INTERNET-DRAFT
Intended Status: Standards Track
Updates: 5070 (if approved)
Expires: December 22, 2014

Adam W. Montville
(Tripwire)
David Black
(EMC)

June 20, 2014

IODEF Enumeration Reference Format
draft-ietf-mile-enum-reference-format-06

Abstract

The Incident Object Description Exchange Format (IODEF) provides a Reference class used to reference external entities (such as enumeration identifiers). However, the method of external entity identification has been left unstructured. This document describes a method to provide structure for referencing external entities for the IODEF Reference class and thus updates IODEF's ReferenceName (RFC5070).

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

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

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

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

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

Table of Contents

1 Introduction

There is an identified need to specify a format to include relevant enumeration values in an IODEF document. It is anticipated that this requirement will exist in other standardization efforts within several IETF Working Groups, but the scope of this document pertains solely to IODEF [IODEF].

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Referencing External Enumerations

The need is to place enumeration identifiers and their references in IODEF [IODEF]’s Reference class. There are several ways to accomplish this goal, but the most appropriate at this point is to require a specific format for the ReferenceName string of the IODEF [IODEF] Reference class, and use an IANA registry to manage the resulting reference formats.

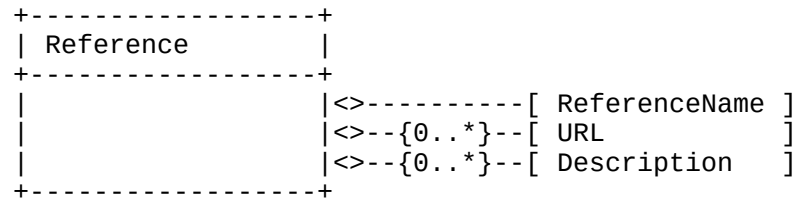

FIGURE 1: IODEF [IODEF] Reference Class

Per IODEF [IODEF] the ReferenceName is of type ML_STRING. This becomes problematic when specific references, especially enumerations such as CVE [CVE], CCE [CCE], CPE [CPE] and so on, are referenced - how is an implementer to know which type of reference this is, and thus how to parse it? One solution, presented here, is to require that ReferenceName follow a particular format.

Inclusion of such enumerations, especially those related to security automation, is important to incident communication and investigation. Typically, an enumeration identifier is simply an identifier with a specific format as defined by an external party.

2.1 Reference Name Format

The Reference Name Format uses XML to provide the structure for enumeration identification, and requires that a specific Index be associated with the ID. An implementer can look up the ID type (as referenced by the Index) in the IANA table (see Section 4) to understand how the ID is structured. The Index field in the XML unambiguously indicates which IANA registry entry is to be used to correctly reference the enumeration specification, which avoids interpretation of version strings that may have specification-specific formats.

```
<Reference>
   <ReferenceName>
      <Index>1</Index>
      <ID>CXI-1234-XYZ</ID>
   </ReferenceName>
   <URL>http://cxi.example.com</URL>
   <Description>Foo</Description>
</Reference>
```

LISTING 1: Example Use of IODEF Enumeration Reference Format

Information in the IANA table (see Section 4) would include:

```
Full Name: Concept X Identifier
Index: 1
Version: any
Specification URI: http://cxi.example.com/spec_url
```

2.3 Reference Method Applicability

While the scope of this document pertains to IODEF [IODEF], it should be readily apparent that any standard needing to reference an enumeration identified by a specially formatted string can use this method of providing structure after the standard has been published. In effect, this method provides a standardized interface for enumerations, thus allowing a loose coupling between a given standard and the enumeration identifiers it needs to reference now and in the future.

3 Security Considerations

Producers of IODEF [IODEF] content SHOULD be careful to ensure a proper mapping of enumeration reference ID elements to the correct Index. Potential consequences of not mapping correctly include inaccurate information references and similar distribution of misinformation.

Use of enumeration reference IDs from trusted sources SHOULD be preferred by implementers to mitigate the risk of receiving and/or providing misinformation. Trust decisions with respect to enumeration reference providers is beyond the scope of this document.

In some cases it might be possible for a third-party to host content associated with an enumeration reference ID. In such a circumstance, trust SHOULD extend from the origin of the enumeration reference ID to the third-party, effectively making the third-party a trusted third-party in the context of providing a particular set of enumeration reference IDs.

4 IANA Considerations

This document specifies an identifier format for the IODEF [IODEF] ReferenceName string of the Reference class.

This memo creates the following registry for IANA to manage:

Name of the Registry: "Enumeration Reference Type Identifiers"

Fields to record in the registry:

Full Name: The full name of the enumeration as a string from the printable ASCII character set.

Abbreviation: An abbreviation may be an acronym - it consists of upper-case characters (at least two, upper-case is used to avoid mismatches due to case differences), as specified by this ABNF [RFC5234] syntax:

```
ABBREVIATION = 2*UC-ALPHA      ; At least two
UC-ALPHA = %x41-5A             ; A-Z
```

Multiple registrations MAY use the same Abbreviation but MUST have different Versions.

Index: This is an IANA-assigned positive integer that

identifies the registration. The first entry added to this registry uses the value 1, and this value is incremented for each subsequent entry added to the registry.

Version: The version of the enumeration as a free-form string from the printable ASCII character set excepting white space.

Specification URI: A list of one or more URIs [RFC3986] from which the registered specification can be obtained. The registered specification MUST be readily and publicly available from that URI. The URI SHOULD be a stable reference to a specific version of the specification. URIs that designate the latest version of a specification (which changes when a new version appears) SHOULD NOT be used.

Initial registry contents: None.

Allocation Policy: Specification Required [RFC5226] (which implies Expert Review [RFC5226]).

The Designated Expert is expected to consult with the MILE (Managed Incident Lightweight Exchange) working group or its successor if any such WG exists (e.g., via email to the working group's mailing list). The Designated Expert is expected to review the request and validate the appropriateness of the enumeration for the attribute. If a specification is associated with the request, it MUST be reviewed by the Designated Expert.

The Designated Expert is expected to ensure that the Full Name, Abbreviation and Version are appropriate and that the information at the Specification URI is sufficient to unambiguously parse identifiers based on that specification. Additionally, the Designated Expert should prefer short Abbreviations over long ones.

5 IODEF XML Schema Changes

The changes to the IODEF [IODEF] schema are detailed below. Note that in addition to the element changes described below, certain attributes of the xs:schema element in the schema document should be updated, as well as certain information in the document class.

The xs:schema attributes are updated as follows:

targetNamespace="urn:ietf:params:xml:ns:iodef-1.01"

xmlns="urn:ietf:params:xml:ns:iodef-1.01"

xmlns:iodef="urn:ietf:params:xml:ns:iodef-1.01"

The IODEF-Document element description is updated to have a fixed version of "1.01" instead of "1.00", such that:

```
<xs:element name="IODEF-Document">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:Incident" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="version" type="xs:string" fixed="1.00"/>
    <xs:attribute name="lang" type="xs:language" use="required"/>
    <xs:attribute name="formatid" type="xs:string"/>
  </xs:complexType>
</xs:element>
```

Is changed to:

```
<xs:element name="IODEF-Document">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:Incident" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="version" type="xs:string" fixed="1.01"/>
    <xs:attribute name="lang" type="xs:language" use="required"/>
    <xs:attribute name="formatid" type="xs:string"/>
  </xs:complexType>
</xs:element>
```

The ReferenceName element is updated by replacing the following line in the 1.00 schema:

```
<xs:element name="ReferenceName" type="iodef:MLStringType"/>
```

With:

```
<xs:element name="ReferenceName">
  <xs:complexType>
    <xs:sequence>
     <xs:element name="Index" type="xs:integer"/>
     <xs:element name="ID" type="xs:NCName"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

LISTING 2: IODEF Enumeration Reference Format Schema Changes

This change to the IODEF [IODEF] schema may cause interoperability issues depending on tool implementation. If strict schema validation is used by a 1.00 tool when parsing an incoming IODEF [IODEF] 1.01

document, the elements under ReferenceName may not be understood and could cause errors. If strict schema validation is not used when parsing an incoming IODEF [IODEF] 1.01 document with a 1.00 tool, the elements under ReferenceName should simply be present in the object model, but this may lead to unpredictable results.

Implementers are encouraged to update their code to handle the IODEF [IODEF] 1.00 schema and the 1.01 schemas explicitly to avoid any unhandled exceptions that may occur when a 1.00 implementation attempts to parse a 1.01 document.

6 References

6.1 Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[IODEF] Danyliw, R., Meijer, J., and Y. Demchenko, "The Incident Object Description Exchange Format", RFC 5070, December 2007.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC5234] Crocker, D., Ed., and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

6.2 Informative References

[CCE] http://cce.mitre.org

[CPE] http://cpe.mitre.org

[CVE] http://cve.mitre.org

Authors' Addresses

Adam W. Montville

EMail: adam@stoicsecurity.com

David Black
EMC Corporation

EMail: david.black@emc.com

MILE
Internet-Draft
Intended status: Informational
Expires: January 5, 2015

C. Inacio
CMU
D. Miyamoto
UTokyo

July 4, 2014

MILE Implementation Report
draft-ietf-mile-implementreport-00

Abstract

This document is a collection of implementation reports from vendors, consortiums, and researchers who have implemented one or more of the standards published from the IETF INCident Handling (INCH) and Management Incident Lightweight Exchange (MILE) working groups.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on January 5, 2015.

Copyright Notice

Table of Contents

1. Introduction

This document is a collection of implementation reports from vendors and researchers who have implemented one or more of the standards published from the INCH and MILE working groups. The standards include:

- Incident Object Description Exchange Format (IODEF) v1, RFC5070,
- Incident Object Description Exchange Format (IODEF) v2, RFC5070-bis,
- Extensions to the IODEF-Document Class for Reporting Phishing, RFC5901
- Sharing Transaction Fraud Data, RFC5941
- IODEF-extension for Structured Cybersecurity Information, RFCXXXX
- Real-time Inter-network Defense (RID), RFC6545

- Transport of Real-time Inter-network Defense (RID) Messages over HTTP/TLS, RFC6546.

The implementation reports included in this document have been provided by the team or product responsible for the implementations of the mentioned RFCs. Additional submissions are welcome and should be sent to the draft editor. A more complete list of implementations, including open source efforts and vendor products, can also be found at the following location:

http://siis.realmv6.org/implementations/

2. Consortiums and Information Sharing and Analysis Centers (ISACs)

2.1. Anti-Phishing Working Group

Description of how IODEF is used will be provided in a future revision.

2.2. Advanced Cyber Defence Centre (ACDC)

Description of how IODEF is used will be provided in a future revision. http://www.botfree.eu/

3. Open Source Implementations

3.1. EMC/RSA RID Agent

The EMC/RSA RID agent is an open source implementation of the Internet Engineering Task Force (IETF) standards for the exchange of incident and indicator data. The code has been released under an MIT license and development will continue with the open source community at the Github site for RSA Intelligence Sharing:

https://github.com/RSAIntelShare/RID-Server.git

The code implements the RFC6545, Real-time Inter-network Defense (RID) and RFC6546, Transport of RID over HTTP/TLS protocol. The code supports the evolving RFC5070-bis Incident Object Description Exchange Format (IODEF) data model from the work in the IETF working group Managed Incident Lightweight Exchange (MILE).

3.2. NICT IODEF-SCI implementation

Japan’s National Institute of Information and Communications Technology (NICT) Network Security Research Institute implemented open source tools for exchanging, accumulating, and locating IODEF-SCI documents.

Three tools are available in GitHub. They assist the exchange of IODEF-SCI documents between parties. IODEF-SCI is the IETF draft that extends IODEF so that IODEF document can embed structured cybersecurity information (SCI). For instance, it can embed MMDEF, CEE, MAEC in XML and CVE identifiers.

The three tools are generator, exchanger, and parser. The generator generates IODEF-SCI document or appends an XML to existing IODEF document. The exchanger sends the IODEF document to its correspondent node. The parser receives, parses, and stores the IODEF-SCI document. It also equips the interface that enable users to locate IODEF-SCI documents it has ever received. The code has been released under an MIT license and development will continue here.

Note that users can enjoy this software with their own responsibility.

Available Online:

https://github.com/TakeshiTakahashi/IODEF-SCI

4. Vendor Implementations

4.1. Deep Secure

Deep-Secure Guards are built to protect a trusted domain from:

- releasing sensitive data that does not meet the organisational security policy
- applications receiving badly constructed or malicious data which could exploit a vulnerability (known or unknown)

Deep-Secure Guards support HTTPS and XMPP (optimised server to server protocol) transports. The Deep-Secure Guards support transfer of XML based business content by creating a schema to translate the known good content to and from the intermediate format. This means that the Deep-Secure Guards can be used to protect:

- IODEF/RID using the HTTPS transport binding (RFC 6546)
- IODEF/RID using an XMPP binding
- ROLIE using HTTPS transport binding (draft-field-mile-rolie-02)
- STIX/TAXII using the HTTPS transport binding

Deep-Secure Guards also support the SMTP transport and perform deep content inspection of content including XML attachments. The Mail Guard supports S/MIME and Deep Secure are working on support for the upcoming PLASMA standard which enables information centric policy enforcement of data.

4.2. IncMan Suite, DFLabs

The Incident Object Description Exchange Format, documented in the RFC 5070, defines a data representation that provides a framework for sharing information commonly exchanged by Computer Security Incident Response Teams (CSIRTs) about computer security incidents. IncMan Suite implements the IODEF standard for exchanging details about incidents, either for exporting and importing activities. This has been introduced to enhance the capabilities of the various CSIRT, to facilitate collaboration and sharing of useful experiences, conveying awareness on specific cases.

The IODEF implementation is specified as an XML schema, therefore all data are stored in an xml file: in this file all data of an incident are organized in a hierarchical structure to describe the various objects and their relationships.

IncMan Suite relies on IODEF as a transport format, composed by various classes for describing the entities which are part of the incident description: for instance the various relevant timestamps (detect time , start time, end time, report time), the techniques used by the intruders to perpetrate the incident, the impact of the incident, either technical and non-technical (time and monetary) and obviously all systems involved in the incident.

4.2.1. Exporting Incidents

Each incident defined in IncMan Suite can be exported via a User Interface feature and it will populate an xml document. Due to the nature of the data processed, the IODEF extraction might be considered privacy sensitive by the parties exchanging the information or by those described by it. For this reason, specific care needs to be taken in ensuring the distribution to an appropriate audience or third party, either during the document exchange and subsequent processing.

The xml document generated will include description and details of the incident along with all the systems involved and the related information. At this stage it can be distributed for import into a remote system.

4.2.2. Importing Incidents

IncMan Suite provides a functionality to import incidents stored in files and transported via IODEF-compliant xml documents. The importing process comprises of two steps: firstly, the file is inspected to validate if well formed, then all data are uploaded inside the system.

If an incident is already existing in the system with the same incident id, the new one being imported will be created under a new id. This approach prevents from accidentally overwriting existing info or merging inconsistent data.

IncMan Suite includes also a feature to upload incidents from emails.

The incident, described in xml format, can be stored directly into the body of the email message or transported as an attachment of the email. At regular intervals, customizable by the user, IncMan Suite monitors for incoming emails, filtered by a configurable white-list and black-list mechanism on the sender’s email account, then a parser processes the received email and a new incident is created automatically, after having validated the email body or the attachment to ensure it is a well formed format.

4.3. Surevine Proof of Concept

XMPP is enhanced and extended through the XMPP Extension Protocols (or XEPs). XEP-0268 (http://xmpp.org/extensions/xep-0268.html) describes incident management (using IODEF) of the XMPP network itself, effectively supporting self-healing the XMPP network. In order to more generically cover incident management of a network and over a network, XEP-0268 requires some updates. We are working on these changes together with a new XEP that supports "social networking" over XMPP, enhancing the publish-and-subscribe XEP (XEP-0060). This now allows nodes to publish any type of content and subscribe to and therefore receive the content. XEP-0268 will be used to describe IODEF content. We now have an alpha version of the server-side software and client-side software required to demonstrate the "social networking" capability and are currently enhancing this to support Cyber Incident management in real-time.

4.4. MANTIS Cyber-Intelligence Management Framework

MANTIS provides an example implementation of a framework for managing cyber threat intelligence expressed in standards such as STIX, CybOX, IODEF, etc. The aims of providing such an example implementation are:

- To aide discussions about emerging standards such as STIX, CybOX et al. with respect to questions regarding tooling: how would a certain aspect be implemented, how do changes affect an implementation? Such discussions become much easier and have a better basis if they can be lead in the context of example tooling that is known to the community.

- To lower the entrance barrier for organizations and teams (esp. CERT teams) in using emerging standards for cyber-threat intelligence management and exchange.

- To provide a platform on the basis of which research and community-driven development in the area of cyber-threat intelligence management can occur.

5. Vendors with Planned Support

5.1. Threat Central, HP

HP has developed HP Threat Central, a security intelligence platform that enables automated, real-time collaboration between organizations to combat today’s increasingly sophisticated cyber attacks. One way automated sharing of threat indicators is achieved is through close integration with the HP ArcSight SIEM for automated upload and consumption of information from the Threat Central Server. In addition HP Threat Central supports open standards for sharing threat information so that participants who do not use HP Security Products can participate in the sharing ecosystem. General availability of Threat Central will be in 2014. It is planned that future versions also support IODEF for the automated upload and download of threat information.

6. Implementation Guide

The section aims at sharing the tips for development of IODEF-capable systems.

6.1. Code Generators

For implementing IODEF-capable systems, it is feasible to employ code generators for XML Schema Document (XSD). The generators are used to save development costs since they automatically create useful libraries for accessing XML attributes, composing messages, and/or validating XML objects. The IODEF XSD was defined in section 8 of RFC 5070, and is availabe at http://www.iana.org/assignments/xml-registry/schema/iodef-1.0.xsd.

However, there still remains some problem. Due to the complexity of IODEF XSD, some code generators could not generate from the XSD file. The tested code generators were as follows.

- XML::Pastor [XSD:Perl] (Perl)
- RXSD [XSD:Ruby] (Ruby)
- PyXB [XSD:Python] (Python)
- JAXB [XSD:Java] (Java)
- CodeSynthesis XSD [XSD:Cxx] (C++)
- Xsd.exe [XSD:CS] (C#)

For instance, we have used XML::Pastor, but it could not properly understand its schema due to the complexity of IODEF XSD. The same applies to RXSD and JAXB. Only PyXB, CodeSynthesis XSD and Xsd.exe were able to understand the schema.

There is no recommended workaround, however, a double conversion of XSD file is one option to go through the situation; it means XSD is serialized to XML, and it is again converted to XSD. The resultant XSD was process-able by the all tools above.

It should be noted that IODEF uses '-' (hyphen) symbols in its classes or attributes, listed as follows.

- IODEF-Document Class; it is the top level class in the IODEF data model described in section 3.1 of [RFC5070].
- The vlan-name and vlan-num Attribute; according to section 3.16.2 of [RFC5070], they are the name and number of Virtual LAN and are the attributes for Address class.
- Extending the Enumerated Values of Attribute; according to section 5.1 of [RFC5070], it is a extension techniques to add new enumerated values to an attribute, and has a prefix of "ext-", e.g., ext-value, ext-category, ext-type, and so on.

According to the language specification, many programing language prohibit to contain '-' symbols in the name of class. The code generators must replace or remove '-' when building the librarlies. They should have the name space to restore '-' when outputting the XML along with IODEF XSD.

6.2. Usability

Here notes some tips to avoid problems.

- IODEF has category attribute for NodeRole class. Though various categories are described, they are not enough. For example, in the case of web mail servers, you should choose either "www" or "mail". One suggestion is selecting "mail" as the category attribute and adding "www" for another attirbute.

- The numbering of Incident ID needs to be considered. Otherwise, information, such as the number of incidents within certain period could be observed by document receivers. For instance, we could randomize the assignment of the numbers.

7. Acknowledgements

The MILE Implementation report has been compiled through the submissions of implementers of INCH and MILE working group standards. A special note of thanks to the following contributors:

John Atherton, Surevine

Humphrey Browning, Deep-Secure

Dario Forte, DFLabs

Tomas Sander, HP

Ulrich Seldeslachts, ACDC

Takeshi Takahashi, National Institute of Information and Communications Technology Network Security Research Institute

Kathleen Moriarty, EMC

Bernd Grobauer, Siemens

8. IANA Considerations

This memo includes no request to IANA.

9. Security Considerations

This draft provides a summary of implementation reports from researchers and vendors who have implemented RFCs and drafts from the MILE and INCH working groups. There are no security considerations added in this draft because of the nature of the document.

10. Informative References

[RFC5070] Danyliw, R., Meijer, J., and Y. Demchenko, "The Incident Object Description Exchange Format", RFC 5070, December 2007.

[RFC5901] Cain, P. and D. Jevans, "Extensions to the IODEF-Document Class for Reporting Phishing", RFC 5901, July 2010.

[RFC5941] M'Raihi, D., Boeyen, S., Grandcolas, M., and S. Bajaj, "Sharing Transaction Fraud Data", RFC 5941, August 2010.

[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.

[XSD:CS] Microsoft, "XML Schema Definition Tool (Xsd.exe)", <http://www.codesynthesis.com/>.

[XSD:Cxx] CodeSynthesis, "XSD - XML Data Binding for C++", <http://www.codesynthesis.com/>.

[XSD:Java] Project Kenai, "JAXB Reference Implementation", <https://jaxb.java.net/>.

[XSD:Perl] Ulsoy, A., "XML::Pastor", <http://search.cpan.org/~aulusoy/XML-Pastor-1.0.4/>.

[XSD:Python] Bigot, P., "PyXB: Python XML Schema Bindings", <https://pypi.python.org/pypi/PyXB>.

[XSD:Ruby] Morsi, M., "RXSD - XSD / Ruby Translator", <https://github.com/movitto/RXSD>.

Authors' Addresses

Chris Inacio
Carnegie Mellon University
4500 5th Ave., SEI 4108
Pittsburgh, PA 15213
US

Email: inacio@andrew.cmu.edu

Daisuke Miyamoto
The Univerisity of Tokyo
2-11-16 Yayoi, Bunkyo
Tokyo 113-8658
JP

Email: daisu-mi@nc.u-tokyo.ac.jp

MILE Working Group
Internet-Draft
Obsoletes: 5070 (if approved)
Intended status: Standards Track
Expires: August 18, 2014

R. Danyliw
CERT
P. Stoecker
RSA

February 14, 2014

The Incident Object Description Exchange Format v2
draft-ietf-mile-rfc5070-bis-06

Abstract

The Incident Object Description Exchange Format (IODEF) defines a data representation that provides a framework 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

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on August 18, 2014.

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:

- cyber intelligence to characterize threats;
- cyber incident reports to document particular cyber security events or relationships between events;
- cyber event mitigation to request proactive and reactive mitigation approaches to cyber intelligence or incidents; and

- 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:

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

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

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

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.

- All of the RFC5070 Errata was implemented.

- Imported the xmlns:ds namespace to include digital signature hash classes.
- The @indicator-* attributes were added to various classes to reference commonly shared indicators.
- The following classes were added to IODEF-Document: AdditionalData.
- The following classes were added to Incident and EventData: Discovery.
- 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.
- The following classes were added to the Record class: FileName and WindowsRegistryKeysModified.
- The following classes were added to the RelatedActivity class: ThreatActor, Campaign, Confidence, Description, and AdditionalData.
- The following classes were added to Assessment: BusinessImpact.
- The following classes were added to Node: PostalAddress and DomainData. The following classes were removed from Node: Removed NodeName and DateTime.
- The following classes were added to the Contact class: ContactTitle.
- The following classes were added to Expectation and HistoryItem: DefinedCOA.
- (for consideration) The following attributes was added to the SoftwareType complexType: user-agent.
- Additional enumerated values were added to the following attributes: @restriction, {Expectation, HistoryItem}@action, NodeRole@category, Incident@purpose, Contact@role, AdditionalData@dtype, System@spoofed.

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.

- 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.

- 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.

- 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.

- 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.

- 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

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

The ML_STRING data type is implemented as an "iodef:MLStringType" in the schema.

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.

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.

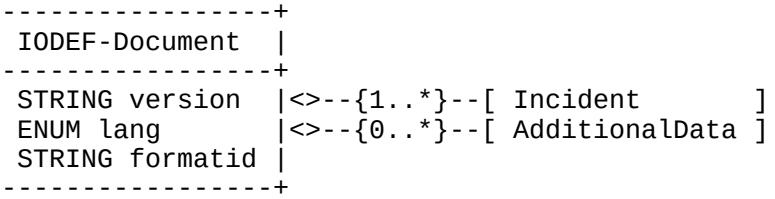

Figure 1: IODEF-Document Class

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"

lang
Required. ENUM. A valid language code per [RFC4646] constrained by the definition of "xs:language". The interpretation of this code is described in Section 6.

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

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.

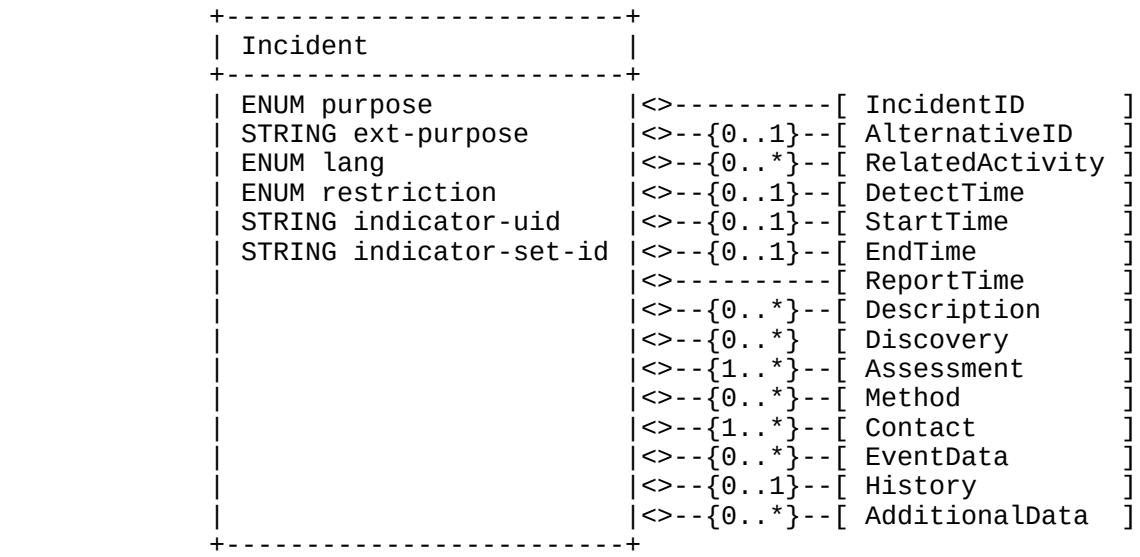

Figure 2: The Incident Class

The aggregate classes that constitute Incident are:

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

AlternativeID
: Zero or one. 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.

ReportTime
: One. The time the incident was reported.

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.

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 five 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). 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.

ext-purpose
Optional. STRING. A means by which to extend the purpose attribute. See Section 5.1.

lang
Optional. ENUM. A valid language code per [RFC4646] constrained by the definition of "xs:language". The interpretation of this code is described in Section 6.

restriction
Optional. ENUM. See Section 3.3.1.

indicator-uid
Optional. STRING. See Section 3.3.2.

indicator-set-id
Optional. STRING. 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.

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’.

3.3.2. Indicator Attributes

For data elements that are commonly used as indicators, the data model uses four attributes to facilitate their ...

indicator-uid
: STRING. See Section 3.3.2.

indicator-set-id
: STRING. See Section 3.3.2.

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.

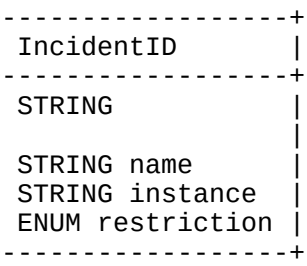

Figure 3: The IncidentID Class

The IncidentID class has three 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".

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.

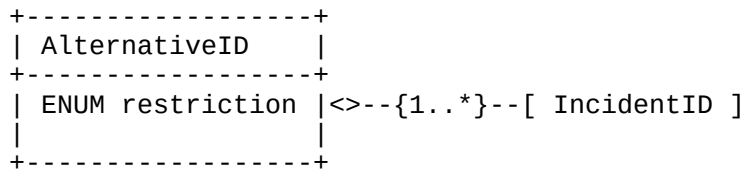

Figure 4: The AlternativeID Class

The aggregate class that constitutes AlternativeID is:

IncidentID
: One or more. The incident tracking number of another CSIRT.

The AlternativeID class has one attribute:

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

3.6. 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.

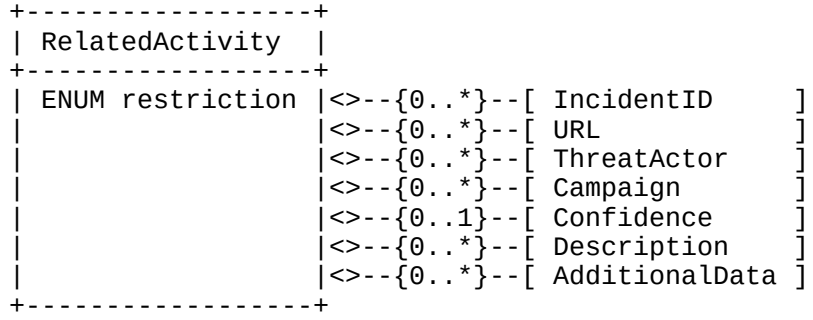

Figure 5: RelatedActivity Class

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 one attribute:

restriction
Optional. ENUM. See Section 3.3.1.

3.7. ThreatActor Class

The ThreatActor class describes a given actor.

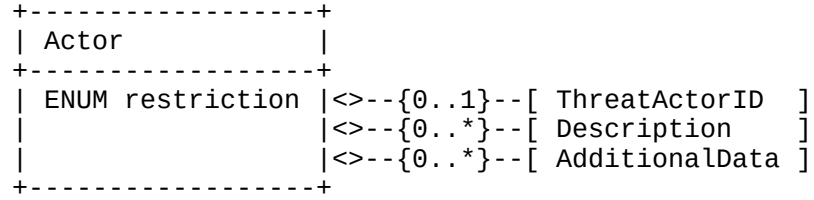

Figure 6: ThreatActor Class

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 one attribute:

restriction
Optional. ENUM. See Section 3.3.1.

3.8. Campaign Class

The Campaign class describes a ...

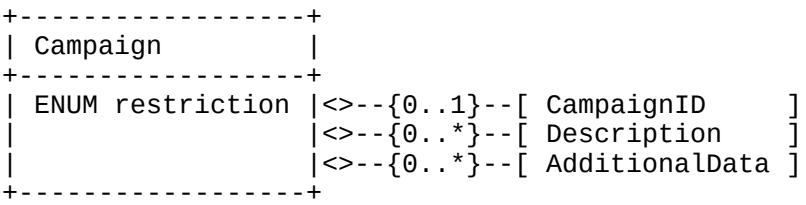

Figure 7: Campaign Class

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 one attribute:

restriction
Optional. ENUM. See Section 3.3.1.

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.

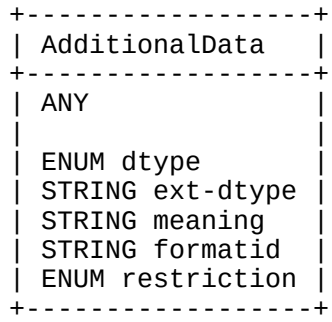

Figure 8: The AdditionalData Class

The AdditionalData class has five attributes:

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

1. boolean. The element 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.

ext-dtype
: Optional. STRING. A means by which to extend the dtype attribute. See Section 5.1.

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.

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.

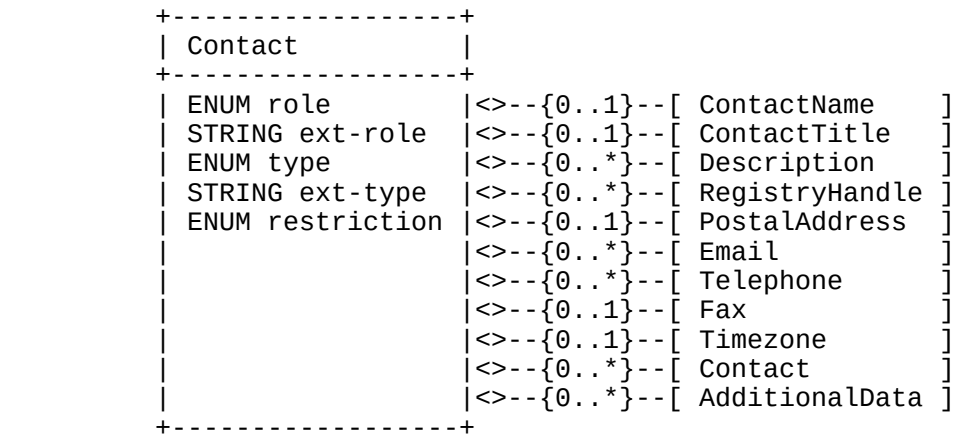

Figure 9: The Contact Class

The aggregate classes that constitute the Contact class are:

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

ContactTitle
: Zero or one. 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 five attributes:

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

1. creator. The entity that generate the 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. le. A law enforcement entity supporting the investigation of activity affecting an asset or organization.

15. vendor. The vendor that produces an asset.

16. ext-value. An escape value used to extend this attribute. See Section 5.1.

ext-role
Optional. STRING. A means by which to extend the role attribute. See Section 5.1.

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

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.

ext-type
Optional. STRING. A means by which to extend the type attribute. See Section 5.1.

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

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.

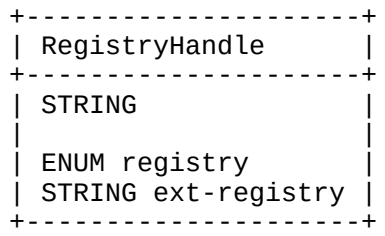

Figure 10: The RegistryHandle Class

The RegistryHandle class has two attributes:

registry
: Required. ENUM. The database to which the handle belongs. 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.

ext-registry
: Optional. STRING. A means by which to extend the registry attribute. See Section 5.1.

3.10.2. PostalAddress Class

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

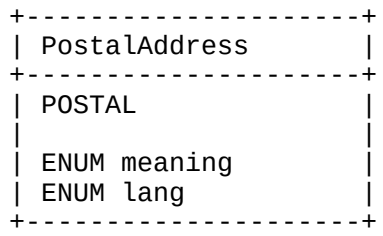

Figure 11: The PostalAddress Class

The PostalAddress class has two attributes:

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

lang
Optional. ENUM. A valid language code per [RFC4646] constrained by the definition of "xs:language". 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).

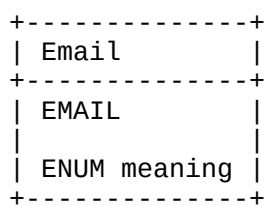

Figure 12: The Email Class

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).

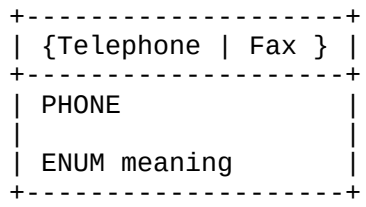

Figure 13: The Telephone and Fax Classes

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

The element content of each class is a timestamp formatted according to the DATETIME data type (see Section 2.8).

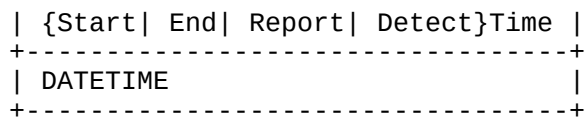

Figure 14: The Time Classes

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. This timestamp MUST be the time at which the IODEF document was generated.

3.11.5. 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.

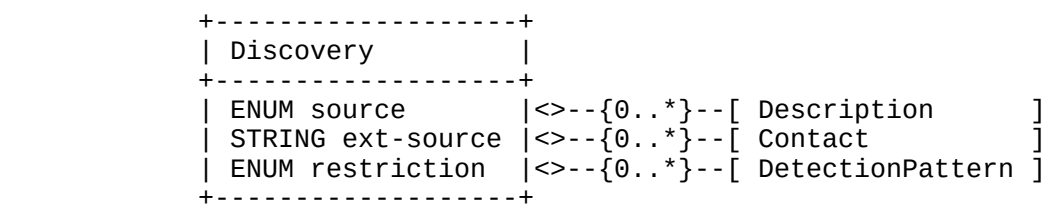

Figure 15: The Discovery Class

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 three attribute:

source
: Optional. ENUM. Categorizes the techniques used to discover the incident. These values are partially derived from Table 3-1 of [NIST800.61rev2].

  1. idps. Intrusion Detection or Prevention system.

2. siem. Security Information and Event Management System.

3. av. Antivirus or and antispam software.

4. file-integrity. File integrity checking software.

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

6. os-log. Operating system logs.

7. application-log. Application logs.

8. device-log. Network device logs.

9. network-flow. Network flow analysis.

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

11. internal-notification. A party within the organization discovered the activity

12. external-notification. A party outside of the organization discovered the activity.

13. unknown. Unknown detection approach.

14. ext-value. An escape value used to extend this attribute. See Section 5.1.

ext-source
Optional. STRING. A means by which to extend the source attribute. See Section 5.1.

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

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.

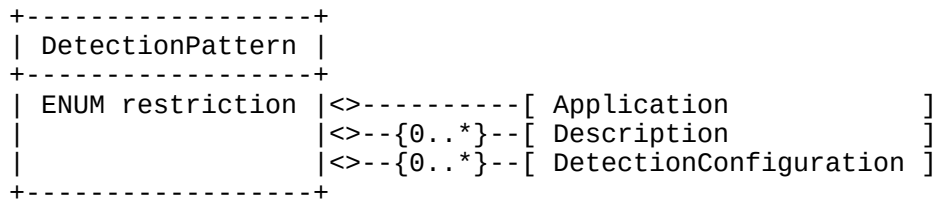

Figure 16: The DetectionPattern Class

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 Method class has one attribute:

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

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.

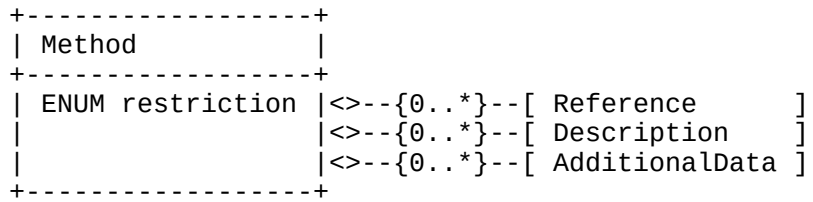

Figure 17: The Method Class

The Method class is composed of three aggregate classes.

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 one attribute:

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

3.13.1. Reference Class

The Reference class is a reference to 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.

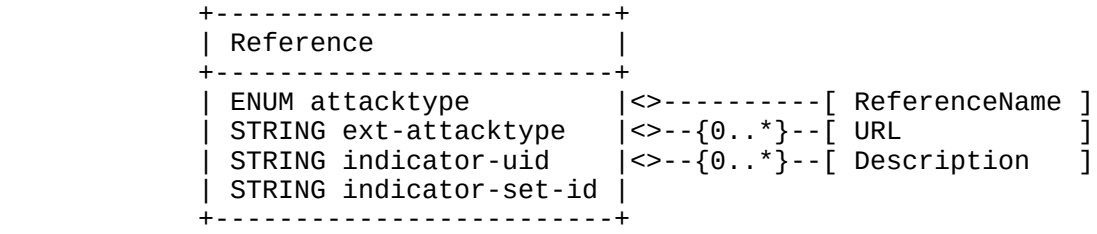

Figure 18: The Reference Class

The aggregate classes that constitute Reference:

ReferenceName
: One. ML_STRING. Name of the reference.

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.

The Reference class has 4 attributes.

attacktype
Optional. ENUM. TODO.

ext-attacktype
Optional. STRING. A mechanism by which to extend the Attack Type.

indicator-uid
Optional. STRING. See Section 3.3.2.

indicator-set-id
Optional. STRING. See Section 3.3.2.

3.14. Assessment Class

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

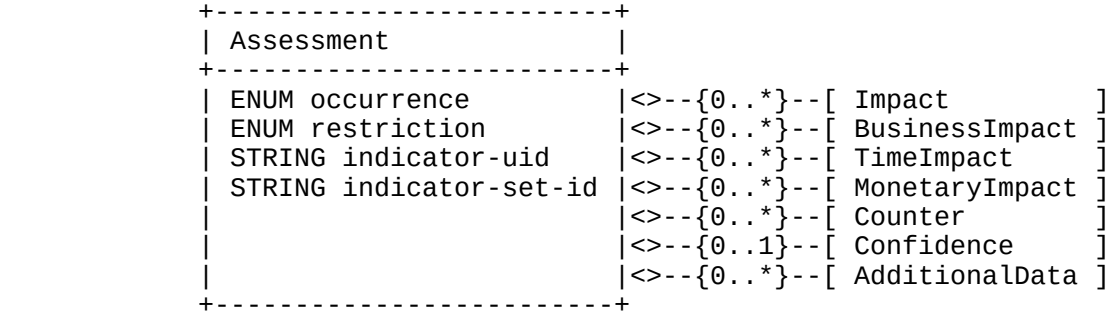

Figure 19: Assessment Class

The aggregate classes that constitute Assessment are:

Impact
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.

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

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. This attribute is defined in Section 3.2.

indicator-uid
Optional. STRING. See Section 3.3.2.

indicator-set-id
Optional. STRING. See Section 3.3.2.

3.14.1. Impact Class

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

This class is based on [RFC4765].

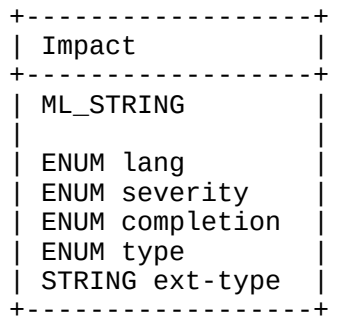

Figure 20: Impact Class

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

The Impact class has five attributes:

lang
: Optional. ENUM. A valid language code per [RFC4646] constrained by the definition of "xs:language". The interpretation of this code is described in 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 malicious activity into incident categories. The permitted values are shown below. The default value is "other".

1. admin. Administrative privileges were attempted.

2. dos. A denial of service was attempted.

3. file. An action that impacts the integrity of a file or database was attempted.

4. info-leak. An attempt was made to exfiltrate information.

5. misconfiguration. An attempt was made to exploit a misconfiguration in a system.

6. policy. Activity violating site's policy was attempted.

7. recon. Reconnaissance activity was attempted.

8. social-engineering. A social engineering attack was attempted.

9. user. User privileges were attempted.

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

11. ext-value. An escape value used to extend this attribute. See Section 5.1.

ext-type
Optional. STRING. A means by which to extend the type attribute. See Section 5.1.

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.

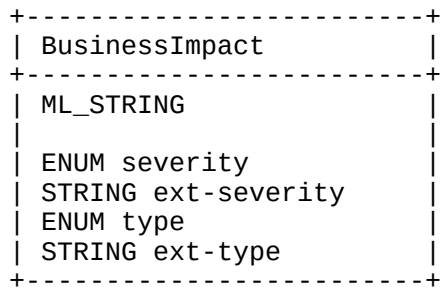

Figure 21: BusinessImpact Class

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

The BusinessImpact class has four attributes:

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".

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.

ext-severity
Optional. STRING. A means by which to extend the severity attribute. See Section 5.1.

type
Required. ENUM. Characterizes the effect this incident had on the business.Classifies the malicious activity into incident

categories. The permitted values are shown below. There is no default value.

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

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

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

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

5. loss-financial. Money or services were stolen.

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

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

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

9. legal. Incident resulted in legal or regulatory action

10. ext-value. An escape value used to extend this attribute. See Section 5.1.

ext-type
Optional. STRING. A means by which to extend the type attribute. See Section 5.1.

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.

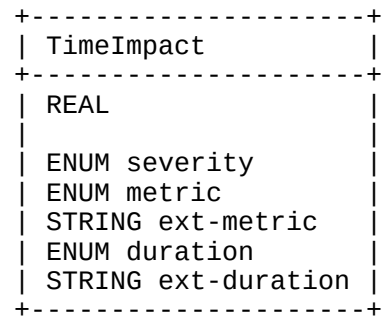

Figure 22: TimeImpact Class

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.

  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.

ext-metric
Optional. STRING. A means by which to extend the metric attribute. See Section 5.1.

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".

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.

ext-duration
Optional. STRING. A means by which to extend the duration attribute. See Section 5.1.

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.

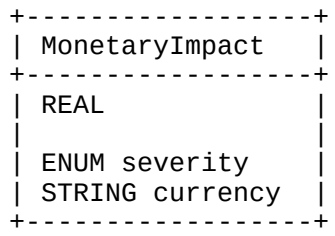

Figure 23: MonetaryImpact Class

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].

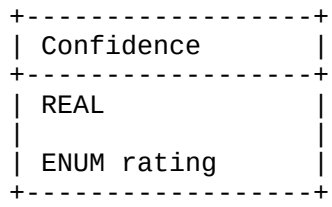

Figure 24: Confidence Class

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.

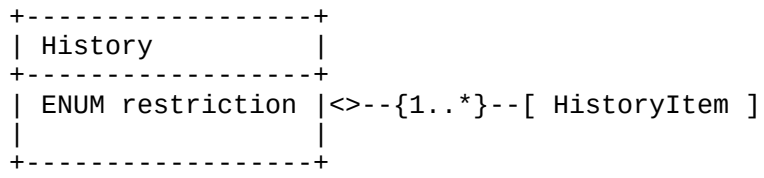

Figure 25: The History Class

The class that constitutes History is:

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

The History class has one attribute:

restriction
Optional. ENUM. This attribute is defined in Section 3.2. The default value is "default".

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.

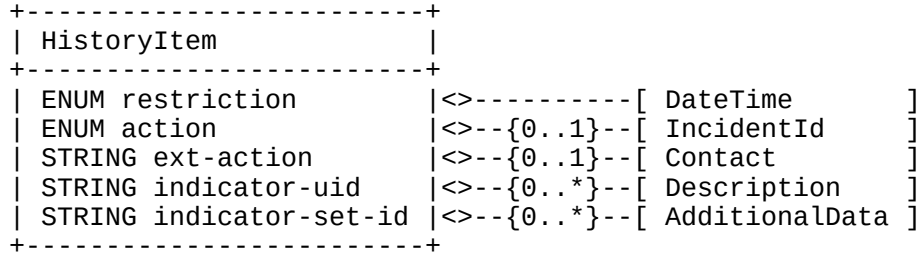

Figure 26: HistoryItem Class

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.

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 category 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.

indicator-uid
Optional. STRING. See Section 3.3.2.

indicator-set-id
Optional. STRING. 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.

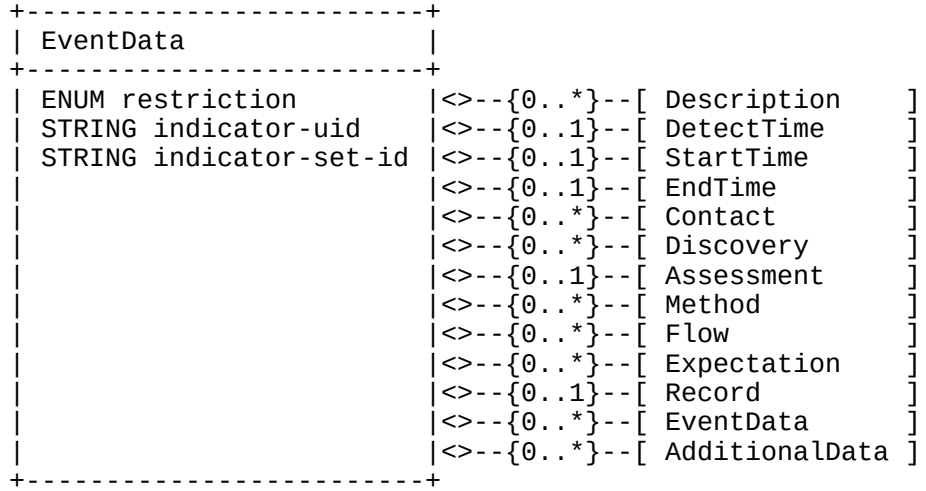

Figure 27: The EventData Class

The aggregate classes that constitute EventData are:

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

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

StartTime
: Zero or one. The time the event started.

EndTime
: Zero or one. The time the event ended.

Contact
: Zero or more. 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 two attributes:

restriction
Optional. ENUM. This attribute is defined in Section 3.2. The default value is "default".

indicator-uid
Optional. STRING. See Section 3.3.2.

indicator-set-id
Optional. STRING. 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 28.

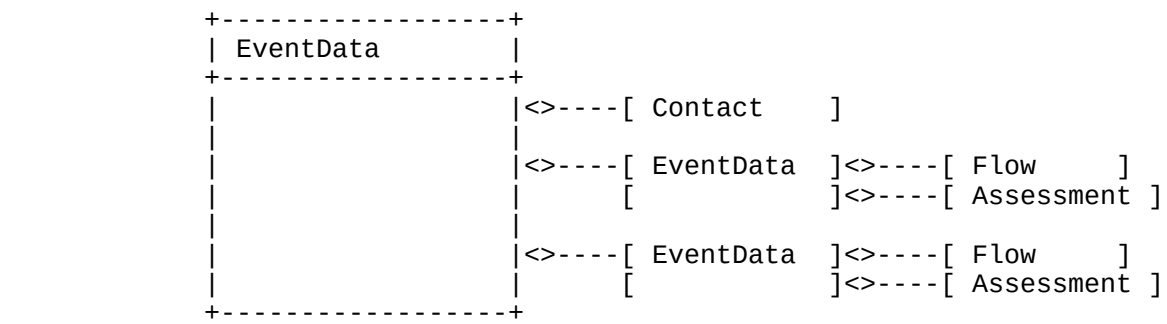

Figure 28: Recursion in the EventData Class

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.

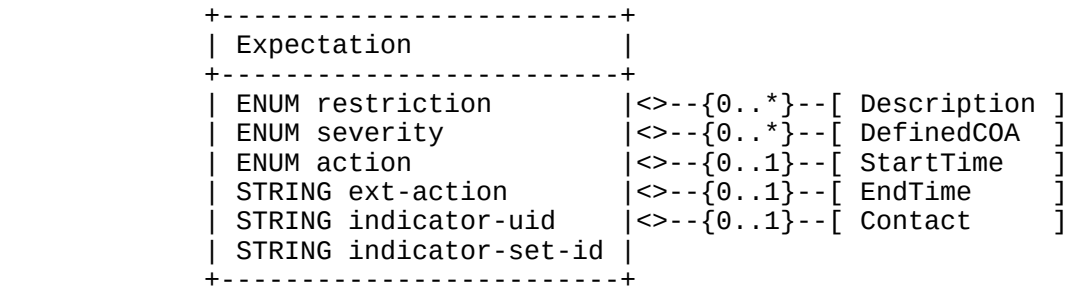

Figure 29: The Expectation Class

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. This attribute is defined in Section 3.2. The default value is "default".

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".

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

13. rebuild-asset. Reinstall the operating system and applications on an asset.

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

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

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

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

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

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

20. ext-value. An escape value used to extend this attribute. See Section 5.1.

ext-action
Optional. STRING. A means by which to extend the action attribute. See Section 5.1.

indicator-uid
Optional. STRING. See Section 3.3.2.

indicator-set-id
Optional. STRING. See Section 3.3.2.

3.18. Flow Class

The Flow class groups related the source and target hosts.

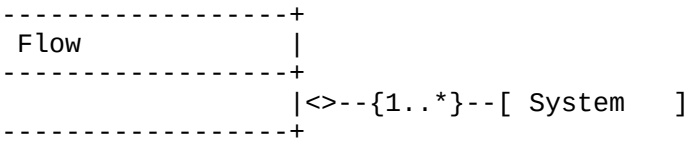

Figure 30: The Flow Class

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.

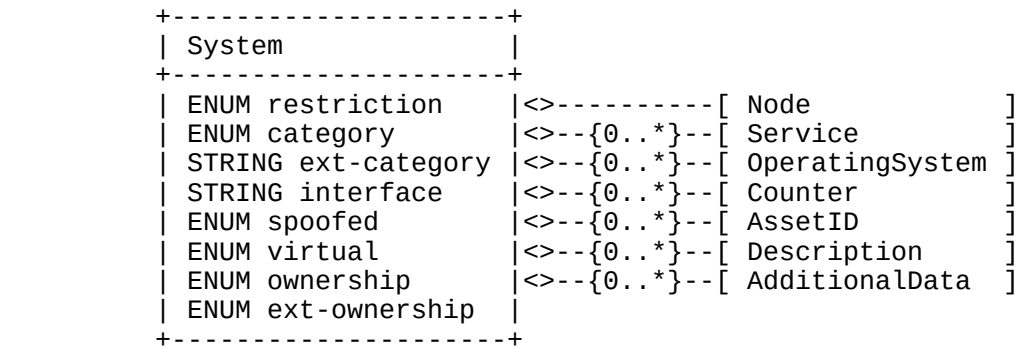

Figure 31: The System Class

The aggregate classes that constitute System are:

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

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

OperatingSystem
Zero or 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 eight attributes:

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

category
Optional. ENUM. Classifies the role the host or network played in the incident. The possible values are:

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

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

3. watchlist-source. The source of the event was on a watchlist.

4. watchlist-target. The target of the event was on a watchlist.

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

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

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

8. ext-value. An escape value used to extend this attribute. See Section 5.1.

ext-category
Optional. STRING. A means by which to extend the category attribute. See Section 5.1.

indicator-set-id
Optional. STRING. See Section 3.3.2.

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. 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.

ext-ownership
Optional. STRING. A means by which to extend the ownership attribute. See Section 5.1.

3.20. Node Class

The Node class names an asset or network.

This class was derived from [RFC4765].

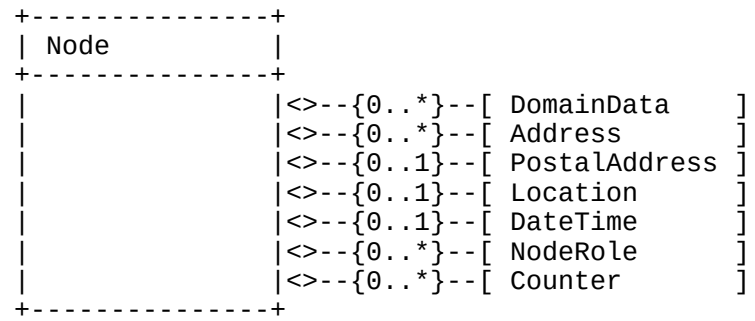

Figure 32: The Node Class

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 one. 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).

NodeRole
: Zero or more. The intended purpose of the Node.

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].

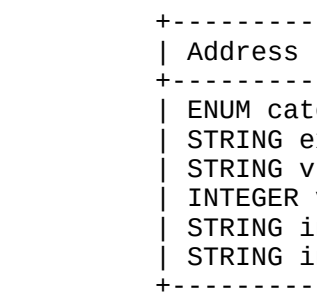

Figure 33: The Address Class

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".

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

10. mac. Media Access Control (MAC) address

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.

ext-category
Optional. STRING. A means by which to extend the category attribute. See Section 5.1.

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.

indicator-uid
Optional. STRING. See Section 3.3.2.

indicator-set-id
Optional. STRING. See Section 3.3.2.

3.20.2. NodeRole Class

The NodeRole class describes the intended function performed by a particular host.

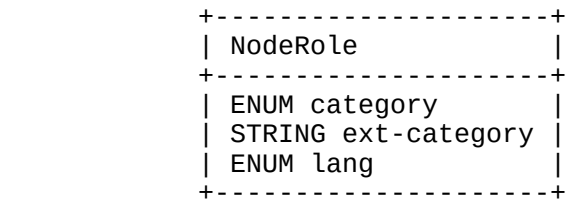

Figure 34: The NodeRole Class

The NodeRole class has three attributes:

category
Required. ENUM. Functionality provided by a node.

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. messaging. Messaging server (e.g., NNTP, IRC, IM)
12. streaming. Streaming-media server
13. voice. Voice server (e.g., SIP, H.323)
14. file. File server (e.g., SMB, CVS, AFS)
15. ftp. FTP server
16. p2p. Peer-to-peer node
17. name. Name server (e.g., DNS, WINS)
18. directory. Directory server (e.g., LDAP, finger, whois)
19. credential. Credential server (e.g., domain controller, Kerberos)
20. print. Print server
21. application. Application server
22. database. Database server
23. backup. Backup server
24. dhcp. DHCP server
25. infra. Infrastructure server (e.g., router, firewall, DHCP)

26. infra-firewall. Firewall

27. infra-router. Router

28. infra-switch. Switch

29. camera. Camera server

30. proxy. Proxy server

31. remote-access. Remote access server

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

33. virtualization. Server running virtual machines

34. pos. Point-of-sale device

35. scada. Supervisory control and data acquisition system

36. scada-supervisory. Supervisory system for a SCADA

37. ext-value. An escape value used to extend this attribute. See Section 5.1.

ext-category
Optional. STRING. A means by which to extend the category attribute. See Section 5.1.

lang
Optional. ENUM. A valid language code per [RFC4646] constrained by the definition of "xs:language". 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.

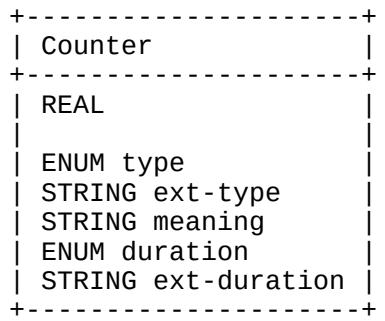

Figure 35: The Counter Class

The Counter class has five attribute:

type
Required. ENUM. Specifies the units of the element content.

1. byte. Count of bytes.

2. packet. Count of packets.

3. flow. Count of network flow records.

4. session. Count of sessions.

5. alert. Count of notifications generated by another system (e.g., IDS or SIM).

6. message. Count of messages (e.g., mail messages).

7. event. Count of events.

8. host. Count of hosts.

9. site. Count of site.

10. organization. Count of organizations.

11. ext-value. An escape value used to extend this attribute. See Section 5.1.

ext-type
Optional. STRING. A means by which to extend the type attribute. See Section 5.1.

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 rather than a count over the entire event. In that case, this attribute specifies the denominator of the rate (where the type attribute specified the nominator). The possible values of this attribute are defined in Section 3.14.3

ext-duration
Optional. STRING. A means by which to extend the duration attribute. See Section 5.1.

3.21. DomainData Class

...TODO...

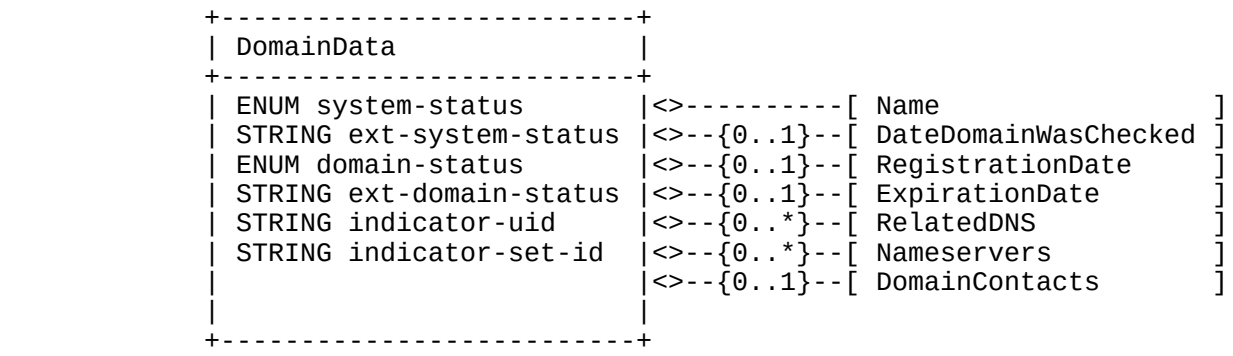

Figure 36: The DomainData Class

The aggregate classes that constitute DomainData are:

Name
One. ML_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. ...TODO...

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 six attribute:

system-status
Required. ENUM. Assesses the domain's involvement in the event.

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.

ext-system-status
Optional. STRING. A means by which to extend the system-status attribute. See Section 5.1.

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].

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. ... TODO -- RFC 5901 has this but doesn't describe it ...

10. unknown. The domain has an unknown status.

11. ext-value. An escape value used to extend this attribute. See Section 5.1.

ext-domain-status
Optional. STRING. A means by which to extend the system-status attribute. See Section 5.1.

indicator-uid
Optional. STRING. See Section 3.3.2.

indicator-set-id
Optional. STRING. See Section 3.3.2.

3.21.1. RelatedDNS

...TODO...

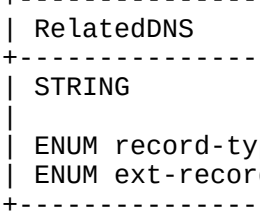

Figure 37: The RelatedDNS Class

3.21.2. Nameservers Class

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

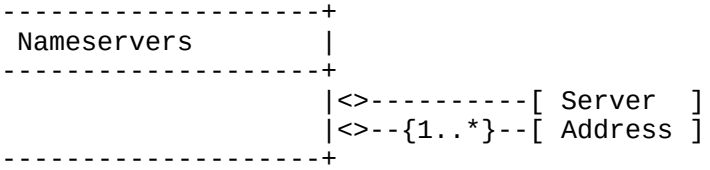

Figure 38: The Nameservers Class

The aggregate classes that constitute Nameservers are:

Server
: One. ML_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.

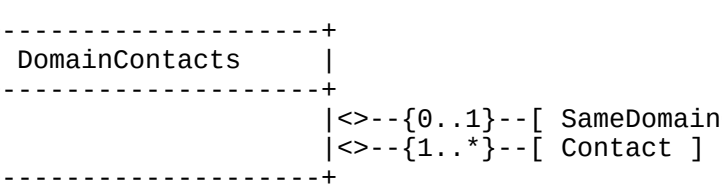

Figure 39: The DomainContacts Class

The aggregate classes that constitute DomainContacts are:

SameDomainContact
: Zero or one. ML_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].

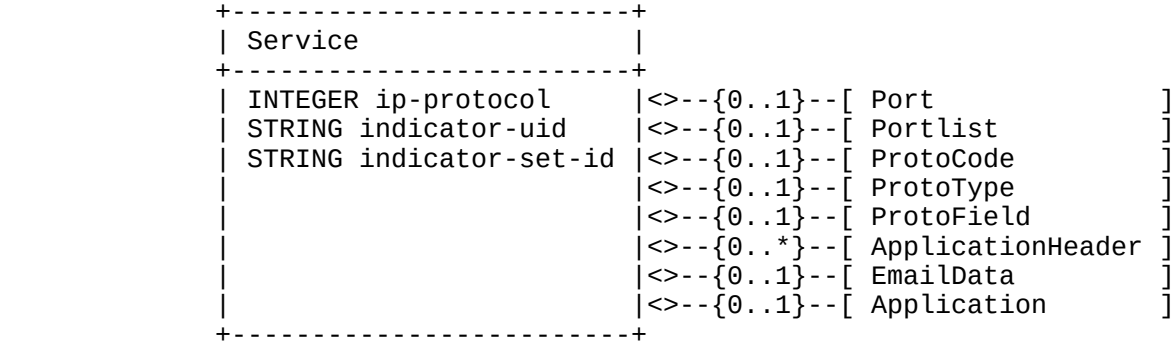

Figure 40: The Service Class

The aggregate classes that constitute Service are:

Port
Zero or one. INTEGER. A port number.

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

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 3.22.1.

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 3.22.2.

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 three attributes:

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

indicator-uid
Optional. STRING. See Section 3.3.2.

indicator-set-id
Optional. STRING. See Section 3.3.2.

3.22.1. ApplicationHeader Class

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

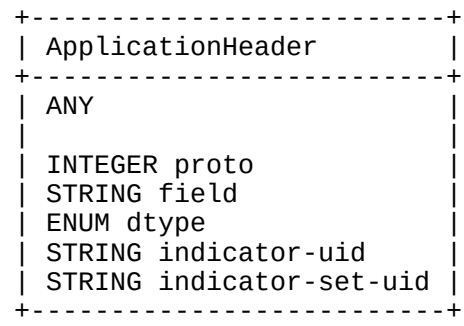

Figure 41: The ApplicationHeader Class

The ApplicationHeader class has five attributes:

proto
: Required. INTEGER. The IANA assigned port number per [IANA.Ports] 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".

  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.

ext-dtype
Optional. STRING. A means by which to extend the dtype attribute. See Section 5.1.

indicator-uid
Optional. STRING. See Section 3.3.2.

indicator-set-id
Optional. STRING. See Section 3.3.2.

3.22.2. Application Class

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

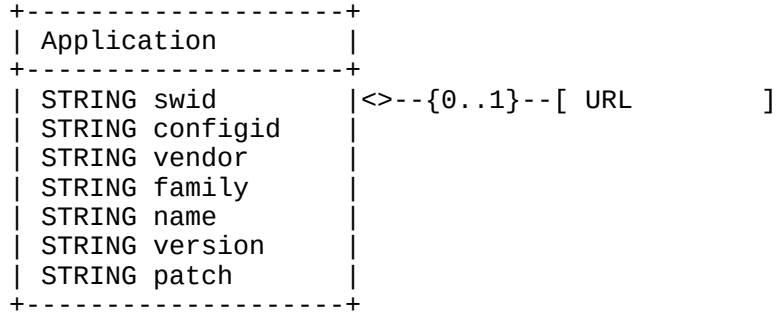

Figure 42: The Application Class

The aggregate class that constitute Application is:

URL
Zero or one. URL. A URL describing the application.

The Application class has seven attributes:

swid
Optional. STRING. An identifier that can be used to reference this software, where the default value is "0".

configid
Optional. STRING. An identifier that can be used to reference a particular configuration of this software, where the default value is "0".

vendor
Optional. STRING. Vendor name of the software.

family
Optional. STRING. Family of the software.

name
Optional. STRING. Name of the software.

version
Optional. STRING. Version of the software.

patch
Optional. STRING. Patch or service pack level of the software.

3.23. OperatingSystem Class

The OperatingSystem class describes the operating system running on a System. The definition is identical to the Application class (Section 3.22.2).

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.

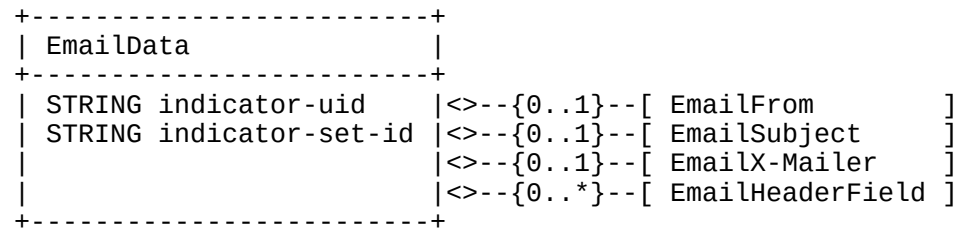

Figure 43: EmailData Class

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 3.22.1. The attributes of EmailHeaderField MUST be set as follows: proto="25" and dtype="string". The name of the email header field MUST be set in the field attribute.

The EmailData class has two attributes:

indicator-uid
: Optional. STRING. See Section 3.3.2.

indicator-set-id
: Optional. STRING. 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.

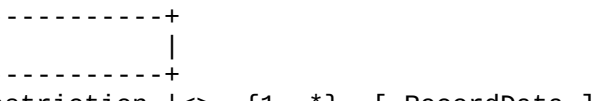

Figure 44: Record Class

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 one attribute:

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

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.

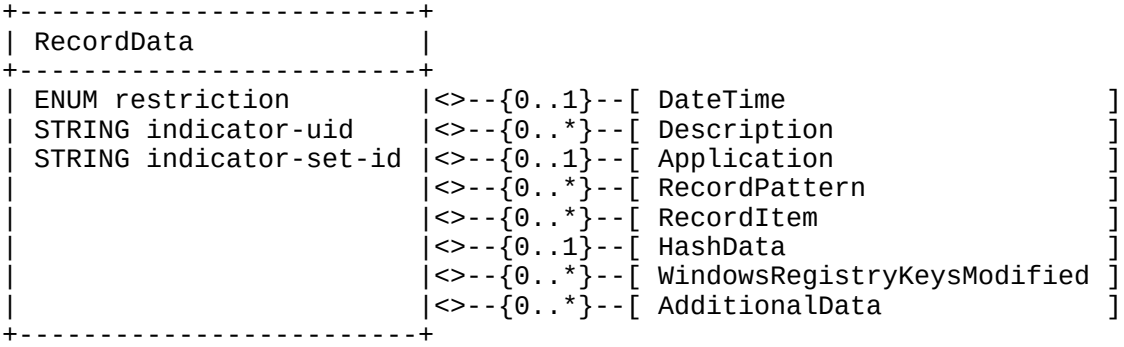

Figure 45: The RecordData Class

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.

HashData
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 attribute:

restriction
Optional. ENUM. See Section 3.3.1.

indicator-uid
Optional. STRING. See Section 3.3.2.

indicator-set-id
Optional. STRING. 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.

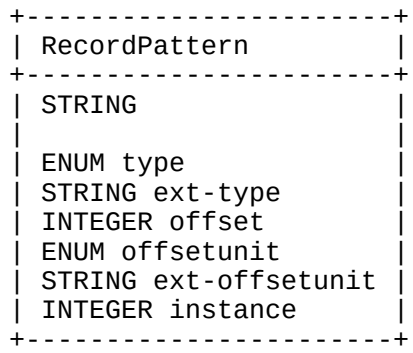

Figure 46: The RecordPattern Class

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".

1. regex. regular expression, per Appendix F of [W3C.SCHEMA.DTYPES].

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.

ext-type
Optional. STRING. A means by which to extend the type attribute. See Section 5.1.

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".

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.

ext-offsetunit
Optional. STRING. A means by which to extend the offsetunit attribute. See Section 5.1.

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].

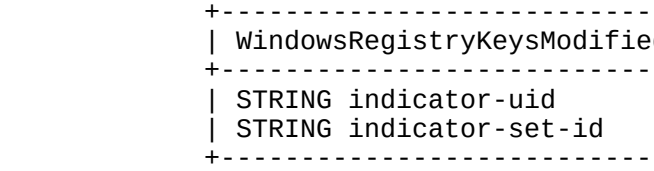

Figure 47: The WindowsRegistryKeysModified Class

The aggregate class that constitutes the WindowsRegistryKeysModified class is:

Key
One or many. The Window registry key.

The WindowsRegistryKeysModified class has two attributes:

indicator-uid
Optional. STRING. See Section 3.3.2.

indicator-set-id
Optional. STRING. 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.

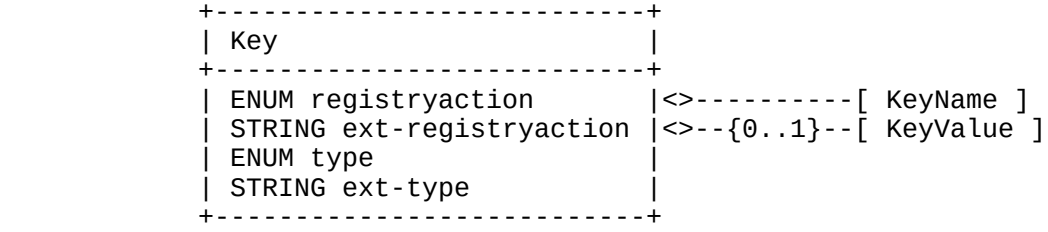

Figure 48: The Key Class

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 four attributes:

registryaction
Optional. ENUM. The type of action taken on the registry key.

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. External value.

ext-registryaction
Optional. A means by which to extend the registryaction attribute. See Section 5.1.

type
Optional. TODO.

1. watchlist. Registry key information that is provided in a watchlist.

2. ext-value. Registry key information from an external source.

ext-type
Optional. A means by which to extend the type attribute. See Section 5.1.

indicator-uid
Optional. STRING. See Section 3.3.2.

indicator-set-id
Optional. STRING. See Section 3.3.2.

3.27. HashData Class

The HashData class describes files, file hashes, ... TODO ...the hash and signature details that are needed for providing context for indicators.

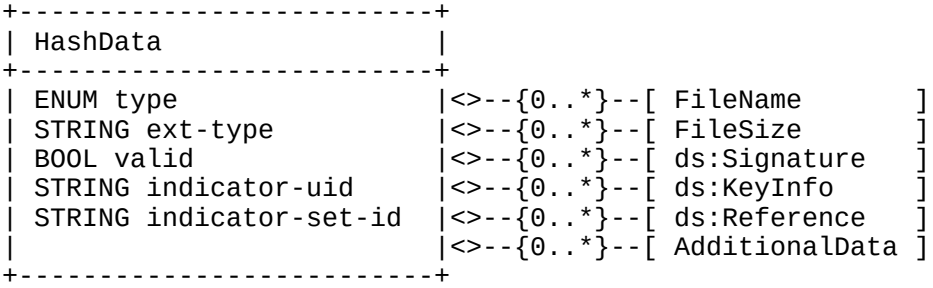

Figure 49: The HashData Class

The aggregate classes that constitutes HashData are:

FileName
Zero or more. ML_STRING. The name of the file.

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

ds:Signature
Zero or more.

ds:KeyInfo
Zero or more.

ds:Reference
Zero or more. The algorithm identification and value of a hash computed over a file. This element is defined in [RFC3275]. Refer to RFC 5901.

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

The HashData class has five attributes:

type
Optional. ENUM. The Hash Type.

1. PKI-email-ds. PKI email digital signature.

2. PKI-file-ds. PKI file digital signature.

3. PKI-email-ds_watchlist. Watchlist of PKI email digital signatures.

4. PKI-file-ds_watchlist. Watchlist of PKI file digital signatures.

5. PGP-email-ds. PGP email digital signature.

6. PGP-file-ds. PGP file digital signature.

7. PGP-email-ds-watchlist. Watchlist of PGP email digital signatures.

8. PGP-file-ds-watchlist. Watchlist of PGP file digital signatures

9. file-hash. A file hash.

10. email-hash. An email hash.

11. file-hash-watchlist. Watchlist of file hashes

12. email-hash-watchlist. Watchlist of email hashes

13. ext-value. An escape value used to extend this attribute. See Section 5.1.

ext-type
Optional. STRING. A means by which to extend the type attribute. See Section 5.1.

valid
Optional. BOOLEAN. Indicates if the signature or hash is valid.

indicator-uid
Optional. STRING. See Section 3.3.2.

indicator-set-id
Optional. STRING. See Section 3.3.2.

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. If UTF-8 encoding is not used, the character encoding MUST also be explicitly specified. 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:

- 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.

- The IODEF-Document@lang and MLStringType@lang attributes are declared as an "xs: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 [RFC4646].

- The MonetaryImpact@currency attribute is declared as an "xs:string", but the list of valid values as defined in [ISO4217].

- All of the aggregated classes Contact and EventData are optional in the schema, but at least one of these aggregated classes MUST be present.

- 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.

- The Confidence@rating attribute determines whether the element content of Confidence should be empty.

- The Address@type attribute determines the format of the element content.

- The attributes AdditionalData@dtype and RecordItem@dtype derived from iodef:ExtensionType determine the semantics and formatting of the element content.

- Symmetry in the enumerated ports of a Portlist class is required between sources and targets. See Section 3.22.

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:

- Renames the Service@ip_protocol attribute to @ip-protocol.

- Removes the Node/NodeName in favor of representing domain names with Node/DomainData/Name. Node/DataTime was also removed so that Node/DomainData/DateDomainWasChecked can represent the time at which the name to address resolution occured.

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 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

The data model supports a means by which to add new enumerated values to an attribute. For each attribute that supports this extension technique, there is a corresponding attribute in the same element whose name is identical, less 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 values. This particular value serves as an escape sequence 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 Impact class would look as follows:

```
 <Impact 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".

5.2. 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
```

6. Internationalization Issues

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

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@lang) 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 [RFC4646].

The data model supports multiple translations of free-form text. In the places where free-text is used for descriptive purposes, the given class always has a one-to-many cardinality to its parent (e.g., Description class). 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.

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 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?
          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
            </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: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:annotation>
     <xs:documentation>
        Incident Object Description Exchange Format v2.0, RFC5070-bis
     </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 name="lang"
                       type="xs:language" use="required"/>
         <xs:attribute name="formatid"
                       type="xs:string"/>
       </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:ReportTime"/>
          <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">
          <xs:simpleType>
            <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:attribute>
        <xs:attribute name="ext-purpose"
                      type="xs:string" use="optional"/>
        <xs:attribute name="lang"
                      type="xs:language"/>
        <xs:attribute name="restriction"
                      type="iodef:restriction-type" default="private"/>
        <xs:attribute name="indicator-uid"
                      type="xs:string" use="optional"/>
        <xs:attribute name="indicator-set-id"
                      type="xs:string" use="optional"/>
      </xs:complexType>
    </xs:element>
  <!--
   ==========================================================================
   ==  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:extension>
      </xs:simpleContent>
    </xs:complexType>

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

```
    </xs:element>

  <!--
   ================================================================
   ==  AlternativeID class                                       ==
   ================================================================
  -->
    <xs:element name="AlternativeID">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="iodef:IncidentID"
                      maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="restriction"
                      type="iodef:restriction-type"/>
      </xs:complexType>
    </xs:element>
  <!--
   ================================================================
   ==  RelatedActivity class                                     ==
   ================================================================
  -->
    <xs:element name="RelatedActivity">
      <xs:complexType>
        <xs:sequence>
          <xs: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: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: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: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"/>
          <xs:element ref="iodef:ContactTitle"
                      minOccurs="0"/>
          <xs:element ref="iodef:Description"
                      minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:RegistryHandle"
                      minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:PostalAddress"
                      minOccurs="0"/>
          <xs:element ref="iodef:Email"
                      minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:Telephone"
                      minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:Fax"
                      minOccurs="0"/>
          <xs:element ref="iodef:Timezone"
                      minOccurs="0"/>
          <xs:element ref="iodef:Contact"
                      minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:AdditionalData"
                      minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="role" use="required">
          <xs:simpleType>
            <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="le"/>
            <xs:enumeration value="vendor"/>
            <xs:enumeration value="ext-value"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="ext-role"
                    type="xs:string" use="optional"/>
      <xs:attribute name="type" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="person"/>
            <xs:enumeration value="organization"/>
            <xs:enumeration value="ext-value"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="ext-type"
                    type="xs:string" use="optional"/>
      <xs:attribute name="restriction"
                    type="iodef:restriction-type"/>
    </xs:complexType>
  </xs:element>
  <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">
            <xs:simpleType>
              <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="internic"/>
                <xs:enumeration value="apnic"/>
                <xs:enumeration value="arin"/>
                <xs:enumeration value="lacnic"/>
                <xs:enumeration value="ripe"/>
                <xs:enumeration value="afrinic"/>
```

```
                  <xs:enumeration value="local"/>
                  <xs:enumeration value="ext-value"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="ext-registry"
                          type="xs:string" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

  <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="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: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 name="DefinedCOA"
                      type="iodef:MLStringType"
                      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="action"
                      type="iodef:action-type" use="required"/>
        <xs:attribute name="ext-action"
        type="xs:string" use="optional"/>
        <xs:attribute name="indicator-uid"
```

```
                        type="xs:string" use="optional"/>
          <xs:attribute name="indicator-set-id"
                        type="xs:string" use="optional"/>
        </xs:complexType>
      </xs:element>
    <!--
     ====================================================================
     ==  Expectation class                                             ==
     ====================================================================
    -->
      <xs:element name="Expectation">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="iodef:Description"
                        minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="DefinedCOA"
                        type="iodef:MLStringType"
                        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="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="indicator-uid"
                        type="xs:string" use="optional"/>
          <xs:attribute name="indicator-set-id"
                        type="xs:string" 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"
                      use="optional" default="unknown">
          <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="idps"/>
              <xs:enumeration value="siem"/>
              <xs:enumeration value="av"/>
              <xs:enumeration value="file-integrity"/>
              <xs:enumeration value="third-party-monitoring"/>
              <xs:enumeration value="os-log"/>
              <xs:enumeration value="application-log"/>
              <xs:enumeration value="device-log"/>
              <xs:enumeration value="network-flow"/>
              <xs:enumeration value="investigation"/>
              <xs:enumeration value="internal-notification"/>
              <xs:enumeration value="external-notification"/>
              <xs:enumeration value="unknown"/>
              <xs:enumeration value="ext-value"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="ext-source"
                      type="xs:string" use="optional"/>
        <xs:attribute name="restriction"
                      type="iodef:restriction-type"/>
      </xs:complexType>
    </xs:element>

    <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: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:complexType>
    </xs:element>
  <!--
   ====================================================================
   ==  Reference class                                               ==
   ====================================================================
  -->
    <xs:element name="Reference">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ReferenceName"
                      type="iodef:MLStringType"/>
          <xs:element ref="iodef:URL"
                      minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:Description"
                      minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="indicator-uid"
                      type="xs:string" use="optional"/>
        <xs:attribute name="indicator-set-id"
                      type="xs:string" use="optional"/>
        <!-- Adding in Attack Type -->
        <xs:attribute name="attacktype" type="att-type"
                      use="required">
        </xs:attribute>
        <xs:attribute name="ext-attacktype"
          type="xs:string" use="optional"/>
      </xs:complexType>
```

```
    </xs:element>

  <!--
   ================================================================
   ==  Assessment class                                          ==
   ================================================================
  -->
    <xs:element name="Assessment">
      <xs:complexType>
        <xs:sequence>
          <xs:choice maxOccurs="unbounded">
            <xs:element ref="iodef:Impact"/>
            <xs:element ref="iodef:BusinessImpact"/>
            <xs:element ref="iodef:TimeImpact"/>
            <xs:element ref="iodef:MonetaryImpact"/>
          </xs:choice>
          <xs:element ref="iodef:Counter"
                      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="indicator-uid"
                      type="xs:string" use="optional"/>
        <xs:attribute name="indicator-set-id"
                      type="xs:string" use="optional"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="Impact">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="iodef:MLStringType">
            <xs:attribute name="severity"
                          type="iodef:severity-type"/>
            <xs:attribute name="completion">
              <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                  <xs:enumeration value="failed"/>
```

```
                <xs:enumeration value="succeeded"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
          <xs:attribute name="type"
                        use="optional" default="unknown">
            <xs:simpleType>
              <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="admin"/>
                <xs:enumeration value="dos"/>
                <xs:enumeration value="extortion"/>
                <xs:enumeration value="file"/>
                <xs:enumeration value="info-leak"/>
                <xs:enumeration value="misconfiguration"/>
                <xs:enumeration value="recon"/>
                <xs:enumeration value="policy"/>
                <xs:enumeration value="social-engineering"/>
                <xs:enumeration value="user"/>
                <xs:enumeration value="unknown"/>
                <xs:enumeration value="ext-value"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
          <xs:attribute name="ext-type"
                        type="xs:string" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="BusinessImpact">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="iodef:MLStringType">
          <xs:attribute name="severity"
                        use="optional">
            <xs:simpleType>
              <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:attribute>
          <xs:attribute name="ext-severity"
                        type="xs:string" use="optional"/>
```

```
            <xs:attribute name="type"
                          use="optional">
              <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                  <xs:enumeration value="breach-proprietary"/>
                  <xs:enumeration value="breach-privacy"/>
                  <xs:enumeration value="loss-of-integrity"/>
                  <xs:enumeration value="loss-of-service" />
                  <xs:enumeration value="loss-financial"/>
                  <xs:enumeration value="degraded-reputation"/>
                  <xs:enumeration value="asset-damage"/>
                  <xs:enumeration value="asset-manipulation"/>
                  <xs:enumeration value="legal"/>
                  <xs:enumeration value="ext-value"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="ext-type"
                          type="xs:string" use="optional"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>

    <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">
              <xs:simpleType>
                <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:attribute>
            <xs:attribute name="ext-metric"
                          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: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">
          <xs:simpleType>
            <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>
        </xs:attribute>
      </xs:complexType>
    </xs:element>
  <!--
   ================================================================
   == 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: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="indicator-uid"
                      type="xs:string" use="optional"/>
        <xs:attribute name="indicator-set-id"
                      type="xs:string" use="optional"/>
      </xs:complexType>
    </xs:element>
  <!--
   ========================================================================
   ==  Flow class                                                        ==
   ========================================================================
  -->
    <!-- Added System unbounded for use only when the source or
         target watchlist is in use, otherwise only one system entry
         is expected.
      -->
    <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: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="category">
          <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="source"/>
              <xs:enumeration value="target"/>
              <!-- CHANGE - adding two new values to cover
                   watchlist groups -->
              <xs:enumeration value="watchlist-source"/>
              <xs:enumeration value="watchlist-target"/>
              <xs:enumeration value="intermediate"/>
              <xs:enumeration value="sensor"/>
              <xs:enumeration value="infrastructure"/>
              <xs:enumeration value="ext-value"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <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">
          <xs:simpleType>
            <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>
        </xs:attribute>
        <xs:attribute name="ext-ownership"
                      type="xs:string" use="optional"/>
      </xs:complexType>
    </xs:element>
  <!--
   ================================================================
   == 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"/>
          <xs:element ref="iodef:NodeRole"
                      minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:Counter"
                      minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Address">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="category" default="ipv4-addr">
              <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                  <xs:enumeration value="asn"/>
                  <xs:enumeration value="atm"/>
                  <xs:enumeration value="e-mail"/>
                  <xs:enumeration value="mac"/>
```

```
                  <xs: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:attribute>
            <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="indicator-uid"
                          type="xs:string" use="optional"/>
           <xs:attribute name="indicator-set-id"
                          type="xs:string" use="optional"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="Location" type="iodef:MLStringType"/>
    <xs:element name="NodeRole">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="iodef:MLStringType">
            <xs:attribute name="category" use="required">
              <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                  <xs:enumeration value="client"/>
                  <xs:enumeration value="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="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="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="ext-value"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="ext-category"
                          type="xs:string" use="optional"/>
            <xs:attribute name="attacktype" type="att-type"
                          use="optional"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  <!--
   ================================================================
   ==  Service Class                                             ==
   ================================================================
  -->
    <xs:element name="Service">
      <xs:complexType>
        <xs:sequence>
          <xs:choice minOccurs="0">
            <xs:element name="Port"
                        type="xs:integer"/>
            <xs:element name="Portlist"
                        type="iodef:PortlistType"/>
          </xs:choice>
          <xs:element name="ProtoType"
```

```
                    type="xs:integer" minOccurs="0"/>
          <xs:element name="ProtoCode"
                    type="xs:integer" minOccurs="0"/>
          <xs:element name="ProtoField"
                    type="xs:integer" minOccurs="0"/>
          <xs:element name="ApplicationHeader"
                    type="iodef:ApplicationHeaderType"
                    minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="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="indicator-uid"
                    type="xs:string" use="optional"/>
        <xs:attribute name="indicator-set-id"
                    type="xs:string" use="optional"/>
      </xs:complexType>
    </xs:element>
    <xs:simpleType name="PortlistType">
      <xs:restriction base="xs:string">
        <xs:pattern value="\d+(\-\d+)?(,\d+(\-\d+)?)*"/>
      </xs:restriction>
    </xs:simpleType>
  <!--
   ================================================================
   ==  Counter class                                             ==
   ================================================================
  -->
    <xs:element name="Counter">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:double">
            <xs:attribute name="type" use="required">
              <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                  <xs:enumeration value="byte"/>
                  <xs:enumeration value="packet"/>
                  <xs:enumeration value="flow"/>
                  <xs:enumeration value="session"/>
                  <xs:enumeration value="event"/>
                  <xs:enumeration value="alert"/>
                  <xs:enumeration value="message"/>
                  <xs:enumeration value="host"/>
                  <xs:enumeration value="site"/>
                  <xs:enumeration value="organization"/>
                  <xs:enumeration value="ext-value"/>
```

```
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
            <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>

  <!--
   ========================================================================
   ==  EmailData class                                                   ==
   ========================================================================
  -->
   <xs:element name="EmailData">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="EmailFrom"
                     type="iodef:MLStringType" minOccurs="0"/>
         <xs:element name="EmailSubject"
                     type="iodef:MLStringType" minOccurs="0"/>
         <xs:element name="EmailX-Mailer"
                     type="iodef:MLStringType" minOccurs="0"/>
         <xs:element name="EmailHeaderField"
                     type="iodef:ApplicationHeaderType"
                     minOccurs="0"/>
       </xs:sequence>
       <xs:attribute name="indicator-uid"
                     type="xs:string" use="optional"/>
       <xs:attribute name="indicator-set-id"
                     type="xs:string" use="optional"/>
     </xs:complexType>
   </xs:element>

  <!--
   ========================================================================
   ==   DomainData class - from RFC5901                                  ==
   ========================================================================
  -->
  <xs:element name="DomainData">
    <xs:complexType>
```

```
      <xs:sequence>
        <xs:element name="Name"
                    type="iodef:MLStringType" maxOccurs="1" />
        <xs:element name="DateDomainWasChecked"
                    type="xs:dateTime"
                    minOccurs="0" maxOccurs="1" />
        <xs:element name="RegistrationDate"
                    type="xs:dateTime"
                    minOccurs="0" maxOccurs="1" />
        <xs:element name="ExpirationDate"
                    type="xs:dateTime"
                    minOccurs="0" maxOccurs="1" />
        <xs:element name="RelatedDNS"
                    type="iodef:RelatedDNSEntryType"
                    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">
        <xs:simpleType>
          <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:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="ext-system-status"
                    type="xs:string" use="optional"/>
      <xs:attribute name="domain-status">
        <xs:simpleType>
          <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:restriction>
```

```
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="ext-domain-status"
                    type="xs:string" use="optional"/>
      <xs:attribute name="indicator-uid"
                    type="xs:string" use="optional"/>
      <xs:attribute name="indicator-set-id"
                    type="xs:string" use="optional"/>
    </xs:complexType>
  </xs:element>

    <xs:element name="RelatedDNS"
                type="iodef:RelatedDNSEntryType"/>
    <xs:complexType name="RelatedDNSEntryType">
     <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:enumeration value="ext-value"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="ext-record-type"
                    type="xs:string" use="optional"/>
    </xs:extension>
   </xs:simpleContent>
  </xs:complexType>

  <xs:element name="Nameservers">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Server" type="iodef:MLStringType"/>
        <xs:element ref="iodef:Address" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="DomainContacts">
    <xs:complexType>
       <xs:choice>
         <xs:element name="SameDomainContact"
                     type="iodef:MLStringType"/>
         <xs:element ref="iodef:Contact"
                     maxOccurs="unbounded" minOccurs="1"/>
       </xs:choice>
    </xs:complexType>
  </xs:element>


 <!--
  ========================================================================
  ==  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: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:HashInformation"
                    minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:WindowsRegistryKeysModified"
                    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="indicator-uid"
                    type="xs:string" use="optional"/>
        <xs:attribute name="indicator-set-id"
                    type="xs:string" use="optional"/>
      </xs:complexType>
    </xs:element>

    <xs:element name="RecordPattern">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="type" use="required">
              <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                  <xs:enumeration value="regex"/>
                  <xs:enumeration value="binary"/>
                  <xs:enumeration value="xpath"/>
                  <xs:enumeration value="ext-value"/>
                </xs:restriction>
              </xs:simpleType>
```

```
            </xs:attribute>
            <xs:attribute name="ext-type"
                          type="xs:string" use="optional"/>
            <xs:attribute name="offset"
                          type="xs:integer" use="optional"/>
            <xs:attribute name="offsetunit"
                          use="optional" default="line">
              <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                  <xs:enumeration value="line"/>
                  <xs:enumeration value="byte"/>
                  <xs:enumeration value="ext-value"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
            <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:element name="RecordItem"
                type="iodef:ExtensionType"/>
  <!--
   ================================================================
   ==  Class to describe Windows Registry Keys                   ==
   ================================================================
  -->
    <xs:element name="WindowsRegistryKeysModified">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Key" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
            <!-- Allows for the value to be optional for cases
                 such as, the registry key was deleted -->
               <xs:element name="KeyName" type="xs:string"/>
               <xs:element name="Value"
                           type="xs:string" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="registryaction">
              <xs:simpleType>
                <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>
            </xs:attribute>
            <xs:attribute name="ext-registryaction"
                          type="xs:string" use="optional"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="indicator-uid"
                    type="xs:string" use="optional"/>
      <xs:attribute name="indicator-set-id"
                    type="xs:string" use="optional"/>
    </xs:complexType>
    </xs:element>

  <!--
    ========================================================================
    ==  Classes that describe hash types, file information          ==
    ==  with certificate properties and digital signature info      ==
    ==  provided through the W3C digital signature schema           ==
    ==  so it does not need to be maintained here.                  ==
    ========================================================================
   -->
   <xs:element name="HashInformation">
   <xs:complexType>
     <xs:sequence>
         <xs:element name="FileName" type="iodef:MLStringType"
                     minOccurs="0" maxOccurs="unbounded"/>
         <xs:element name="FileSize" type="xs:integer"
                     minOccurs="0" maxOccurs="unbounded"/>
     <!-- CHANGE: Represent file hash information via digsig schema
       and the Reference class.  You may need any of the other classes
       and in particular the KeyInfo (see RFC3275 sect 4.4.4/4.4.5),
       which has been added.  KeyName, KeyValue, SignatureProperties
       classes may be useful, so Signature was added, but you can use
       KeyInfo and Reference directly to avoid some bloat.  -->
         <xs:element ref="ds:Signature"
                     minOccurs="0" maxOccurs="unbounded"/>
         <xs:element ref="ds:KeyInfo"
                     minOccurs="0" maxOccurs="unbounded"/>
         <xs:element ref="ds:Reference"
                     minOccurs="0" maxOccurs="unbounded"/>
         <xs:element ref="iodef:AdditionalData"
                     minOccurs="0" maxOccurs="unbounded"/>
```

```
      </xs:sequence>
      <xs:attribute name="type" use="optional">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="PKI-email-ds"/>
            <xs:enumeration value="PKI-file-ds"/>
            <xs:enumeration value="PKI-email-ds-watchlist"/>
            <xs:enumeration value="PKI-file-ds-watchlist"/>
            <xs:enumeration value="PGP-email-ds"/>
            <xs:enumeration value="PGP-file-ds"/>
            <xs:enumeration value="PGP-email-ds-watchlist"/>
            <xs:enumeration value="PGP-file-ds-watchlist"/>
            <xs:enumeration value="file-hash"/>
            <xs:enumeration value="email-hash"/>
            <xs:enumeration value="file-hash-watchlist"/>
            <xs:enumeration value="email-hash-watchlist"/>
            <!-- QUESTION: Are values needed to differentiate the
                 key information shared when the ds:KeyInfo class
                 is referenced? -->
            <xs:enumeration value="ext-value"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="ext-type"
                    type="xs:string" use="optional"/>
      <xs:attribute name="valid"
                    type="xs:boolean" use="optional" />
      <xs:attribute name="indicator-uid"
                    type="xs:string" use="optional"/>
      <xs:attribute name="indicator-set-id"
                    type="xs:string" use="optional"/>
      <xs:attribute name="restriction"
                    type="iodef:restriction-type"/>
    </xs:complexType>
    </xs:element>

  <!--
   ========================================================================
   ==  Classes that describe software                                    ==
   ========================================================================
  -->
    <xs:complexType name="SoftwareType">
      <xs:sequence>
        <xs:element ref="iodef:URL"
                    minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="swid"
                    type="xs:string" default="0"/>
```

```
      <xs:attribute name="configid"
                    type="xs:string" default="0"/>
      <xs:attribute name="vendor"
                    type="xs:string"/>
      <xs:attribute name="family"
                    type="xs:string"/>
      <xs:attribute name="name"
                    type="xs:string"/>
     <!-- CHANGE: Should UserAgent or HTTPUserAgent fit in
          SoftwareTypes? This is typically intended to mean
          servers, but the category seems more appropriate
          than others.
     -->
      <xs:attribute name="user-agent"
                    type="xs:string"/>
      <xs:attribute name="version"
                    type="xs:string"/>
      <xs:attribute name="patch"
                    type="xs:string"/>
    </xs:complexType>
    <xs:element name="Application"
                type="iodef:SoftwareType"/>
    <xs:element name="OperatingSystem"
                type="iodef:SoftwareType"/>

  <!--
   ================================================================
   == Miscellaneous simple classes                               ==
   ================================================================
  -->
    <xs:element name="Description"
                type="iodef:MLStringType"/>
    <xs:element name="URL"
                type="xs:anyURI"/>
  <!--
   ================================================================
   == 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="lang"
                        type="xs:language" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>

    <xs:complexType name="ExtensionType" mixed="true">
      <xs:sequence>
        <xs:any namespace="##any" processContents="lax"
                minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="dtype"
                    type="iodef:dtype-type" use="required"/>
      <xs:attribute name="ext-dtype"
                    type="xs:string" use="optional"/>
      <xs:attribute name="meaning"
                    type="xs:string"/>
      <xs:attribute name="formatid"
                    type="xs:string"/>
      <xs:attribute name="restriction"
                    type="iodef:restriction-type"/>
    </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="required"/>
      <xs:attribute name="field"
                    type="xs:string" use="required"/>
      <xs:attribute name="dtype"
                    type="iodef:proto-dtype-type"
                    use="required"/>
      <xs:attribute name="indicator-uid"
                    type="xs:string" use="optional"/>
      <xs:attribute name="indicator-set-id"
                    type="xs:string" 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: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="upgrade-software"/>
        <xs:enumeration value="rebuild-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:simpleType name="att-type">
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="c2-server"/>
          <xs:enumeration value="sink-hole"/>
          <xs:enumeration value="malware-distribution"/>
          <xs:enumeration value="phishing"/>
          <xs:enumeration value="spear-phishing"/>
          <xs:enumeration value="recruiting"/>
          <xs:enumeration value="fraudulent-site"/>
          <xs:enumeration value="dns-spoof"/>
          <xs:enumeration value="other"/>
          <xs:enumeration value="unknown"/>
          <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 uses URNs to describe an XML namespace and schema conforming to a registry mechanism described in [RFC3688]

Registration for the IODEF namespace:

- URI: urn:ietf:params:xml:ns:iodef-2.0
- Registrant Contact: See the first author of the "Author’s Address" section of this document.
- XML: None. Namespace URIs do not represent an XML specification.

Registration for the IODEF XML schema:

- URI: urn:ietf:params:xml:schema:iodef-2.0
- Registrant Contact: See the first author of the "Author’s Address" section of this document.

- XML: See the "IODEF Schema" in Section 8 of this document.

11. Acknowledgments

The following groups and individuals, listed alphabetically, contributed substantially to this document and should be recognized for their efforts.

- Kathleen Moriarty, EMC Corporation
- Brian Trammell, ETH Zurich
- Patrick Cain, Cooper-Cain Group, Inc.
- ... 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/>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[RFC4646] Philips, A. and M. Davis, "Tags for Identifying of Languages", RFC 4646, September 2006.

[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.

[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.

[RFC3275] Eastlake, D., Reagle, J., and D. Solo, "(Extensible Markup Language) XML-Signature Syntax and Processing", RFC 3275, March 2002.

[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>.

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.

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