeration value="byte"/>
      <xs:enumeration value="bytes"/>
      <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="ext-value"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="proto-dtype-type">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="boolean"/>
      <xs:enumeration value="byte"/>
      <xs:enumeration value="bytes"/>
      <xs:enumeration value="character"/>
      <xs:enumeration value="date-time"/>
      <xs:enumeration value="integer"/>
      <xs:enumeration value="real"/>
      <xs:enumeration value="string"/>
      <xs:enumeration value="xml"/>
      <xs:enumeration value="ext-value"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

9. **Security Considerations**

   The IODEF data model itself does not directly introduce security
   issues. Rather, it simply defines a representation for incident
   information. As the data encoded by the IODEF might be considered
   privacy sensitive by the parties exchanging the information or by
   those described by it, care needs to be taken in ensuring the
   appropriate disclosure during both document exchange and subsequent
   processing. The former must be handled by a messaging format, but
   the latter risk must be addressed by the systems that process, store,
   and archive IODEF documents and information derived from them.

   Executable content could be embedded into the IODEF document directly
   or through an extension. The IODEF parser should handle this content
   with care to prevent unintentional automated execution.

   The contents of an IODEF document may include a request for action or
   an IODEF parser may independently have logic to take certain actions
   based on information that it finds. For this reason, care must be
   taken by the parser to properly authenticate the recipient of the
   document and ascribe an appropriate confidence to the data prior to
   action.

   The underlying messaging format and protocol used to exchange
   instances of the IODEF MUST provide appropriate guarantees of
   confidentiality, integrity, and authenticity. The use of a
   standardized security protocol is encouraged. The Real-time Inter-
   network Defense (RID) protocol [RFC6545] and its associated transport
   binding IODEF/RID over HTTP/TLS [RFC6546] provide such security.

   In order to suggest data processing and handling guidelines of the
   encoded information, the IODEF allows a document sender to convey a
   privacy policy using the restriction attribute. The various
   instances of this attribute allow different data elements of the
   document to be covered by dissimilar policies. While flexible, it
   must be stressed that this approach only serves as a guideline from
   the sender, as the recipient is free to ignore it. The issue of
   enforcement is not a technical problem.

10. **IANA Considerations**

   This document registers a namespace, XML schema, and a number of
   registries that map to enumerated values defined in the schema.

10.1. **Namespace and Schema**

   This document uses URNs to describe an XML namespace and schema
   conforming to a registry mechanism described in [RFC3688]

   Registration for the IODEF namespace:

   o URI: urn:ietf:params:xml:ns:iodef-2.0

   o Registrant Contact: See the first author of the "Author's Address"
      section of this document.

   o XML: None. Namespace URIs do not represent an XML specification.

   Registration for the IODEF XML schema:

   o URI: urn:ietf:params:xml:schema:iodef-2.0

   o Registrant Contact: See the first author of the "Author's Address"
      section of this document.

   o XML: See the "IODEF Schema" in Section 8 of this document.

10.2. **Enumerated Value Registries**

   This document creates xx identically structured registries to be
   managed by IANA:

   o Name of the parent registry: "Incident Object Description Exchange
      Format v2 (IODEF)"

   o URL of the registry: http://www.iana.org/assignments/iodef2

   o Namespace format: A registry entry consists of:

      * Value. An enumerated value for a given IODEF attribute.

      * Description. A short description of the enumerated value.

      * Reference. An optional list of URIs to further describe the
         value.

   o Allocation policy: Expert Review per [RFC5226]

   The registries to be created are named in the table below in the
   "Registry Name" column. The initial values for the Value and
   Description fields of a given registry are listed in the "IV (Value)"
   and "IV (Description)" columns respectively. The "IV (Value)" points

to a given schema attribute or type per Section 8. Each enumerated
value in the schema gets a corresponding entry in a given registry.
The "IV (Description)" points to a section in the text of this
document. The initial value of the Reference field of every registry
entry described below should be this document.

Registry Name	IV (Value)	IV (Description)
Restriction	iodef-restriction-type	Section 3.3.1
Incident-purpose	Incident@purpose	Section 3.2
Incident-status	Incident@status	Section 3.2
Contact-role	Contact@role	Section 3.10
Contact-type	Contact@type	Section 3.10
RegistryHandle-registry	RegistryHandle@registry	Section 3.10.1
Expectation-action	iodef:action-type	Section 3.17
Discovery-source	Discovery@source	Section 3.12
SystemImpact-type	SystemImpact@type	Section 3.14.1
BusinessImpact-severity	BusinessImpact@severity	Section 3.14.2
BusinessImpact-type	BusinessImpact@type	Section 3.14.2
TimeImpact-metrics	TimeImpact@metric	Section 3.14.3
TimeImpact-duration	iodef:duration-type	Section 3.14.3
NodeRole-category	NodeRole@category	Section 3.20.2
System-category	System@category	Section 3.19
System-ownership	System@ownership	Section 3.19
Address-category	Address@category	Section 3.20.1
Counter-type	Counter@type	Section 3.20.3

Counter-unit	Counter@unit	Section 3.20.3	
DomainData-system-status	DomainData@system-status	Section 3.21	
DomainData-domain-status	DomainData@domain-status	Section 3.21	
RelatedDNS-record-type	RelatedDNS@record-type	Section 3.21.1	
RecordPattern-type	RecordPattern@type	Section 3.25.2	
RecordPattern-offsetunit	RecordPattern@offsetunit	Section 3.25.2	
Key-registryaction	Key@registryaction	Section 3.26.1	
HashData-scope	HashData@scope	Section 3.29	
BulkObservable-type	BulkObservable@type	Section 3.32.3.1	
AdditionalData-dtype	iodef:dtype-type	Section 3.9	
ApplicationHeader-proto-dtype	iodef:proto-dtype-type	Section 3.22.2	
SoftwareReference-dtype	SoftwareReference	Section 3.22.4	

Table 1: IANA Enumerated Value Registries

11. Acknowledgments

The following groups and individuals, listed alphabetically,
contributed substantially to this document and should be recognized
for their efforts.

o Kathleen Moriarty, EMC Corporation

o Brian Trammell, ETH Zurich

o Patrick Cain, Cooper-Cain Group, Inc.

o ... TODO many more to add ...

12. References

12.1. Normative References

[W3C.XML]   World Wide Web Consortium, "Extensible Markup Language
            (XML) 1.0 (Second Edition)", W3C Recommendation , October
            2000, <http://www.w3.org/TR/2000/REC-xml-20001006>.

[W3C.SCHEMA]
            World Wide Web Consortium, "XML XML Schema Part 1:
            Structures Second Edition", W3C Recommendation , October
            2004, <http://www.w3.org/TR/xmlschema-1/>.

[W3C.SCHEMA.DTYPES]
            World Wide Web Consortium, "XML Schema Part 2: Datatypes
            Second Edition", W3C Recommendation , October 2004,
            <http://www.w3.org/TR/xmlschema-2/>.

[W3C.XMLNS]
            World Wide Web Consortium, "Namespaces in XML", W3C
            Recommendation , January 1999,
            <http://www.w3.org/TR/REC-xml-names/>.

[W3C.XPATH]
            World Wide Web Consortium, "XML Path Language (XPath)
            2.0", W3C Candidate Recommendation , June 2006,
            <http://www.w3.org/TR/xpath20/>.

[W3C.XMLSIG]
            World Wide Web Consortium, "XML Signature Syntax and
            Processing 2.0", W3C Candidate Recommendation , June 2008,
            <http://www.w3.org/TR/xmldsig-core/>.

[IEEE.POSIX]
            Institute of Electrical and Electronics Engineers,
            "Information Technology - Portable Operating System
            Interface (POSIX) - Part 1: Base Definitions",
            IEEE 1003.1, June 2001.

[RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
            Requirement Levels", RFC 2119, March 1997.

[RFC5646]   Philips, A. and M. Davis, "Tags for Identifying of
            Languages", RFC 5646, September 2009.

[RFC3986]   Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform
            Resource Identifiers (URI): Generic Syntax", RFC 3986,
            January 2005`.

[RFC2978]   Freed, N. and J. Postel, "IANA Charset Registration
            Procedures", BCP 2978, October 2000.

[RFC4519]   Sciberras, A., "Schema for User Applications", RFC 4519,
            June 2006.

[RFC5322]   Resnick, P., "Internet Message Format", RFC 5322, October
            2008.

[RFC3339]   Klyne, G. and C. Newman, "Date and Time on the Internet:
            Timestamps", RFC 3339, July 2002.

[RFC-ENUM]
            Montville, A. and D. Black, "IODEF Enumeration Reference
            Format", RFC ENUM, January 2015.

[ISO8601]   International Organization for Standardization,
            "International Standard: Data elements and interchange
            formats - Information interchange - Representation of
            dates and times", ISO 8601, Second Edition, December 2000.

[ISO4217]   International Organization for Standardization,
            "International Standard: Codes for the representation of
            currencies and funds, ISO 4217:2001", ISO 4217:2001,
            August 2001.

[RFC3688]   Mealling, M., "The IETF XML Registry", RFC 3688, January
            2004.

[IANA.Ports]
            Internet Assigned Numbers Authority, "Service Name and
            Transport Protocol Port Number Registry", January 2014,
            <http://www.iana.org/assignments/service-names-port-
            numbers/service-names-port-numbers.txt>.

[IANA.Protocols]
            Internet Assigned Numbers Authority, "Assigned Internet
            Protocol Numbers", January 2014,
            <http://www.iana.org/assignments/protocol-numbers/
            protocol-numbers.txt>.

[RFC3629]   Yergeau, F., "UTF-8, a transformation format of ISO
            10646", RFC 3629, November 2003.

[RFC2781]   Hoffman, P. and F. Yergeau, "UTF-16, an encoding of ISO
            10646", RFC 2781, February 2000.

[IANA.Media]
          Internet Assigned Numbers Authority, "Media Types", March
          2015, <http://www.iana.org/assignments/media-types/
          media-types.xhtml>.

12.2. Informative References

[RFC5070] Danyliw, R., Meijer, J., and Y. Demchenko, "Incident
          Object Description Exchange Format", RFC 5070, December
          2007.

[refs.requirements]
          Keeni, G., Demchenko, Y., and R. Danyliw, "Requirements
          for the Format for Incident Information Exchange (FINE)",
          Work in Progress, June 2006.

[RFC4765] Debar, H., Curry, D., Debar, H., and B. Feinstein,
          "Intrusion Detection Message Exchange Format", RFC 4765,
          March 2007.

[RFC6545] Moriarty, K., "Real-time Inter-network Defense (RID)",
          RFC 6545, April 2012.

[RFC6546] Trammell, B., "Transport of Real-time Inter-network
          Defense (RID) Messages over HTTP/TLS", RFC 6546, April
          2012.

[RFC5901] Cain, P. and D. Jevans, "Extensions to the IODEF-Document
          Class for Reporting Phishing", RFC 5901, July 2010.

[NIST800.61rev2]
          Cichonski, P., Millar, T., Grance, T., and K. Scarfone,
          "NIST Special Publication 800-61 Revision 2: Computer
          Security Incident Handling Guide", January 2012,
          <http://csrc.nist.gov/publications/nistpubs/800-61rev2/
          SP800-61rev2.pdf>.

[RFC3982] Newton, A. and M. Sanz, "IRIS: A Domain Registry (dreg)
          Type for the Internet Registry Information Service
          (IRIS)", RFC 3982, January 2005.

[KB310516]
          Microsoft Corporation, "How to add, modify, or delete
          registry subkeys and values by using a registration
          entries (.reg) file", December 2007.

[RFC4180] Shafranovich, Y., "Common Format and MIME Type for Comma-
          Separated Values (CSV) File", RFC 4180, October 2005.

   [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an
              IANA Considerations Section in RFCs", RFC 5226, May 2008.

Authors' Addresses

   Roman Danyliw
   CERT - Software Engineering Institute
   Pittsburgh, PA
   USA

   EMail: rdd@cert.org

   Paul Stoecker
   RSA
   Reston, VA
   USA

   EMail: paul.stoecker@rsa.com