MILE
Internet-Draft
Intended status: Informational
Expires: May 24, 2015

C. Inacio
CMU
D. Miyamoto
UTokyo

November 20, 2014

MILE Implementation Report
draft-ietf-mile-implementreport-01

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 May 24, 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. Other Implementations

6.1. Collaborative Incident Management System

Collaborative Incident Management System (CIMS) is a proof-of-concept system for collaborative incident handling and for the sharing of cyber defence situational awareness information between the participants, developed for the Cyber Coalition 2013 (CC13) exercise organized by NATO. CIMS was implemented based on Request Tracker

(RT), an open source software widely used for handling incident response by many CERTs and CSIRTs.

One of the functionality implemented in CIMS was the ability to import and export IODEF messages in the body of emails. The intent was to verify the suitability of IODEF to achieve the objective of collaborative incident handling. The customized version of RT could be configured to send an email message containing an IODEF message whenever an incident ticket was created, modified or deleted. These IODEF messages would then be imported into other incident handling systems in order to allow participating CSIRTs to use their usual means for incident handling, while still interacting with those using the proof-of-concept CIMS. Having an IODEF message generated for every change made to the incident information in RT (and for the system to allow incoming IODEF email messages to be associated to an existing incident) would in some way allow all participating CSIRTs to actually work on a "common incident ticket", at least at the conceptual level. Of particular importance was the ability for users to exchange information between each other concerning actions taken in the handling of a particular incident, thus creating a sort of common action log, as well as requesting/tasking others to provide information or perform specified action and correlating received responses to the original request or tasking. As well, a specific "profile" was developed to identify a subset of the IODEF classes that would be used during the exercise, in an attempt to channel all users into a common usage pattern of the otherwise flexible IODEF standard.

6.2. n6

n6 is a platform for processing security-related information, developed by NASK, CERT Polska. Its API provides a common and unified way of representing data across the different sources that participate in knowledge management.

n6 exposes a REST-ful API over HTTPS with mandatory authentication via TLS client certificates, to ensure confidential and trustworthy communications. Moreover, it uses an event-based data model for representation of all types of security information.

Each event is represented as a JSON object with a set of mandatory and optional attributes. It also supports alternative output data formats for keeping compatibility with existing systems - IODEF and CSV - although they lack some of the attributes that may be present in the native JSON format.

7. Implementation Guide

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

7.1. 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.

7.2. 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.

8. 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

Dandurand Luc, NATO

Pawel Pawlinski, NASK

9. IANA Considerations

This memo includes no request to IANA.

10. 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.

11. 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
Intended status: Informational
Expires: March 11, 2018

P. Kampanakis
Cisco Systems
M. Suzuki
NICT

September 7, 2017

Incident Object Description Exchange Format Usage Guidance
draft-ietf-mile-iodef-guidance-11

Abstract

The Incident Object Description Exchange Format (IODEF) v2 (RFC7970) defines a data representation that provides a framework for sharing information about computer security incidents commonly exchanged by Computer Security Incident Response Teams (CSIRTs) . Since the IODEF model includes a wealth of available options that can be used to describe a security incident or issue, it can be challenging for security practitioners to develop tools that leverage IODEF for incident sharing. This document provides guidelines for IODEF implementers. It addresses how common security indicators can be represented in IODEF and use-cases of how IODEF is being used. This document aims to make IODEF's adoption by vendors easier and encourage faster and wider adoption of the model by CSIRTs around the world.

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 https://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 March 11, 2018.

Copyright Notice

Table of Contents

1. Introduction

The Incident Object Description Exchange Format (IODEF) v2 [RFC7970] defines a data representation that provides a framework for sharing computer security incident information commonly exchanged by Computer Security Incident Response Teams (CSIRTs). The IODEF data model

consists of multiple classes and data types that are defined in the IODEF XML schema.

The IODEF schema was designed to describe all the possible fields needed in a security incident exchange. Thus, IODEF contains a plethora of data constructs which could make it hard for IODEF implementers to decide which are important. Additionally, in the IODEF schema, there exist multiple fields and classes which do not necessarily need to be used in every possible data exchange. Moreover, some IODEF classes are useful only in rare circumstances. This document tries to address these concerns. It also presents how common security indicators can be represented in IODEF. It points out the most important IODEF classes for an implementer and describes other ones that are not as important. Also, it presents some common pitfalls for IODEF implementers and how to address them. The end goal of this document is to make IODEF's use by vendors easier and encourage wider adoption of the model by CSIRTs around the world.

Section 3 discusses the recommended classes and how an IODEF implementer should choose the classes to implement. Section 4 presents common considerations a practitioner will come across and how to address them. Section 5 goes over some common uses of IODEF.

2. Terminology

The terminology used in this document follows the one defined in [RFC7970] and [RFC7203].

3. Implementation and Use Strategy

It is important for IODEF implementers to distinguish how the IODEF classes will be used in incident information exchanges. It is also important to understand the most common IODEF classes that describe common security incidents or indicators. This section describes the most important classes and factors an IODEF practitioner should take into consideration before using IODEF or designing an implementation.

3.1. Minimal IODEF document

An IODEF document must include at least an Incident class, an xml:lang attribute that defines the supported language and the IODEF version attribute. An Incident must contain a purpose attribute and three mandatory-to-implement elements. These elements are Generation time class that describes the time of the incident, an IncidentID class and at least one Contact class. The structure of the minimal IODEF-Document is shown in Figure 1.

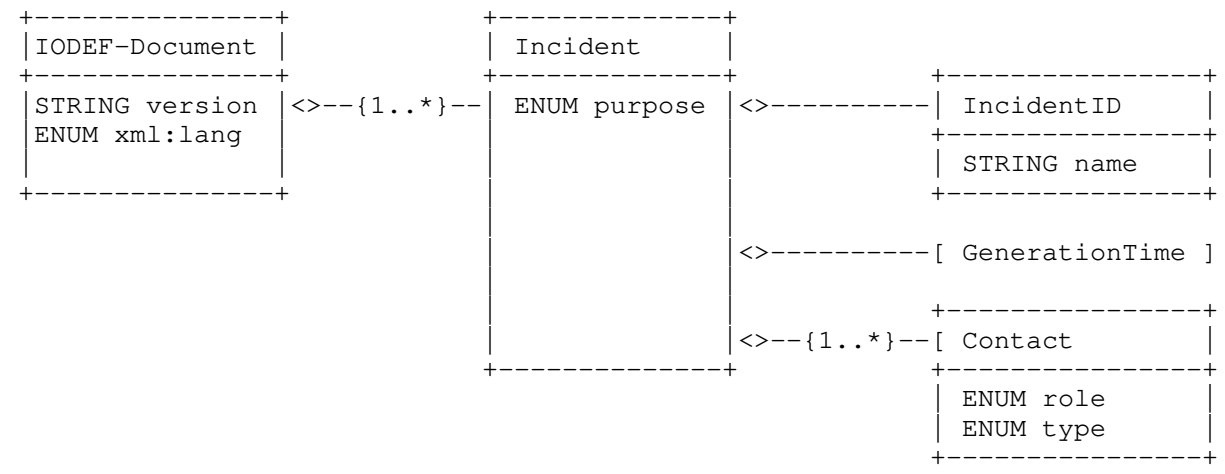

Figure 1: Minimal IODEF-Document class

The IncidentID class must contain at least a name attribute.

In turn, the Contact class requires the type and role attributes, but no elements are required by the IODEF v2 specification. Nevertheless, at least one of the elements in the Contact class, such as an Email class, should be implemented so that the IODEF document is useful.

Section 7.1 of [RFC7970] presents a minimal IODEF document with only the mandatory classes and attributes. Implementers can also refer to Section 7 of [RFC7970] and Appendix B for example IODEF v2 documents.

3.2. Information represented

There is no need for a practitioner to use or implement IODEF classes and fields other than the minimal ones (Section 3.1) and the ones necessary for her use-cases. The implementer should carefully look into the schema and decide which classes to implement (or not).

For example, if we have Distributed Denial of Service (DDoS) as a potential use-case, then the Flow class and its included information are the most important classes to use. The Flow class describes information related to the attacker and victim hosts, which information could help automated filtering or sink-hole operations.

Another potential use-case is malware command and control (c2). After modern malware infects a device, it usually proceeds to connect to one or more c2 servers to receive instructions from its master and potentially exfiltrate information. To protect against such

activity, it is important to interrupt the c2 communication by filtering the activity. IODEF can describe c2 activities using the Flow and the ServiceName classes.

For use-cases where indicators need to be described, the IndicatorData class will be implemented instead of the EventData class.

In summary, an implementer should identify her use-cases and find the classes that are necessary to support in IODEF v2. Implementing and parsing all IODEF classes can be cumbersome in some occasions and unnecessary. Other external schemata can also be used in IODEF to describe incidents or indicators. External schemata should be parsed accordingly only if the implementer's IODEF use-cases require external schema information. But even when an IODEF implementation cannot parse an external schema, the IODEF report can still be valuable to an incident response team. The information can also be useful when shared further with content consumers able to parse this information.

IODEF supports multiple language translations of free-form, ML_STRING text in all classes [RFC7970]. That way, text in Description elements can be translated to different languages by using a translation identifier in the class. Implementers should be able to parse iodef:MLStringType classes and extract only the information relevant to languages of interest.

3.3. IODEF Classes

[RFC7970] contains classes that can describe attack Methods, Events, Incidents, Indicators, how they were discovered and the Assessment of the repercussions for the victim. It is important for IODEF users to know the distinction between these classes in order to decide which ones fulfill their use-cases.

An IndicatorData class depicts a threat indicator or observable that could be used to describe a threat that resulted in an attempted attack. For example, we could see an attack happening but it might have been prevented and not have resulted in an incident or security event. On the other hand, an EventData class usually describes a security event and can be considered as a report of something that took place.

Classes like Discovery, Assessment, Method, and RecoveryTime are used in conjunction with EventData as they related to the incident report described in the EventData. The RelatedActivity class can reference an incident, an indicator or other related threat activity.

While deciding what classes are important for the needed use-cases, IODEF users should carefully evaluate the necessary classes and how these are used in order to avoid unnecessary work. For example, if we want to only describe indicators in IODEF, the implementation of Method or Assessment might not be important.

4. IODEF usage considerations

Implementers need to consider some common, standardized options for their IODEF use strategy.

4.1. External References

The IODEF format includes the Reference class used for externally defined information such as a vulnerability, Intrusion Detection System (IDS) alert, malware sample, advisory, or attack technique. To facilitate the exchange of information, the Reference class was extended to the Enumeration Reference Format [RFC7495]. The Enumeration Reference Format specifies a means to use external enumeration specifications (e.g. CVE) that could define an enumeration format, specific enumeration values, or both. As external enumerations can vary greatly, implementers should only support the ones expected to describe their specific use-cases.

4.2. Extensions

The IODEF data model ([RFC7970]) is extensible. Many attributes with enumerated values can be extended using the "ext-*" prefix. Additional classes can also be defined by using the AdditionalData and RecordItem classes. An extension to the AdditionalData class for reporting Phishing emails is defined in [RFC5901]. Information about extending IODEF class attributes and enumerated values can be found in Section 5 of [RFC7970].

Additionally, IODEF can import existing schemata by using an extension framework defined in [RFC7203]. The framework enables IODEF users to embed XML data inside an IODEF document using external schemata or structures defined by external specifications. Examples include CVE, CVRF and OVAL. [RFC7203] enhances the IODEF capabilities without further extending the data model.

IODEF implementers should not use their own IODEF extensions unless data cannot be represented using existing standards or importing them in an IODEF document using [RFC7203] is not a suitable option.

4.3. Indicator predicate logic

An IODEF [RFC7970] document can describe incident reports and indicators. The Indicator class can include references to other indicators, observables and more classes that contain details about the indicator. When describing security indicators, it is often common to need to group them together in order to form a group of indicators that constitute a security threat. For example, a botnet might have multiple command and control servers. For that reason, IODEF v2 introduced the IndicatorExpression class that is used to add the indicator predicate logic when grouping more than one indicators or observables.

Implementations must be able to parse and apply the Boolean logic offered by an IndicatorExpression in order to evaluate the existence of an indicator. As explained in Section 3.29.5 of [RFC7970] the IndicatorExpression element operator defines the operator applied to all the child element of the IndicatorExpression. If no operator is defined "and" should be assumed. IndicatorExpressions can also be nested together. Child IndicatorExpressions should be treated as child elements of their parent and they should be evaluated first before evaluated with the operator of their parent.

Users can refer to Appendix A for example uses of the IndicatorExpressions in an IODEF v2.

4.4. Disclosure level

Access to information in IODEF documents should be tightly locked since the content may be confidential. IODEF has a common attribute, called "restriction", which indicates the disclosure guideline to which the sender expects the recipient to adhere to for the information represented in the class and its children. That way, the sender can express the level of disclosure for each component of an IODEF document. Appropriate external measures could be implemented based on the restriction level. One example is when Real-time Inter-network Defense (RID) [RFC6545] is used to transfer the IODEF documents, it can provide policy guidelines for handling IODEF documents by using the RIDPolicy class.

The enforcement of the disclosure guidelines is out of scope for IODEF. The recipient of the IODEF document needs to follow the guidelines, but these guidelines themselves do not provide any enforcement measures. For that purpose, implementers should consider appropriate privacy control measures, technical or operational for their implementation.

5. IODEF Uses

IODEF is currently used by various organizations in order to represent security incidents and share incident and threat information between security operations organizations.

5.1. Implementations

In order to use IODEF, tools like IODEF parsers are necessary. [RFC8134] describes a set of IODEF implementations and uses by various vendors and Computer Emergency Readiness Team (CERT) organizations. The document does not specify any specific mandatory to implement (MTI) IODEF classes but provides a list of real world uses. Perl and Python modules (XML::IODEF, Iodef::Pb, iodeflib) are some examples. Moreover, implementers are encouraged to refer to Section 7 of [RFC8134] practical IODEF usage guidelines. [implementations], on the other hand, includes various vendor incident reporting products that can consume and export in IODEF format.

5.2. Inter-vendor and Service Provider Exercise

As an interoperability exercise, in 2013 a limited number of vendors organized and executed threat indicators exchanges in IODEF. The transport protocol used was RID. The threat information shared included indicators from DDoS attacks; and Malware incidents and Spear-Phishing that targets specific individuals after harvesting information about them. The results served as proof-of-concept (PoC) about how seemingly competing entities could use IODEF to exchange sanitized security information. As this was a PoC exercise only example information (no real threats) were shared as part of the exchanges.

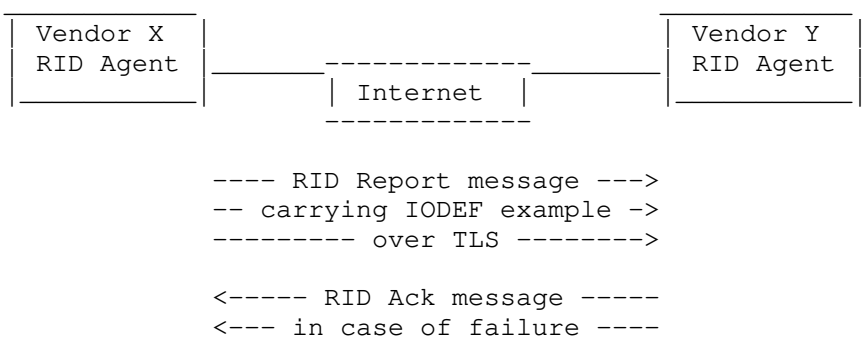

Figure 2: PoC peering topology

Figure 2 shows how RID interactions took place during the PoC. Participating organizations were running RID Agent software on-premises. The RID Agents formed peering relationships with other participating organizations. When Entity X had a new incident to exchange it would package it in IODEF and send it to Entity Y over TLS in a RID Report message. In case there was an issue with the message, Entity Y would send an RID Acknowledgement message back to Entity X which included an application level message to describe the issue. Interoperability between RID agents implementing [RFC6545] and [RFC6546] was also confirmed.

The first use-case included sharing of Malware Data Related to an Incident between CSIRTs. After Entity X detected an incident, she would put data about malware found during the incident in a backend system. Entity X then decided to share the incident information with Entity Y about the malware discovered. This could be a human decision or part of an automated process.

Below are the steps followed for the malware information exchange that was taking place:

(1) Entity X has a sharing agreement with Entity Y, and has already been configured with the IP address of Entity Y's RID Agent.

(2) Entity X's RID Agent connects to Entity Y's RID Agent, and mutual authentication occurs using PKI digital certificates.

(3) Entity X pushes out a RID Report message which contains information about N pieces of discovered malware. IODEF is used in RID to describe the

    (a) Hash of malware files

    (b) Registry settings changed by the malware

    (c) C&C Information for the malware

(4) Entity Y receives RID Report message, sends RID Acknowledgement message

(5) Entity Y stores the data in a format that makes it possible for the back end to know which source the data came from.

Another use-case was sharing a DDoS attack as explained in the following scenario: Entity X, a Critical Infrastructure and Key Resource (CIKR) company detects that their internet connection is saturated with an abnormal amount of traffic. Further investigation determines that this is an actual DDoS attack. Entity X's CSIT

contacts their ISP, Entity Y, and shares information with them about the attack traffic characteristics. Entity X's ISP is being overwhelmed by the amount of traffic, so it shares attack signatures and IP addresses of the most prolific hosts with its adjacent ISPs.

Below are the steps followed for a DDoS information exchange:

(1) Entity X has a sharing agreement with Entity Y, and has already been configured with the IP address of Entity Y's RID Agent.

(2) Entity X's RID Agent connects to Entity Y's RID Agent, and mutual authentication occurs using PKI digital certificates.

(3) Entity X pushes out a RID Report message which contains information about the DDoS attack. IODEF is used in RID to describe the

   (a) Start and Detect dates and times

   (b) IP Addresses of nodes sending DDoS Traffic

   (c) Sharing and Use Restrictions

   (d) Traffic characteristics (protocols and ports)

   (e) HTTP User-Agents used

   (f) IP Addresses of C&C for a botnet

(4) Entity Y receives RID Report message, sends RID Acknowledgement message

(5) Entity Y stores the data in a format that makes it possible for the back end to know which source the data came from.

(6) Entity Y shares information with other ISP Entities it has an established relationship with.

One more use-case was sharing spear-phishing email information as explained in the following scenario: The board members of several defense contractors receive a targeted email inviting them to attend a conference in San Francisco. The board members are asked to provide their personally identifiable information such as their home address, phone number, corporate email, etc in an attached document which came with the email. The board members are also asked to click on a URL which would allow them to reach the sign up page for the conference. One of the recipients believes the email to be a phishing attempt and forwards the email to their corporate CSIRT for

analysis. The CSIRT identifies the email as an attempted spear phishing incident and distributes the indicators to their sharing partners.

Below are the steps followed for a spear-phishing information exchange between CSIRTs that was part of this PoC.

(1) Entity X has a sharing agreement with Entity Y, and has already been configured with the IP address of Entity Y's RID Agent.

(2) Entity X's RID Agent connects to Entity Y's RID Agent, and mutual authentication occurs using PKI digital certificates.

(3) Entity X pushes out a RID Report message which contains information about the spear-phishing email. IODEF is used in RID to describe the

   (a) Attachment details (file Name, hash, size, malware family

   (b) Target description (IP, domain, NSLookup)

   (c) Email information (From, Subject, header information, date/ time, digital signature)

   (d) Confidence Score

(4) Entity Y receives RID Report message, sends RID Acknowledgement message

(5) Entity Y stores the data in a format that makes it possible for the back end to know which source the data came from.

Appendix B includes some of the incident IODEF example information that was exchanged by the organizations' RID Agents as part of this proof-of-concept.

5.3. Use-cases

Other use-cases of IODEF, other than the ones described above, could be:

(1) ISP notifying a national CERT or organization when it identifies and acts upon an incident and CERTs notifying ISPs when they are aware of incidents.

(2) Suspected phishing emails could be shared amongst organizations and national agencies. Automation could validate web content that the suspicious emails are pointing to. Identified

malicious content linked in a phishing email could then be shared using IODEF. Phishing campaigns could thus be subverted much faster by automating information sharing using IODEF.

(3) When finding a certificate that should be revoked, a third-party would forward an automated IODEF message to the CA with the full context of the certificate and the CA could act accordingly after checking its validity. Alternatively, in the event of a compromise of the private key of a certificate, a third-party could alert the certificate owner about the compromise using IODEF.

6. IANA Considerations

This memo does not require any IANA actions.

7. Security Considerations

This document does not incur any new security issues, since it only talks about the usage of IODEFv2 defined RFC7970. Nevertheless, readers of this document should refer to the Security Considerations section of [RFC7970].

8. References

8.1. Normative References

[RFC5901] Cain, P. and D. Jevans, "Extensions to the IODEF-Document Class for Reporting Phishing", RFC 5901, DOI 10.17487/RFC5901, July 2010, <https://www.rfc-editor.org/info/rfc5901>.

[RFC6545] Moriarty, K., "Real-time Inter-network Defense (RID)", RFC 6545, DOI 10.17487/RFC6545, April 2012, <https://www.rfc-editor.org/info/rfc6545>.

[RFC7203] Takahashi, T., Landfield, K., and Y. Kadobayashi, "An Incident Object Description Exchange Format (IODEF) Extension for Structured Cybersecurity Information", RFC 7203, DOI 10.17487/RFC7203, April 2014, <https://www.rfc-editor.org/info/rfc7203>.

[RFC7495] Montville, A. and D. Black, "Enumeration Reference Format for the Incident Object Description Exchange Format (IODEF)", RFC 7495, DOI 10.17487/RFC7495, March 2015, <https://www.rfc-editor.org/info/rfc7495>.

[RFC7970] Danyliw, R., "The Incident Object Description Exchange Format Version 2", RFC 7970, DOI 10.17487/RFC7970, November 2016, <https://www.rfc-editor.org/info/rfc7970>.

8.2. Informative References

[implementations] "Implementations on IODEF", <http://siis.realmv6.org/implementations/>.

[RFC6546] Trammell, B., "Transport of Real-time Inter-network Defense (RID) Messages over HTTP/TLS", RFC 6546, DOI 10.17487/RFC6546, April 2012, <https://www.rfc-editor.org/info/rfc6546>.

[RFC8134] Inacio, C. and D. Miyamoto, "Management Incident Lightweight Exchange (MILE) Implementation Report", RFC 8134, DOI 10.17487/RFC8134, May 2017, <https://www.rfc-editor.org/info/rfc8134>.

Appendix A. Indicator predicate logic examples

In the following example the EventData class evaluates as a Flow of one System with source address being (192.0.2.104 OR 192.0.2.106) AND target address 198.51.100.1.

```
<!-- ...XML code omitted... -->
 <IndicatorData>
   <Indicator>
     <IndicatorID name="csirt.example.com" version="1">
     G90823490
     </IndicatorID>
     <Description>C2 domains</Description>
     <IndicatorExpression operator="and">
       <IndicatorExpression operator="or">
         <Observable>
           <System category="source" spoofed="no">
             <Node>
               <Address category="ipv4-addr">
                 192.0.2.104
               </Address>
             </Node>
           </System>
         </Observable>
         <Observable>
           <System category="source" spoofed="no">
             <Node>
               <Address category="ipv4-addr">
                 192.0.2.106
               </Address>
             </Node>
           </System>
         </Observable>
       </IndicatorExpression>
       <Observable>
         <System category="target" spoofed="no">
           <Node>
             <Address category="ipv4-addr">
               198.51.100.1
             </Address>
           </Node>
         </System>
       </Observable>
     </IndicatorExpression>
   </Indicator>
 </IndicatorData>
<!-- ...XML code omitted... -->
```

Similarly, the FileData Class can be an observable in an IndicatorExpression. The hash values of two files can be used to match against an indicator using Boolean "or" logic. In the following example the indicator consists of either of the two files with two different hashes.

```
<!-- ...XML code omitted... -->
 <IndicatorData>
   <Indicator>
     <IndicatorID name="csirt.example.com" version="1">
     A4399IWQ
     </IndicatorID>
     <Description>File hash watchlist</Description>
     <IndicatorExpression operator="or">
         <Observable>
           <FileData>
             <File>
               <FileName>dummy.txt</FileName>
               <HashData scope="file-contents">
                 <Hash>
                  <ds:DigestMethod Algorithm=
                  "http://www.w3.org/2001/04/xmlenc#sha256"/>
                  <ds:DigestValue>
                   141accec23e7e5157de60853cb1e01bc38042d
                   08f9086040815300b7fe75c184
                  </ds:DigestValue>
                 </Hash>
               </HashData>
             </File>
           </FileData>
         </Observable>
         <Observable>
           <FileData>
             <File>
               <FileName>dummy2.txt</FileName>
               <HashData scope="file-contents">
                 <Hash>
                  <ds:DigestMethod Algorithm=
                  "http://www.w3.org/2001/04/xmlenc#sha256"/>
                  <ds:DigestValue>
                   141accec23e7e5157de60853cb1e01bc38042d
                   08f9086040815300b7fe75c184
                  </ds:DigestValue>
                 </Hash>
               </HashData>
             </File>
           </FileData>
         </Observable>
     </IndicatorExpression>
   </Indicator>
 </IndicatorData>
<!-- ...XML code omitted... -->
```

Appendix B. Inter-vendor and Service Provider Exercise Examples

Below some of the incident IODEF example information that was exchanged by the vendors as part of this proof-of-concept Inter-vendor and Service Provider Exercise.

B.1. Malware Delivery URL

This example indicates malware and related URL for file delivery.

```
<?xml version="1.0" encoding="UTF-8"?>
<IODEF-Document version="2.00"
                xmlns="urn:ietf:params:xml:ns:iodef-2.0"
                xmlns:iodef="urn:ietf:params:xml:ns:iodef-2.0"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <iodef:Incident purpose="reporting">
    <iodef:IncidentID name="csirt.example.com">
      189801
    </iodef:IncidentID>
    <iodef:ReportTime>2012-12-05T12:20:00+00:00</iodef:ReportTime>
    <iodef:GenerationTime>2012-12-05T12:20:00+00:00</iodef:GenerationTime>
    <iodef:Description>Malware and related indicators</iodef:Description>
    <iodef:Assessment occurrence="potential">
      <iodef:SystemImpact severity="medium" type="breach-privacy">
        <iodef:Description>Malware with C&amp;amp;C
        </iodef:Description>
      </iodef:SystemImpact>
    </iodef:Assessment>
    <iodef:Contact role="creator" type="organization">
      <iodef:ContactName>example.com CSIRT
      </iodef:ContactName>
      <iodef:Email>
        <iodef:EmailTo>contact@csirt.example.com
        </iodef:EmailTo>
      </iodef:Email>
    </iodef:Contact>
    <iodef:EventData>
      <iodef:Flow>
        <iodef:System category="source">
          <iodef:Node>
            <iodef:Address category="ipv4-addr">192.0.2.200
            </iodef:Address>
                        <iodef:Address category="site-uri">
              /log-bin/lunch_install.php?aff_id=1&amp;lunch_id=1&amp;maddr=&amp;
action=install
            </iodef:Address>
          </iodef:Node>
          <iodef:NodeRole category="www"/>
        </iodef:System>
      </iodef:Flow>
    </iodef:EventData>
  </iodef:Incident>
</IODEF-Document>
```

B.2. DDoS

The DDoS test exchanged information that described a DDoS including protocols and ports, bad IP addresses and HTTP User-Agent fields.

The IODEF version used for the data representation was based on [RFC7970].

```
<?xml version="1.0" encoding="UTF-8"?>
<IODEF-Document version="2.00"
                xmlns="urn:ietf:params:xml:ns:iodef-2.0"
                xmlns:iodef="urn:ietf:params:xml:ns:iodef-2.0"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <iodef:Incident purpose="reporting" restriction="default">
    <iodef:IncidentID name="csirt.example.com">
      189701
    </iodef:IncidentID>
    <iodef:DetectTime>2013-02-05T01:15:45+00:00</iodef:DetectTime>
    <iodef:StartTime>2013-02-05T00:34:45+00:00</iodef:StartTime>
    <iodef:ReportTime>2013-02-05T01:34:45+00:00</iodef:ReportTime>
    <iodef:GenerationTime>2013-02-05T01:15:45+00:00</iodef:GenerationTime>
    <iodef:Description>DDoS Traffic Seen</iodef:Description>
    <iodef:Assessment occurrence="actual">
      <iodef:SystemImpact severity="medium" type="availability-system">
        <iodef:Description>DDoS Traffic
        </iodef:Description>
      </iodef:SystemImpact>
      <iodef:Confidence rating="high"/>
    </iodef:Assessment>
    <iodef:Contact role="creator" type="organization">
      <iodef:ContactName>Dummy Test</iodef:ContactName>
      <iodef:Email>
        <iodef:EmailTo>contact@dummytest.com
        </iodef:EmailTo>
      </iodef:Email>
    </iodef:Contact>
    <iodef:EventData>
      <iodef:Description>
        Dummy Test sharing with ISP1
      </iodef:Description>
      <iodef:Method>
        <iodef:Reference>
          <iodef:URL>
            http://blog.spiderlabs.com/2011/01/loic-ddos-
            analysis-and-detection.html
          </iodef:URL>
          <iodef:URL>
            http://en.wikipedia.org/wiki/Low_Orbit_Ion_Cannon
          </iodef:URL>
          <iodef:Description>
            Low Orbit Ion Cannon User Agent
          </iodef:Description>
        </iodef:Reference>
```

```
      </iodef:Method>
      <iodef:Flow>
        <iodef:System category="source" spoofed="no">
          <iodef:Node>
            <iodef:Address category="ipv4-addr">
              192.0.2.104
            </iodef:Address>
          </iodef:Node>
          <iodef:Service ip-protocol="6">
            <iodef:Port>1337</iodef:Port>
          </iodef:Service>
        </iodef:System>
        <iodef:System category="source" spoofed="no">
          <iodef:Node>
            <iodef:Address category="ipv4-addr">
              192.0.2.106
            </iodef:Address>
          </iodef:Node>
          <iodef:Service ip-protocol="6">
            <iodef:Port>1337</iodef:Port>
          </iodef:Service>
        </iodef:System>
        <iodef:System category="source" spoofed="yes">
          <iodef:Node>
            <iodef:Address category="ipv4-net">
              198.51.100.0/24
            </iodef:Address>
          </iodef:Node>
          <iodef:Service ip-protocol="6">
            <iodef:Port>1337</iodef:Port>
          </iodef:Service>
        </iodef:System>
        <iodef:System category="source" spoofed="yes">
          <iodef:Node>
            <iodef:Address category="ipv6-addr">
              2001:db8:dead:beef::1
            </iodef:Address>
          </iodef:Node>
          <iodef:Service ip-protocol="6">
            <iodef:Port>1337</iodef:Port>
          </iodef:Service>
        </iodef:System>
        <iodef:System category="target">
          <iodef:Node>
            <iodef:Address category="ipv4-addr">
              203.0.113.1
            </iodef:Address>
          </iodef:Node>
```

```
          <iodef:Service ip-protocol="6">
            <iodef:Port>80</iodef:Port>
          </iodef:Service>
        </iodef:System>
        <iodef:System category="sensor">
          <iodef:Node>
          </iodef:Node>
          <iodef:Description>
            Information provided in Flow class instance is from
            Inspection of traffic from network tap
          </iodef:Description>
        </iodef:System>
      </iodef:Flow>
      <iodef:Expectation action="other"/>
    </iodef:EventData>
    <iodef:IndicatorData>
      <iodef:Indicator>
        <iodef:IndicatorID name="csirt.example.com" version="1">
          G83345941
        </iodef:IndicatorID>
        <iodef:Description>
          User-Agent string
        </iodef:Description>
        <iodef:Observable>
          <iodef:BulkObservable type="http-user-agent">
            <iodef:BulkObservableList>
              user-agent="Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US;
 rv:1.9.2.12) Gecko/20101026 Firefox/3.6.12">
            </iodef:BulkObservableList>
          </iodef:BulkObservable>
        </iodef:Observable>
      </iodef:Indicator>
    </iodef:IndicatorData>
  </iodef:Incident>
</IODEF-Document>
```

B.3. Spear-Phishing

The Spear-Phishing test exchanged information that described a Spear-Phishing email including DNS records and addresses about the sender, malicious attached file information and email data. The IODEF version used for the data representation was based on [RFC7970].

```
<?xml version="1.0" encoding="UTF-8"?>
<IODEF-Document version="2.00"
               xmlns="urn:ietf:params:xml:ns:iodef-2.0"
               xmlns:iodef="urn:ietf:params:xml:ns:iodef-2.0"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
```

```
  <iodef:Incident purpose="reporting">
    <iodef:IncidentID name="csirt.example.com">
      189601
    </iodef:IncidentID>
    <iodef:DetectTime>2013-01-04T08:06:12+00:00</iodef:DetectTime>
    <iodef:StartTime>2013-01-04T08:01:34+00:00</iodef:StartTime>
    <iodef:EndTime>2013-01-04T08:31:27+00:00</iodef:EndTime>
    <iodef:ReportTime>2013-01-04T09:15:45+00:00</iodef:ReportTime>
    <iodef:GenerationTime>2013-01-04T09:15:45+00:00</iodef:GenerationTime>
    <iodef:Description>
      Zeus Spear Phishing E-mail with Malware Attachment
    </iodef:Description>
    <iodef:Assessment occurrence="potential">
      <iodef:SystemImpact severity="medium" type="takeover-system">
        <iodef:Description>
          Malware with Command and Control Server and System Changes
        </iodef:Description>
      </iodef:SystemImpact>
    </iodef:Assessment>
    <iodef:Contact role="creator" type="organization">
      <iodef:ContactName>example.com CSIRT</iodef:ContactName>
      <iodef:Email>
        <iodef:EmailTo>contact@csirt.example.com</iodef:EmailTo>
        </iodef:Email>
    </iodef:Contact>
    <iodef:EventData>
      <iodef:Description>
        Targeting Defense Contractors,
        specifically board members attending Dummy Con
      </iodef:Description>
      <iodef:Method>
        <iodef:Reference observable-id="ref-1234">
          <iodef:Description>Zeus</iodef:Description>
        </iodef:Reference>
      </iodef:Method>
      <iodef:Flow>
        <iodef:System category="source">
          <iodef:Node>
            <iodef:Address category="site-uri">
              http://www.zeusevil.example.com
            </iodef:Address>
            <iodef:Address category="ipv4-addr">
              192.0.2.166
            </iodef:Address>
            <iodef:Address category="asn">
              65535
            </iodef:Address>
            <iodef:Address category="ext-value"
```

```
                        ext-category="as-name">
            EXAMPLE-AS - University of Example"
          </iodef:Address>
          <iodef:Address category="ext-value"
                        ext-category="as-prefix">
            192.0.2.0/24
          </iodef:Address>
        </iodef:Node>
        <iodef:NodeRole category="malware-distribution"/>
      </iodef:System>
    </iodef:Flow>
    <iodef:Flow>
      <iodef:System category="source">
        <iodef:Node>
          <iodef:DomainData>
            <Name>mail1.evildave.example.com</Name>
          </iodef:DomainData>
          <iodef:Address category="ipv4-addr">
            198.51.100.6
          </iodef:Address>
          <iodef:Address category="asn">
            65534
          </iodef:Address>
          <iodef:Address category="ext-value"
                        ext-category="as-name">
            EXAMPLE-AS - University of Example
          </iodef:Address>
          <iodef:DomainData>
            <iodef:Name>evildave.example.com</iodef:Name>
            <iodef:DateDomainWasChecked>2013-01-04T09:10:24+00:00
            </iodef:DateDomainWasChecked>
            <!-- <iodef:RelatedDNS RecordType="MX"> -->
            <iodef:RelatedDNS dtype="string">
              evildave.example.com MX prefernce = 10, mail exchanger
              = mail1.evildave.example.com
            </iodef:RelatedDNS>
            <iodef:RelatedDNS dtype="string">
              mail1.evildave.example.com
              internet address = 198.51.100.6
            </iodef:RelatedDNS>
            <iodef:RelatedDNS dtype="string">
              zuesevil.example.com. IN TXT \"v=spf1 a mx -all\"
            </iodef:RelatedDNS>
          </iodef:DomainData>
        </iodef:Node>
        <iodef:NodeRole category="mail">
          <iodef:Description>
            Sending phishing mails
```

```
            </iodef:Description>
          </iodef:NodeRole>
          <iodef:Service>
            <iodef:EmailData>
              <iodef:EmailFrom>
                emaildave@evildave.example.com
              </iodef:EmailFrom>
              <iodef:EmailSubject>
                Join us at Dummy Con
              </iodef:EmailSubject>
              <iodef:EmailX-Mailer>
                StormRider 4.0
              </iodef:EmailX-Mailer>
            </iodef:EmailData>
          </iodef:Service>
        </iodef:System>
        <iodef:System category="target">
          <iodef:Node>
            <iodef:Address category="ipv4-addr">
              203.0.113.2
            </iodef:Address>
          </iodef:Node>
        </iodef:System>
      </iodef:Flow>
      <iodef:Expectation action="other"/>
      <iodef:Record>
        <iodef:RecordData>
          <iodef:FileData observable-id="fd-1234">
            <iodef:File>
              <iodef:FileName>
                Dummy Con Sign Up Sheet.txt
              </iodef:FileName>
              <iodef:FileSize>
                152
              </iodef:FileSize>
              <iodef:HashData scope="file-contents">
                <iodef:Hash>
                  <ds:DigestMethod
                          Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                  <ds:DigestValue>
                    141accec23e7e5157de60853cb1e01bc38042d
                    08f9086040815300b7fe75c184
                  </ds:DigestValue>
                </iodef:Hash>
              </iodef:HashData>
            </iodef:File>
          </iodef:FileData>
        </iodef:RecordData>
```

```
        <iodef:RecordData>
          <iodef:CertificateData>
            <iodef:Certificate>
              <ds:X509Data>
                <ds:X509IssuerSerial>
                  <ds:X509IssuerName>FakeCA
                  </ds:X509IssuerName>
                  <ds:X509SerialNumber>
                    57482937101
                  </ds:X509SerialNumber>
                </ds:X509IssuerSerial>
                <ds:X509SubjectName>EvilDaveExample
                </ds:X509SubjectName>
              </ds:X509Data>
            </iodef:Certificate>
          </iodef:CertificateData>
        </iodef:RecordData>
      </iodef:Record>
    </iodef:EventData>
  </iodef:Incident>
</IODEF-Document>
```

B.4. Malware

In this test, malware information was exchanged using RID and IODEF. The information included file hashes, registry setting changes and the C&C servers the malware uses.

```
<?xml version="1.0" encoding="UTF-8"?>
<IODEF-Document version="2.00"
                xmlns="urn:ietf:params:xml:ns:iodef-2.0"
                xmlns:iodef="urn:ietf:params:xml:ns:iodef-2.0"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <iodef:Incident purpose="reporting">
    <iodef:IncidentID name="csirt.example.com">
      189234
    </iodef:IncidentID>
    <iodef:ReportTime>2013-03-07T16:14:56.757+05:30</iodef:ReportTime>
    <iodef:GenerationTime>2013-03-07T16:14:56.757+05:30</iodef:GenerationTime>
    <iodef:Description>
      Malware and related indicators identified
    </iodef:Description>
    <iodef:Assessment occurrence="potential">
      <iodef:SystemImpact severity="medium" type="breach-proprietary">
        <iodef:Description>
          Malware with Command and Control Server and System Changes
        </iodef:Description>
```

```
      </iodef:SystemImpact>
    </iodef:Assessment>
    <iodef:Contact role="creator" type="organization">
      <iodef:ContactName>example.com CSIRT</iodef:ContactName>
      <iodef:Email>
        <iodef:EmailTo>contact@csirt.example.com</iodef:EmailTo>
      </iodef:Email>
    </iodef:Contact>
    <iodef:EventData>
      <iodef:Method>
        <iodef:Reference>
          <iodef:URL>
            http://www.threatexpert.example.com/report.aspx?
            md5=e2710ceb088dacdcb03678db250742b7
          </iodef:URL>
          <iodef:Description>Zeus</iodef:Description>
        </iodef:Reference>
      </iodef:Method>
      <iodef:Flow>
        <iodef:System category="source">
          <iodef:Node>
            <iodef:Address category="ipv4-addr" observable-id="addr-c2-91011-001
">
              203.0.113.200
            </iodef:Address>
            <iodef:Address category="site-uri" observable-id="addr-c2-91011-002"
>
              http://zeus.556677889900.example.com/log-bin/
              lunch_install.php?aff_id=1&amp;amp;
              lunch_id=1&amp;amp;maddr=&amp;amp;
              action=install
            </iodef:Address>
          </iodef:Node>
          <iodef:NodeRole category="c2-server"/>
        </iodef:System>
      </iodef:Flow>
      <iodef:Record>
        <iodef:RecordData>
          <iodef:FileData observable-id="file-91011-001">
            <iodef:File>
              <iodef:HashData scope="file-contents">
                <iodef:Hash>
                  <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#s
ha1"/>
                  <ds:DigestValue>
                    MHg2NzUxQTI1MzQ4M0E2N0Q4NkUwRjg0NzYwRjYxRjEwQkJDQzJFREZG
                  </ds:DigestValue>
                </iodef:Hash>
              </iodef:HashData>
            </iodef:File>
            <iodef:File>
```

```
              <iodef:HashData scope="file-contents">
                <iodef:Hash>
                  <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#m
d5"/>
                  <ds:DigestValue>
                    MHgyRTg4ODA5ODBENjI0NDdFOTc5MEFGQTg5NTEzRjBBNA==
                  </ds:DigestValue>
                </iodef:Hash>
              </iodef:HashData>
            </iodef:File>
          </iodef:FileData>
          <iodef:WindowsRegistryKeysModified observable-id="regkey-91011-001">
            <iodef:Key registryaction="add-value">
              <iodef:KeyName>
                HKLM\Software\Microsoft\Windows\
                CurrentVersion\Run\tamg
              </iodef:KeyName>
              <iodef:Value>
                ?\?\?%System%\wins\mc.exe\?\??
              </iodef:Value>
            </iodef:Key>
            <iodef:Key registryaction="modify-value">
              <iodef:KeyName>HKLM\Software\Microsoft\
                Windows\CurrentVersion\Run\dqo
              </iodef:KeyName>
              <iodef:Value>"\"\"%Windir%\Resources\
                Themes\Luna\km.exe\?\?"
              </iodef:Value>
            </iodef:Key>
          </iodef:WindowsRegistryKeysModified>
        </iodef:RecordData>
      </iodef:Record>
    </iodef:EventData>
    <iodef:EventData>
      <iodef:Method>
        <iodef:Reference>
          <iodef:URL>
            http://www.threatexpert.example.com/report.aspx?
            md5=c3c528c939f9b176c883ae0ce5df0001
          </iodef:URL>
          <iodef:Description>Cridex</iodef:Description>
        </iodef:Reference>
      </iodef:Method>
      <iodef:Flow>
        <iodef:System category="source">
          <iodef:Node>
            <iodef:Address category="ipv4-addr" observable-id="addr-c2-91011-003
">
              203.0.113.100
            </iodef:Address>
```

```
          </iodef:Node>
          <iodef:NodeRole category="c2-server"/>
          <iodef:Service ip-protocol="6">
            <iodef:Port>8080</iodef:Port>
          </iodef:Service>
        </iodef:System>
      </iodef:Flow>
      <iodef:Record>
        <iodef:RecordData>
          <iodef:FileData observable-id="file-91011-002">
            <iodef:File>
              <iodef:HashData scope="file-contents">
                <iodef:Hash>
                  <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#s
ha1"/>
                  <ds:DigestValue>
                    MHg3MjYzRkUwRDNBMDk1RDU5QzhFMEM4OTVBOUM1ODVFMzQzRTcxNDFD
                  </ds:DigestValue>
                </iodef:Hash>
              </iodef:HashData>
            </iodef:File>
          </iodef:FileData>
          <iodef:FileData observable-id="file-91011-003">
            <iodef:File>
              <iodef:HashData scope="file-contents">
                <iodef:Hash>
                  <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#m
d5"/>
                  <ds:DigestValue>
                    MHg0M0NEODUwRkNEQURFNDMzMEE1QkVBNkYxNkVFOTcxQw==
                  </ds:DigestValue>
                </iodef:Hash>
              </iodef:HashData>
            </iodef:File>
          </iodef:FileData>
          <iodef:WindowsRegistryKeysModified observable-id="regkey-91011-002">
            <iodef:Key registryaction="add-value">
              <iodef:KeyName>
                HKLM\Software\Microsoft\Windows\
                CurrentVersion\Run\KB00121600.exe
              </iodef:KeyName>
              <iodef:Value>
                \?\?%AppData%\KB00121600.exe\?\?
              </iodef:Value>
            </iodef:Key>
          </iodef:WindowsRegistryKeysModified>
        </iodef:RecordData>
      </iodef:Record>
    </iodef:EventData>
    <iodef:IndicatorData>
```

```
      <iodef:Indicator>
        <iodef:IndicatorID name="csirt.example.com" version="1">
          ind-91011
        </iodef:IndicatorID>
        <iodef:Description>
          evil c2 server, file hash, and registry key
        </iodef:Description>
        <iodef:IndicatorExpression operator="or">
          <iodef:IndicatorExpression operator="or">
            <iodef:Observable>
              <iodef:Address category="site-uri" observable-id="addr-qrst">
                http://foo.example.com:12345/evil/cc.php
              </iodef:Address>
            </iodef:Observable>
            <iodef:Observable>
              <iodef:Address category="ipv4-addr" observable-id="addr-stuv">
                192.0.2.1
              </iodef:Address>
            </iodef:Observable>
            <iodef:Observable>
              <iodef:Address category="ipv4-addr" observable-id="addr-tuvw">
                198.51.100.1
              </iodef:Address>
            </iodef:Observable>
            <iodef:Observable>
              <iodef:Address category="ipv6-addr" observable-id="addr-uvwx">
                2001:db8:dead:beef::1
              </iodef:Address>
            </iodef:Observable>
            <iodef:ObservableReference uid-ref="addr-c2-91011-001"/>
            <iodef:ObservableReference uid-ref="addr-c2-91011-002"/>
            <iodef:ObservableReference uid-ref="addr-c2-91011-003"/>
          </iodef:IndicatorExpression>
          <iodef:IndicatorExpression operator="and">
            <iodef:Observable>
              <iodef:FileData observable-id="file-91011-000">
                <iodef:File>
                  <iodef:HashData scope="file-contents">
                    <iodef:Hash>
                      <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmle
nc#sha256"/>
                      <ds:DigestValue>
                        141accec23e7e5157de60853cb1e01bc38042d08f9086040815300b7
fe75c184
                      </ds:DigestValue>
                    </iodef:Hash>
                  </iodef:HashData>
                </iodef:File>
              </iodef:FileData>
            </iodef:Observable>
```

```
            <iodef:Observable>
              <iodef:WindowsRegistryKeysModified observable-id="regkey-91011-000
">
                <iodef:Key registryaction="add-key"
                           observable-id="regkey-vwxy">
                  <iodef:KeyName>
                    HKLM\SYSTEM\CurrentControlSet\
                    Services\.Net CLR
                  </iodef:KeyName>
                </iodef:Key>
                <iodef:Key registryaction="add-key"
                           observable-id="regkey-wxyz">
                  <iodef:KeyName>
                    HKLM\SYSTEM\CurrentControlSet\
                    Services\.Net CLR\Parameters
                  </iodef:KeyName>
                  <iodef:Value>
                    \"\"%AppData%\KB00121600.exe\"\"
                  </iodef:Value>
                </iodef:Key>
                <iodef:Key registryaction="add-value"
                           observable-id="regkey-xyza">
                  <iodef:KeyName>
                    HKLM\SYSTEM\CurrentControlSet\Services\
                    .Net CLR\Parameters\ServiceDll
                  </iodef:KeyName>
                  <iodef:Value>C:\bad.exe</iodef:Value>
                </iodef:Key>
                <iodef:Key registryaction="modify-value"
                           observable-id="regkey-zabc">
                  <iodef:KeyName>
                    HKLM\SYSTEM\CurrentControlSet\
                    Services\.Net CLR\Parameters\Bar
                  </iodef:KeyName>
                  <iodef:Value>Baz</iodef:Value>
                </iodef:Key>
              </iodef:WindowsRegistryKeysModified>
            </iodef:Observable>
          </iodef:IndicatorExpression>
          <iodef:IndicatorExpression operator="or">
            <iodef:IndicatorExpression operator="and">
              <iodef:ObservableReference uid-ref="file-91011-001"/>
              <iodef:ObservableReference uid-ref="regkey-91011-001"/>
            </iodef:IndicatorExpression>
            <iodef:IndicatorExpression operator="and">
              <iodef:IndicatorExpression operator="or">
                <iodef:ObservableReference uid-ref="file-91011-002"/>
                <iodef:ObservableReference uid-ref="file-91011-003"/>
              </iodef:IndicatorExpression>
```

```
              <iodef:ObservableReference uid-ref="regkey-91011-002"/>
            </iodef:IndicatorExpression>
          </iodef:IndicatorExpression>
        </iodef:IndicatorExpression>
      </iodef:Indicator>
    </iodef:IndicatorData>
  </iodef:Incident>
</IODEF-Document>
```

B.5. IoT Malware

The IoT Malware test exchanged information that described a bad IP address of IoT malware and its scanned ports. This example information is extracted from alert messages of a Darknet monitoring system referred in [RFC8134]. The IODEF version used for the data representation was based on [RFC7970].

```
<?xml version="1.0" encoding="UTF-8"?>
<IODEF-Document version="2.00"
                xmlns="urn:ietf:params:xml:ns:iodef-2.0"
                xmlns:iodef="urn:ietf:params:xml:ns:iodef-2.0"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <iodef:Incident purpose="reporting">
    <iodef:IncidentID name="csirt.example.com">
      189802
    </iodef:IncidentID>
    <iodef:ReportTime>2017-03-01T01:15:00+09:00</iodef:ReportTime>
    <iodef:GenerationTime>2017-03-01T01:15:00+09:00</iodef:GenerationTime>
    <iodef:Description>IoT Malware and related indicators</iodef:Description>
    <iodef:Assessment occurrence="potential">
      <iodef:SystemImpact severity="medium" type="takeover-system">
        <iodef:Description>IoT Malware is scanning other hosts
        </iodef:Description>
      </iodef:SystemImpact>
    </iodef:Assessment>
    <iodef:Contact role="creator" type="organization">
      <iodef:ContactName>example.com CSIRT
      </iodef:ContactName>
      <iodef:Email>
        <iodef:EmailTo>contact@csirt.example.com
        </iodef:EmailTo>
      </iodef:Email>
    </iodef:Contact>
    <iodef:EventData>
      <iodef:Discovery source="nidps">
        <iodef:Description>
          Detected by darknet monitoring
        </iodef:Description>
```

```
      </iodef:Discovery>
      <iodef:Flow>
        <iodef:System category="source">
          <iodef:Node>
            <iodef:Address category="ipv4-addr">
              192.0.2.210
            </iodef:Address>
          </iodef:Node>
          <iodef:NodeRole category="camera"/>
          <iodef:Service ip-protocol="6">
            <iodef:Port>23</iodef:Port>
          </iodef:Service>
          <iodef:OperatingSystem>
            <iodef:Description>
              Example Surveillance Camera OS 2.1.1
            </iodef:Description>
          </iodef:OperatingSystem>
        </iodef:System>
      </iodef:Flow>
      <iodef:EventData>
        <iodef:Flow>
          <iodef:System category="target">
            <iodef:Node>
              <iodef:Address category="ipv4-addr">
                198.51.100.1
              </iodef:Address>
            </iodef:Node>
            <iodef:NodeRole category="honeypot"/>
            <iodef:Service ip-protocol="6">
              <iodef:Port>23</iodef:Port>
            </iodef:Service>
          </iodef:System>
        </iodef:Flow>
      </iodef:EventData>
      <iodef:EventData>
        <iodef:Flow>
          <iodef:System category="target">
            <iodef:Node>
              <iodef:Address category="ipv4-addr">
                198.51.100.94
              </iodef:Address>
            </iodef:Node>
            <iodef:NodeRole category="honeypot"/>
            <iodef:Service ip-protocol="6">
              <iodef:Port>23</iodef:Port>
            </iodef:Service>
          </iodef:System>
        </iodef:Flow>
```

```
      </iodef:EventData>
      <iodef:EventData>
        <iodef:Flow>
          <iodef:System category="target">
            <iodef:Node>
              <iodef:Address category="ipv4-addr">
                198.51.100.237
              </iodef:Address>
            </iodef:Node>
            <iodef:NodeRole category="honeypot"/>
            <iodef:Service ip-protocol="6">
              <iodef:Port>2323</iodef:Port>
            </iodef:Service>
          </iodef:System>
        </iodef:Flow>
      </iodef:EventData>
    </iodef:EventData>
  </iodef:Incident>
</IODEF-Document>
```

Authors' Addresses

Panos Kampanakis
Cisco Systems

Email: pkampana@cisco.com

Mio Suzuki
NICT
4-2-1, Nukui-Kitamachi
Koganei, Tokyo 184-8795
JP

Email: mio@nict.go.jp

MILE Working Group
Internet-Draft
Obsoletes: 5070 (if approved)
Intended status: Standards Track
Expires: May 13, 2015

R. Danyliw
CERT
P. Stoecker
RSA

November 9, 2014

The Incident Object Description Exchange Format v2
draft-ietf-mile-rfc5070-bis-10

Abstract

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

Status of This Memo

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 May 13, 2015.

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 following classes were added to IODEF-Document: AdditionalData.
- The following class was added to Incident: IndicatorData.
- 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: HashData and WindowsRegistryKeysModified.
- The following classes were added to the RelatedActivity class: ThreatActor, Campaign, Confidence, Description, and AdditionalData.
- The following classes were added to Assessment: IncidentCategory, SystemImpact, BusinessImpact, IntendedImpact and MitigatingFactor.
- 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.
- Additional enumerated values were added to the following attributes: @restriction, {Expectation, HistoryItem}@action, NodeRole@category, Incident@purpose, Contact@role, AdditionalData@dtype, System@spoofed.
- Removed all "ext-" attributes in favor of using an IANA registry for extending attributes.
- Removed Impact class in favor of using SystemImpact and IncidentCategory.

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.

2.16. Identifiers and Identifier References

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

The ID and IDREF data types are implemented as "xs:ID" and "xs:IDREF" in the schema.

3. The IODEF Data Model

In this section, the individual components of the IODEF data model will be discussed in detail. For each class, the semantics will be described and the relationship with other classes will be depicted with UML. When necessary, specific comments will be made about corresponding definition in the schema in Section 8

3.1. IODEF-Document Class

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

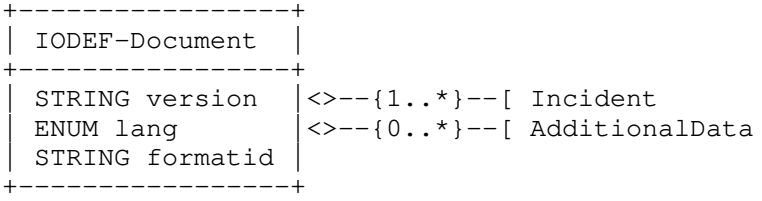

Figure 1: IODEF-Document Class

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 [RFC5646] 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.

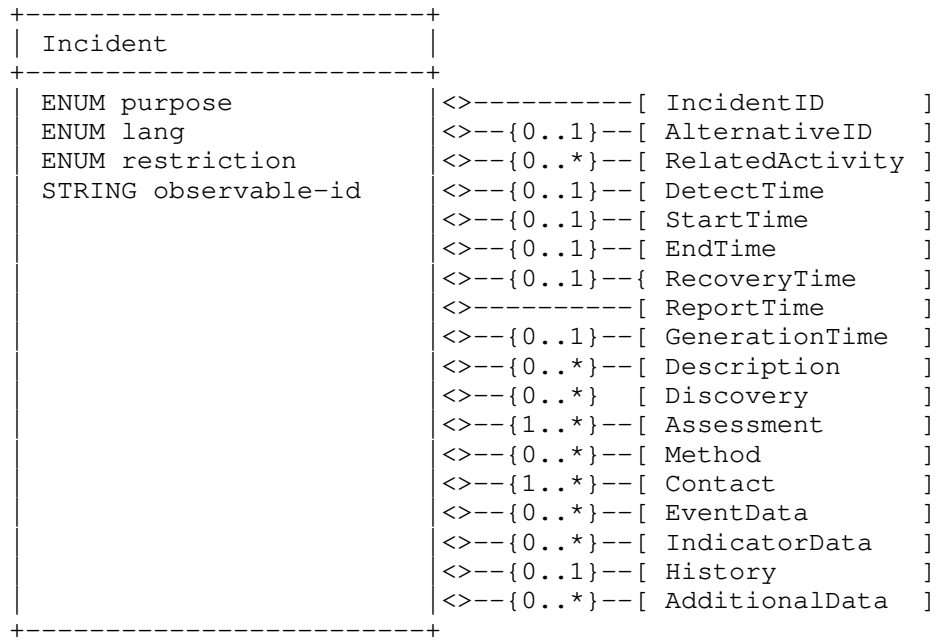

Figure 2: The Incident Class

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.

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

ReportTime
One. The time the incident was reported.

GenerationTime
One. The time the content in this Incident class was generated.

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

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

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

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

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

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

IndicatorData
Zero or more. Description of indicators.

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

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

The Incident class has three attributes:

purpose
Required. ENUM. The purpose attribute represents the reason why the IODEF document was created. It is closely related to the Expectation class (Section 3.17). These values are maintained in the "Incident-purpose" IANA registry per Table 1. This attribute is defined as an enumerated list:

1. traceback. The document was sent for trace-back purposes.

2. mitigation. The document was sent to request aid in mitigating the described activity.

3. reporting. The document was sent to comply with reporting requirements.

4. watch. The document was sent to convey indicators to watch for particular activity.

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

lang
Optional. ENUM. A valid language code per [RFC5646] 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.

observable-id
Optional. ID. See Section 3.3.2.

3.3. Common Attributes

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

3.3.1. restriction Attribute

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

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

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

This attribute is defined as an enumerated value with a default value of "private". Note that the default value of the restriction attribute is only defined in the context of the Incident class. In other classes where this attribute is used, no default is specified.

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

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

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

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

4. private. The information may not be shared.

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

6. white. Same as 'public'.

7. green. Same as 'partner'.

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

9. red. Same as 'private'.

3.3.2. observable-id Attribute

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

3.4. IncidentID Class

The IncidentID class represents an incident tracking number that is unique in the context of the CSIRT and identifies the activity characterized in an IODEF Document. This identifier would serve as an index into the CSIRT incident handling system. The combination of the name attribute and the string in the element content MUST be a globally unique identifier describing the activity. Documents generated by a given CSIRT MUST NOT reuse the same value unless they are referencing the same incident.

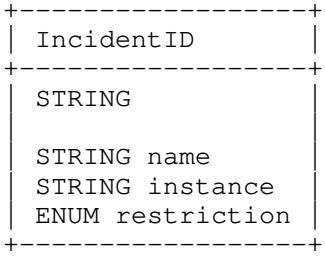

Figure 3: The IncidentID Class

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.

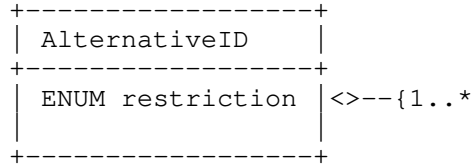

Figure 4: The AlternativeID Class

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.

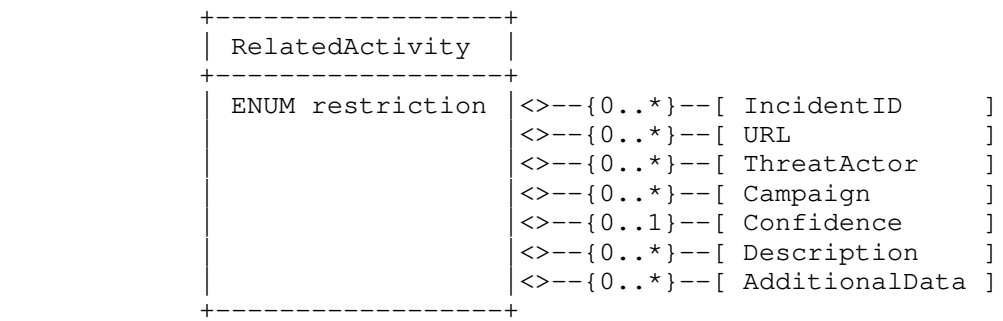

Figure 5: RelatedActivity Class

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.

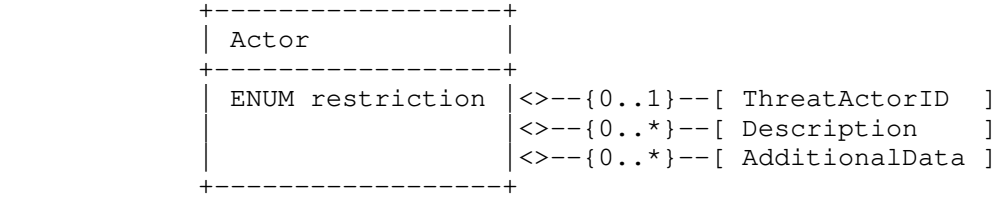

Figure 6: ThreatActor Class

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 ...

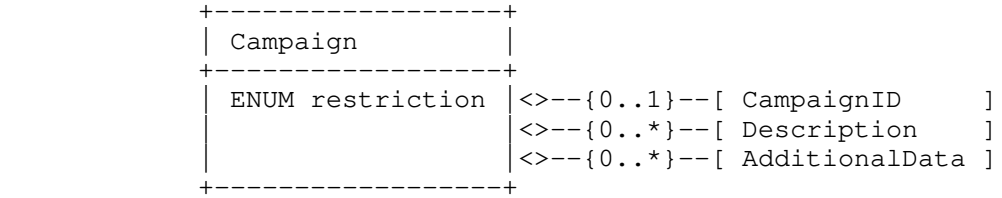

Figure 7: Campaign Class

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.

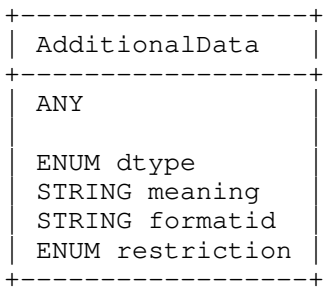

Figure 8: The AdditionalData Class

The AdditionalData class has four attributes:

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

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

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

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

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

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

6. ntpstamp. Same as date-time.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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.

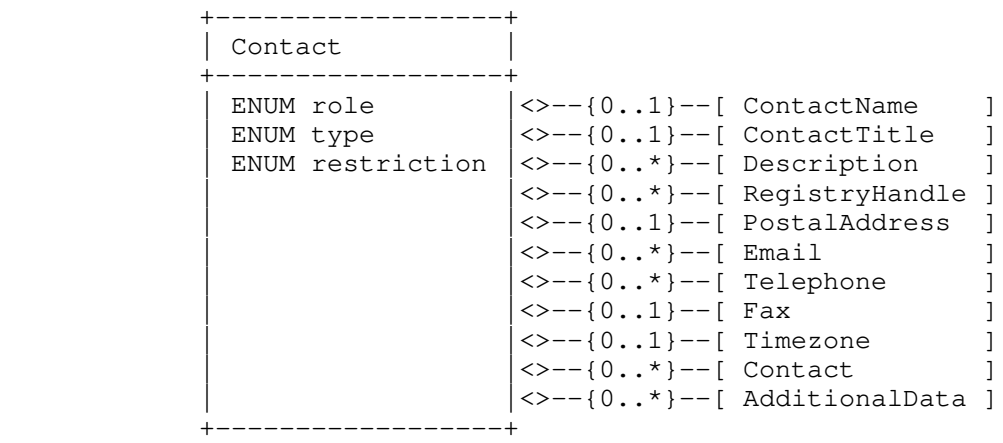

Figure 9: The Contact Class

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 three attributes:

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

  1. creator. The entity that generate the document.

  2. reporter. The entity that reported the information.

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

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

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

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

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

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

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

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

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

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

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

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

15. vendor. The vendor that produces an asset.

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

17. victim. A victim in the incident.

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

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

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

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

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.

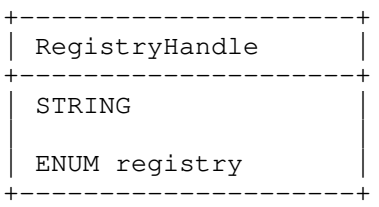

Figure 10: The RegistryHandle Class

The RegistryHandle class has one attributes:

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

1. internic. Internet Network Information Center

2. apnic. Asia Pacific Network Information Center

3. arin. American Registry for Internet Numbers

4. lacnic. Latin-American and Caribbean IP Address Registry

5. ripe. Reseaux IP Europeens

6. afrinic. African Internet Numbers Registry

7. local. A database local to the CSIRT

3.10.2. PostalAddress Class

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

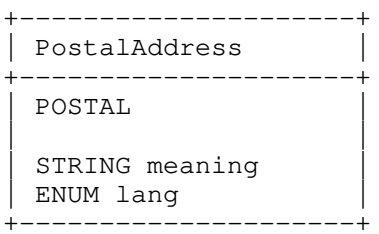

Figure 11: The PostalAddress Class

The PostalAddress class has two attributes:

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

lang
Optional. ENUM. A valid language code per [RFC5646] 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).

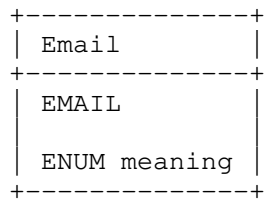

Figure 12: The Email Class

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).

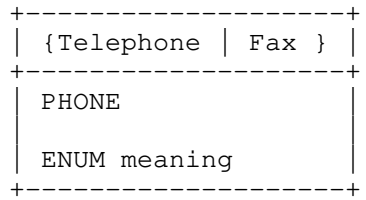

Figure 13: The Telephone and Fax Classes

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).

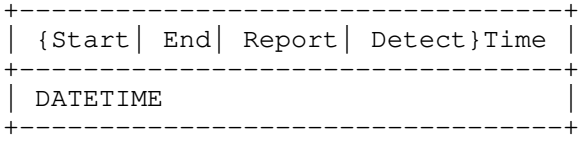

Figure 14: The Time Classes

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.

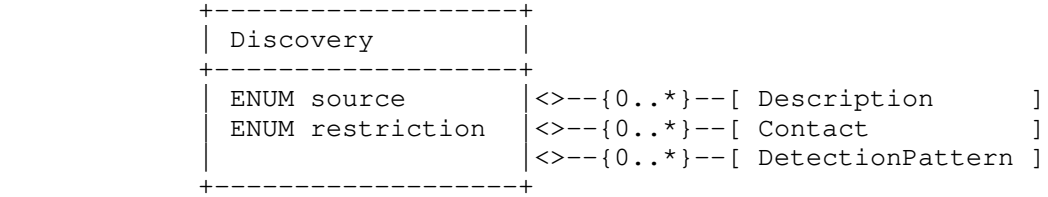

Figure 15: The Discovery Class

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 two attribute:

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

  1. nidps. Network Intrusion Detection or Prevention system.

  2. hips. Host-based Intrusion Prevention system.

  3. siem. Security Information and Event Management System.

  4. av. Antivirus or and antispam software.

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

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

7. os-log. Operating system logs.

8. application-log. Application logs.

9. device-log. Network device logs.

10. network-flow. Network flow analysis.

11. passive-dns. Passive DNS analysis.

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

13. audit. Security audit.

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

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

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

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

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

19. unknown. Unknown detection approach.

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.

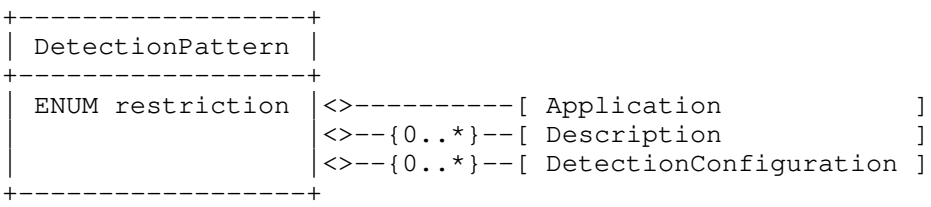

Figure 16: The DetectionPattern Class

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.

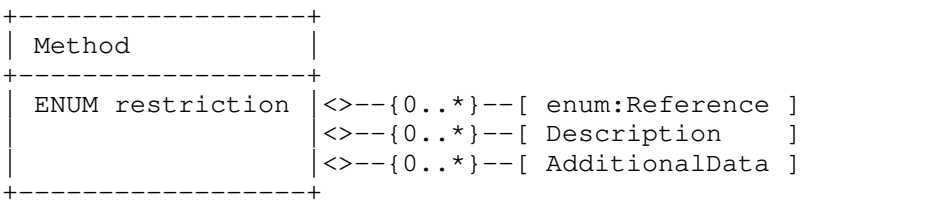

Figure 17: The Method Class

The Method class is composed of three aggregate classes.

enum:Reference
: Zero or more. A reference to a vulnerability, malware sample, advisory, or analysis of an attack technique per [RFC-ENUM].

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.

3.14. Assessment Class

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

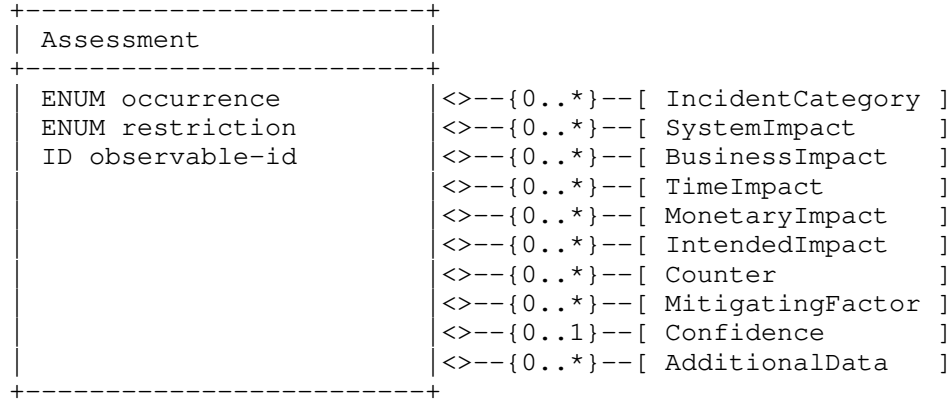

Figure 18: Assessment Class

The aggregate classes that constitute Assessment are:

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

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

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

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

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

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

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

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

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

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

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

The Assessment class has three 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.

observable-id
Optional. ID. See Section 3.3.2.

3.14.1. SystemImpact Class

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

This class is based on [RFC4765].

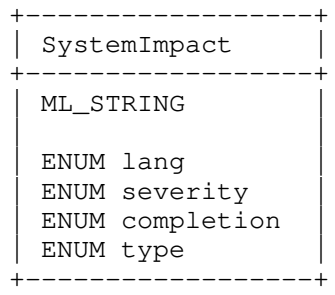

Figure 19: SystemImpact Class

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

The SystemImpact class has four attributes:

lang
Optional. ENUM. A valid language code per [RFC5646] 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 impact. The permitted values are shown below. The default value is "unknown". These values are maintained in the "SystemImpact-type" IANA registry per Table 1.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

23. unknown. The impact is unknown.

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.

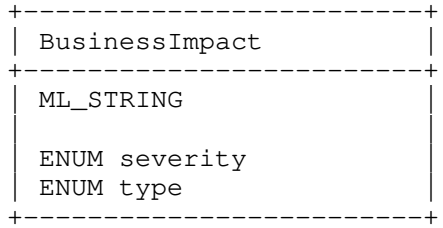

Figure 20: BusinessImpact Class

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

The BusinessImpact class has two 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". These values are maintained in the "BusinessImpact-severity" IANA registry per Table 1.

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

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

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

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

5. unknown. The impact is not known.

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

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

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

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

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

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

6. theft-financial. Money was stolen.

7. theft-service. Services were misappropriated.

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

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

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

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

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

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.

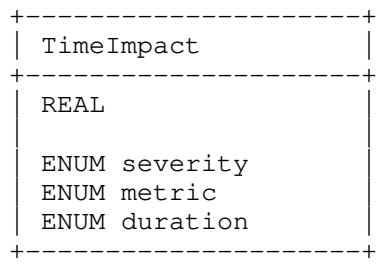

Figure 21: TimeImpact Class

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 three attributes:

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

1. low. Low severity

2. medium. Medium severity

3. high. High severity

metric
Required. ENUM. Defines the metric in which the time is expressed. The permitted values are shown below. There is no default value. These values are maintained in the "TimeImpact-metric" IANA registry per Table 1.

1. labor. Total staff-time to recovery from the activity (e.g., 2 employees working 4 hours each would be 8 hours).

2. elapsed. Elapsed time from the beginning of the recovery to its completion (i.e., wall-clock time).

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

duration
Optional. ENUM. Defines a unit of time, that when combined with the metric attribute, fully describes a metric of impact that will be conveyed in the element content. The permitted values are shown below. The default value is "hour". These values are maintained in the "TimeImpact-duration" IANA registry per Table 1.

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

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

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

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

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

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

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

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.

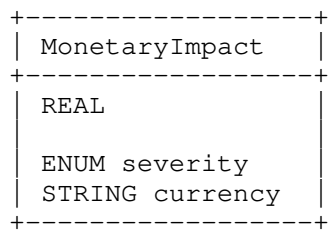

Figure 22: MonetaryImpact Class

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].

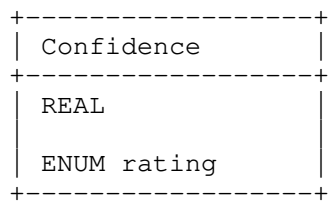

Figure 23: Confidence Class

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.

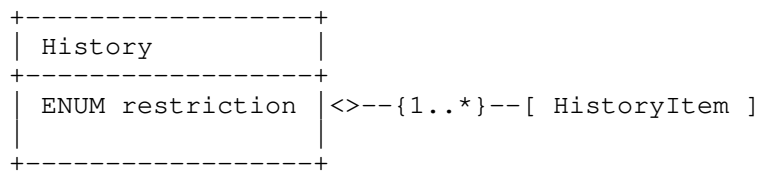

Figure 24: The History Class

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.

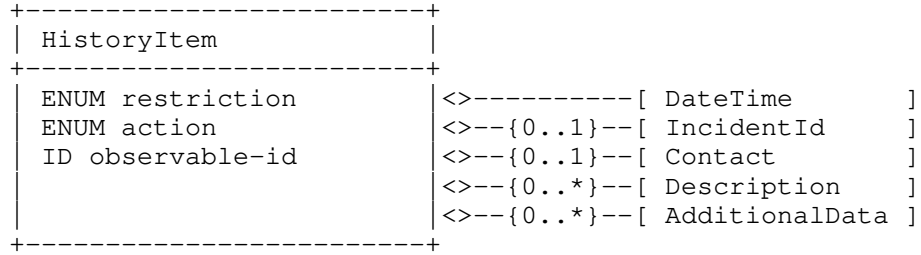

Figure 25: HistoryItem Class

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

observable-id
Optional. ID. See Section 3.3.2.

3.16. EventData Class

The EventData class describes a particular event of the incident for a given set of hosts or networks. This description includes the systems from which the activity originated and those targeted, an assessment of the techniques used by the intruder, the impact of the activity on the organization, and any forensic evidence discovered.

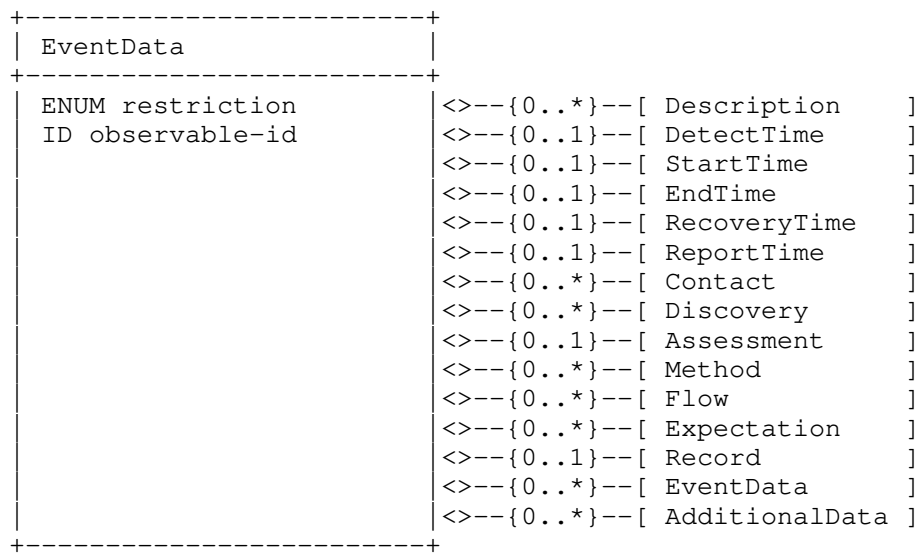

Figure 26: The EventData Class

The aggregate classes that constitute EventData are:

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

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

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

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

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

ReportTime
: One. The time the event was reported.

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

Discovery

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

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

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

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

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

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

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

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

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

The EventData class has two attributes:

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

observable-id
Optional. ID. See Section 3.3.2.

3.16.1. Relating the Incident and EventData Classes

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

3.16.2. Cardinality of EventData

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

The recursive definition (or instance property inheritance) of the EventData class (the EventData class is aggregated into the EventData class) provides a way to relate information without requiring the explicit use of unique attribute identifiers in the classes or duplicating information. Instead, the relative depth (nesting) of a class is used to group (relate) information.

For example, an EventData class might be used to describe two machines involved in an incident. This description can be achieved using multiple instances of the Flow class. It happens that there is a common technical contact (i.e., Contact class) for these two machines, but the impact (i.e., Assessment class) on them is different. A depiction of the representation for this situation can be found in Figure 27.

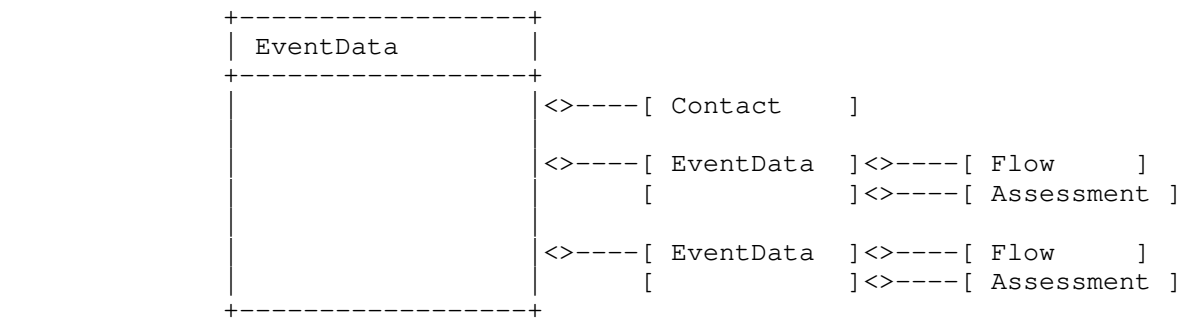

Figure 27: Recursion in the EventData Class

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.

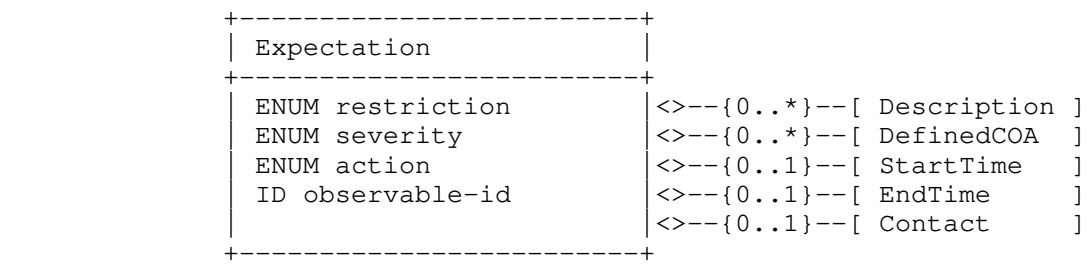

Figure 28: The Expectation Class

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 four 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". These values are maintained in the "Expectation-action" IANA registry per Table 1.

1. nothing. No action is requested. Do nothing with the information.

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

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

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

5. investigate. Investigate the systems(s) listed in the event.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

observable-id
Optional. ID. See Section 3.3.2.

3.18. Flow Class

The Flow class groups related the source and target hosts.

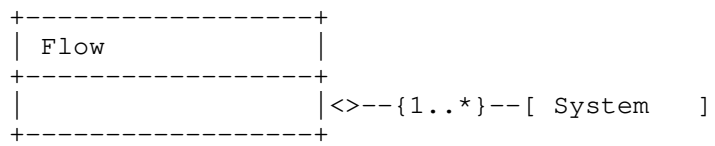

Figure 29: The Flow Class

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.

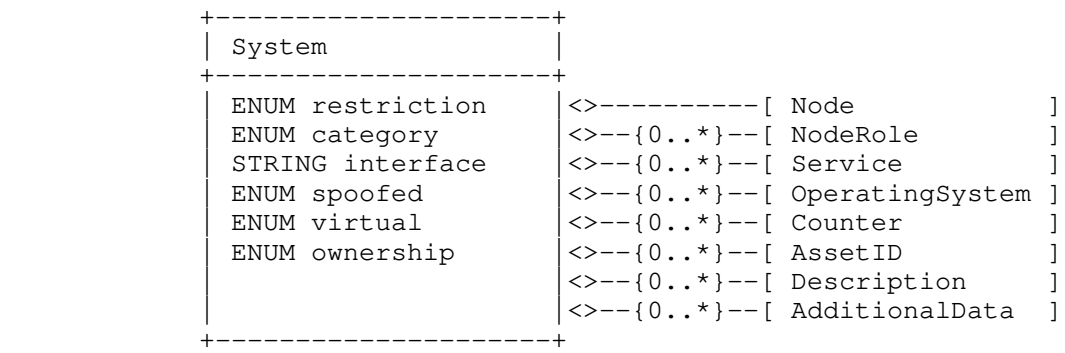

Figure 30: The System Class

The aggregate classes that constitute System are:

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

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

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

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

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

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

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

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

The System class has six 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. These values are maintained in the "System-category" IANA registry per Table 1. The possible values are:

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

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

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

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

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

interface
Optional. STRING. Specifies the interface on which the event(s) on this System originated. If the Node class specifies a network rather than a host, this attribute has no meaning.

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

1. unknown. The accuracy of the category attribute value is unknown.

2. yes. The category attribute value is probably incorrect. In the case of a source, the System is likely a decoy; with a target, the System was likely not the intended victim.

3. no. The category attribute value is believed to be correct.

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

1. yes. The System is a virtual device.

2. no. The System is a physical device.

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

ownership

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

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

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

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

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

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

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

3.20. Node Class

The Node class names an asset or network.

This class was derived from [RFC4765].

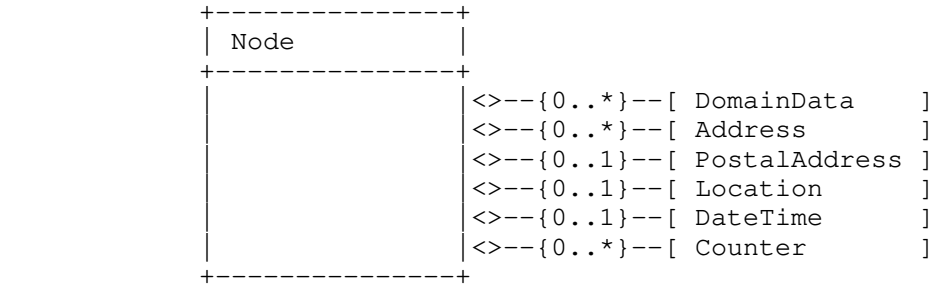

Figure 31: The Node Class

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).

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].

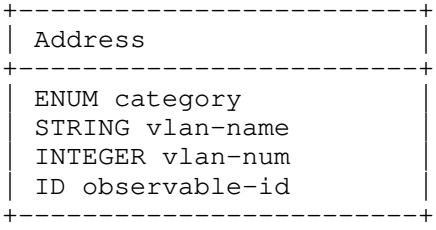

Figure 32: The Address Class

The Address class has four attributes:

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

1. asn. Autonomous System Number

2. atm. Asynchronous Transfer Mode (ATM) address

3. e-mail. Electronic mail address (RFC 822)

4. ipv4-addr. IPv4 host address in dotted-decimal notation (a.b.c.d)

5. ipv4-net. IPv4 network address in dotted-decimal notation, slash, significant bits (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.

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

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

observable-id
Optional. ID. See Section 3.3.2.

3.20.2. NodeRole Class

The NodeRole class describes the function performed by a particular .

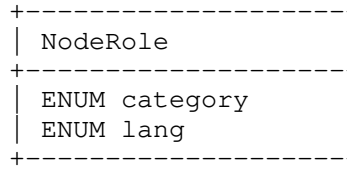

Figure 33: The NodeRole Class

The NodeRole class has two attributes:

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

1. client. Client computer
2. client-enterprise. Client computer on the enterprise network
3. client-partner. Client computer on network of a partner
4. client-remote. Client computer remotely connected to the enterprise network
5. client-kiosk. Client computer is serves as a kiosk
6. client-mobile. Client is a mobile device
7. server-internal. Server with internal services
8. server-public. Server with public services
9. www. WWW server
10. mail. Mail server
11. webmail. Web mail server
12. messaging. Messaging server (e.g., NNTP, IRC, IM)
13. streaming. Streaming-media server
14. voice. Voice server (e.g., SIP, H.323)
15. file. File server (e.g., SMB, CVS, AFS)
16. ftp. FTP server
17. p2p. Peer-to-peer node
18. name. Name server (e.g., DNS, WINS)
19. directory. Directory server (e.g., LDAP, finger, whois)
20. credential. Credential server (e.g., domain controller, Kerberos)
21. print. Print server

22. application. Application server
23. database. Database server
24. backup. Backup server
25. dhcp. DHCP server
26. assessment. Assessment server (e.g., vulnerability scanner, end-point assessment)
27. source-control. Source code control server
28. config-management. Configuration management server
29. monitoring. Security monitoring server (e.g., IDS)
30. infra. Infrastructure server (e.g., router, firewall, DHCP)
31. infra-firewall. Firewall
32. infra-router. Router
33. infra-switch. Switch
34. camera. Camera and video system
35. proxy. Proxy server
36. remote-access. Remote access server
37. log. Log server (e.g., syslog)
38. virtualization. Server running virtual machines
39. pos. Point-of-sale device
40. scada. Supervisory control and data acquisition system
41. scada-supervisory. Supervisory system for a SCADA
42. sinkhole. Traffic sinkhole destination
43. honeypot. Honeypot server
44. anonymization. Anonymization server (e.g., Tor node)
45. c2. Malicious command and control server

46. malware-distribution. Server that distributes malware

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

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

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

50. phishing-site. Site hosting phishing content

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

52. recruiting-site. Site to recruit

53. fraudulent-site. Fraudulent site.

lang
Optional. ENUM. A valid language code per [RFC5646] 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.

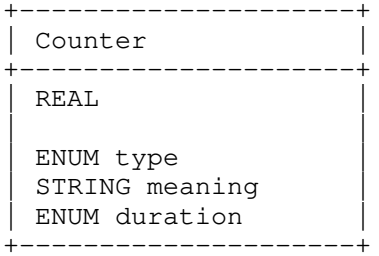

Figure 34: The Counter Class

The Counter class has three attribute:

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

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.

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

3.21. DomainData Class

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

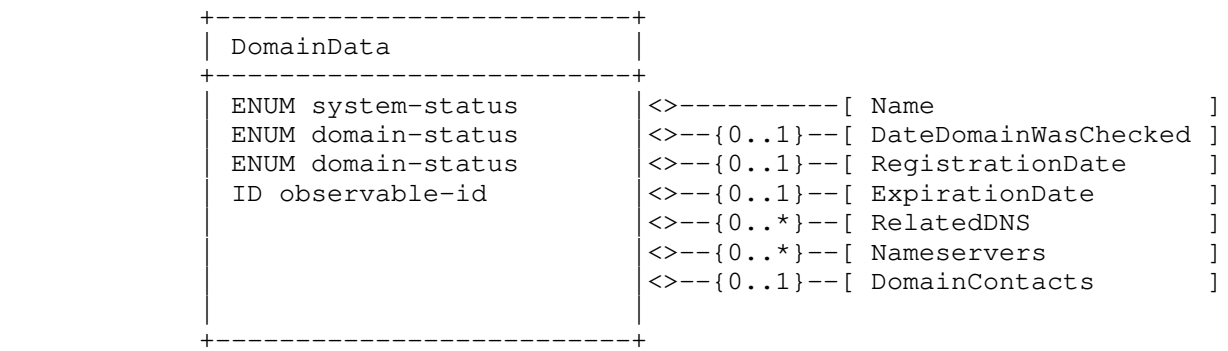

Figure 35: The DomainData Class

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. Additional DNS records associated with this domain.

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

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

The DomainData class has four attribute:

system-status

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

1. spoofed. This domain was spoofed.

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

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

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

5. unknown. No categorization for this domain known.

domain-status

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

1. reservedDelegation. The domain is permanently inactive.

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

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

4. assignedAndOnHold. The domain is under dispute.

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

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

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

8. registrarLock. Same as "registryLock".

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

10. unknown. The domain has an unknown status.

observable-id

Optional. ID. See Section 3.3.2.

3.21.1. RelatedDNS

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

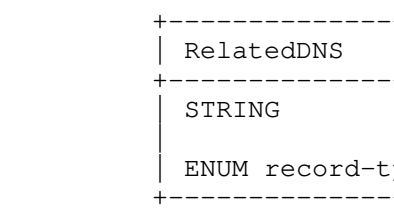

Figure 36: The RelatedDNS Class

The RelatedDNS class has one attribute:

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

3.21.2. Nameservers Class

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

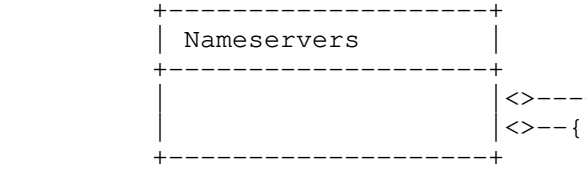

Figure 37: The Nameservers Class

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.

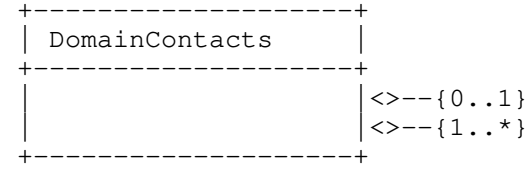

Figure 38: The DomainContacts Class

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].

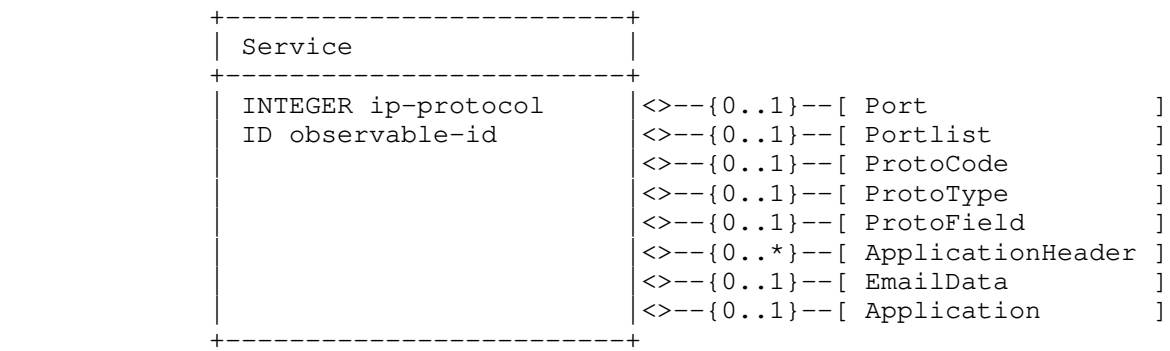

Figure 39: The Service Class

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 two attributes:

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

observable-id
: Optional. ID. See Section 3.3.2.

3.22.1. ApplicationHeader Class

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

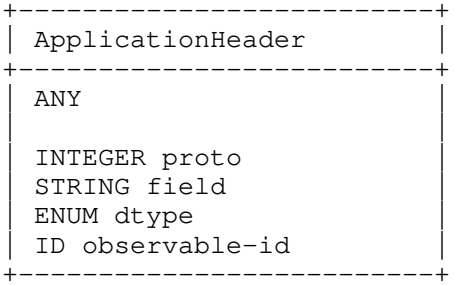

Figure 40: The ApplicationHeader Class

The ApplicationHeader class has four 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.

observable-id
Optional. ID. See Section 3.3.2.

3.22.2. Application Class

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

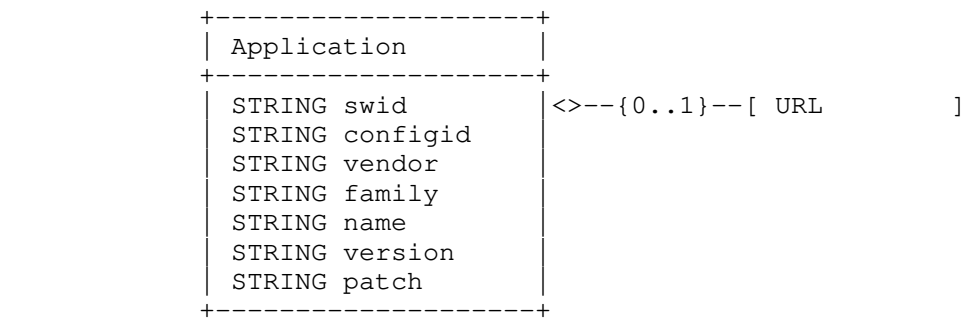

Figure 41: The Application Class

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.

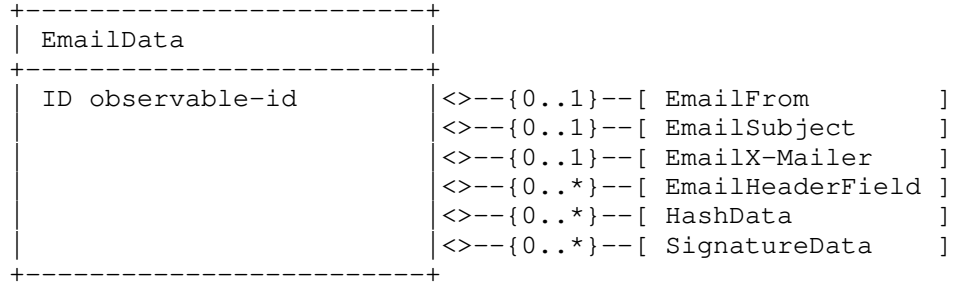

Figure 42: EmailData Class

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.

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

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

The EmailData class has one attribute:

observable-id
Optional. ID. See Section 3.3.2.

3.25. Record Class

The Record class is a container class for log and audit data that provides supportive information about the incident. The source of this data will often be the output of monitoring tools. These logs substantiate the activity described in the document.

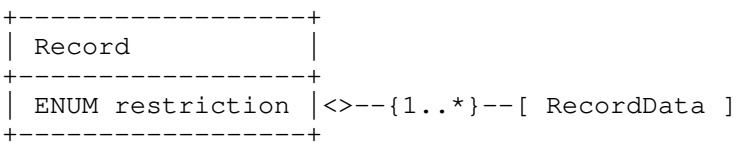

Figure 43: Record Class

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.

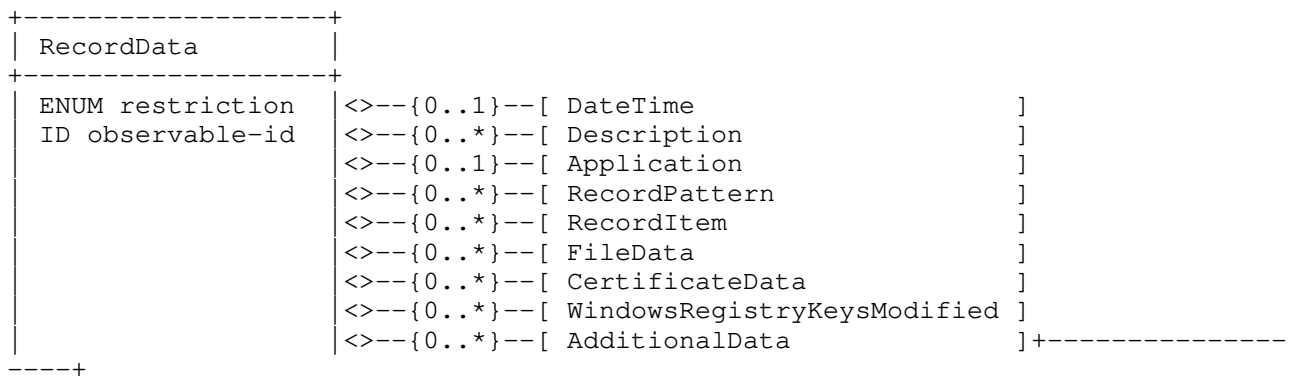

Figure 44: The RecordData Class

The aggregate classes that constitutes RecordData is:

DateTime
: Zero or one. Timestamp of the RecordItem data.

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

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

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

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

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

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

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

The RecordData class has two attribute:

restriction
Optional. ENUM. See Section 3.3.1.

observable-id
Optional. ID. See Section 3.3.2.

3.25.2. RecordPattern Class

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

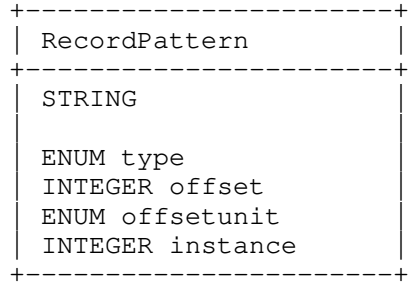

Figure 45: The RecordPattern Class

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

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

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

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

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

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

offsetunit

Optional. ENUM. Describes the units of the offset attribute. The default is "line". These values are maintained in the "RecordPattern-offsetunit" IANA registry per Table 1.

1. line. Offset is a count of lines.

2. byte. Offset is a count of bytes.

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].

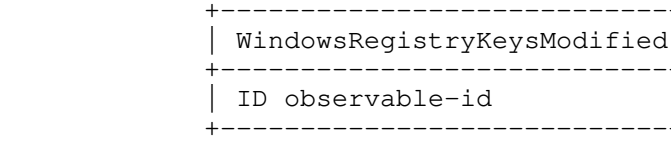

Figure 46: The WindowsRegistryKeysModified Class

The aggregate class that constitutes the WindowsRegistryKeysModified class is:

Key
One or many. The Window registry key.

The WindowsRegistryKeysModified class has one attribute:

observable-id
Optional. ID. See Section 3.3.2.

3.26.1. Key Class

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

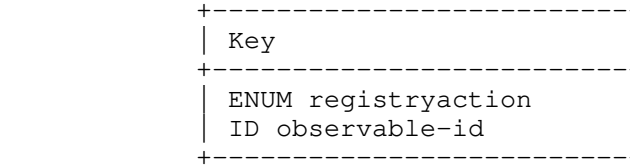

Figure 47: The Key Class

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 two attributes:

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

  1. add-key. Registry key added.

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

  3. delete-key. Registry key deleted.

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

  5. modify-key. Registry key modified.

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

observable-id
: Optional. ID. See Section 3.3.2.

3.27. CertificateData Class

The CertificateData class describes X.509 certificates.

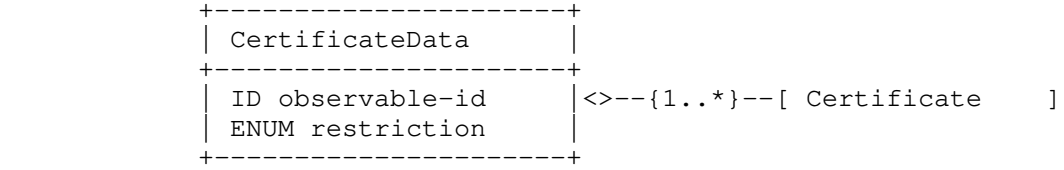

Figure 48: The CertificateData Class

The aggregate classes that constitutes CertificateData are:

Certificate
: One or more. A certificate.

The CertificateData class has two attribute:

observable-id
: Optional. ID. See Section 3.3.2.

restriction
: Optional. ENUM. See Section 3.3.1.

3.27.1. Certificate Class

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

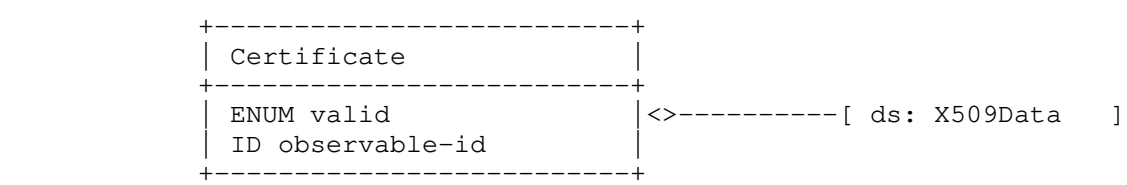

Figure 49: The Certificate Class

The aggregate classes that constitutes Certificate are:

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

The Certificate class has one attribute:

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

1. yes. The certificate is valid.

2. no. The certificate is not valid.

observable-id
Optional. ID. See Section 3.3.2.

3.28. FileData Class

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

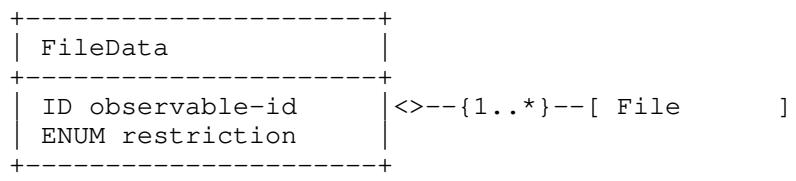

Figure 50: The FileData Class

The aggregate class that constitutes FileData is:

File
One or more. A description of a file.

The FileData class has two attribute:

observable-id
Optional. ID. See Section 3.3.2.

restriction
Optional. ENUM. See Section 3.3.1.

3.28.1. File Class

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

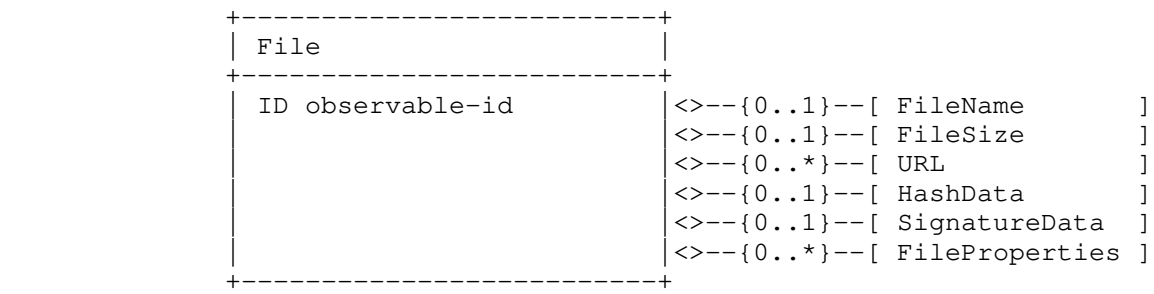

Figure 51: The File Class

The aggregate classes that constitutes File are:

FileName
: Zero or One. ML_STRING. The name of the file.

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

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

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

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

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

The File class has one attribute:

observable-id
: Optional. ID. See Section 3.3.2.

3.29. HashData Class

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

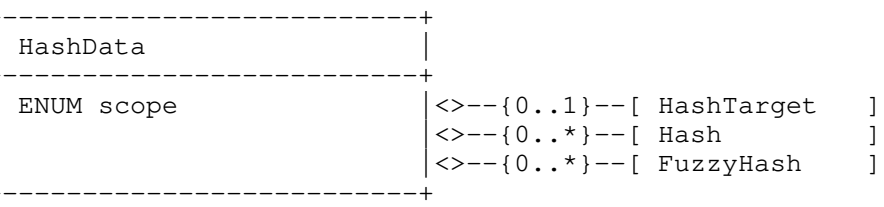

Figure 52: The HashData Class

The aggregate classes that constitutes HashData are:

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

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

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

A single instance of Hash or FuzzyHash MUST be present.

The HashData class has one attribute:

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

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

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

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

  4. file-pe-resource. A hash computed on a given resource in a PE file. If set to this value, the HashTargetId class MUST

identify the resource being hashed. This resource is identified by an ordinal number (starting at 1) corresponding to the oder in which the given resource is declared in the Resource Directory of the Data Dictionary in the PE file header.

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

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

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

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

3.29.1. Hash Class

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

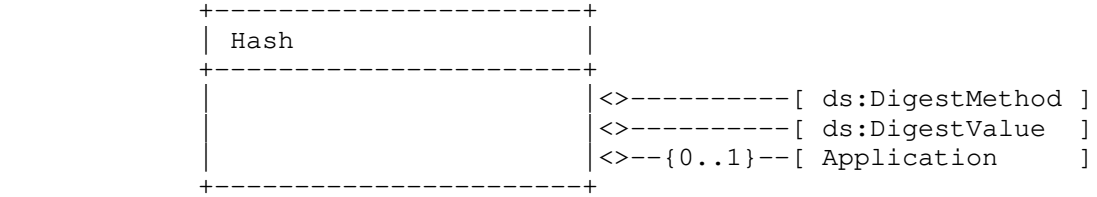

Figure 53: The Hash Class

The aggregate classes that constitutes Hash are:

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

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

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

The HashData class has no attribute:

3.29.2. FuzzyHash Class

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

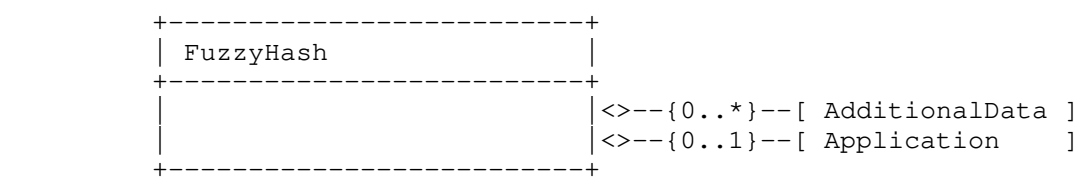

Figure 54: The FuzzyHash Class

The aggregate classes that constitutes FuzzyHash are:

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

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

The FuzzyData class has no attribute:

3.30. SignatureData Class

The SignatureData class describes different signatures on an given object.

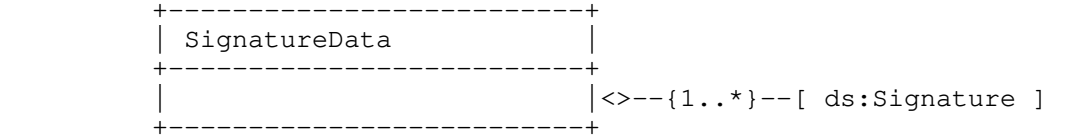

Figure 55: The SignatureData Class

The aggregate classes that constitutes SignatureData are:

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

The SignatureData class has no attribute:

3.31. IndicatorData Class

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

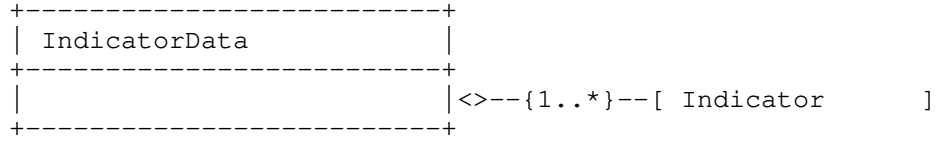

Figure 56: The IndicatorData Class

The aggregate class that constitutes IndicatorData is:

Indicator
One or more. An indicator from the incident.

The IndicatorData class has no attributes.

3.32. Indicator Class

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

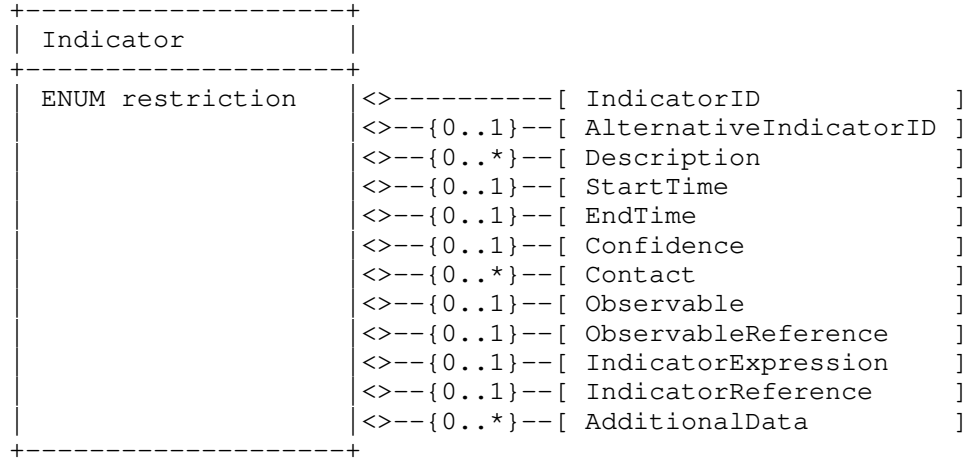

Figure 57: The Indicator Class

The aggregate classes that constitute Indicator are:

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

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

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

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

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

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

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

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

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

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

IndicatorReference
Zero or one. A reference to an indicator.

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

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

The StartTime and EndTime classes can be used to define an interval during which the indicator is valid. If both classes are present, the indicator is consider valid only during the described interval. If neither class is provided, the indicator is considered valid during any time interval. If only a StartTime is provided, the indicator is valid anytime after this timestamp. If only an EndTime is provided, the indicator is valid anytime prior to this timestamp.

The Indicator class has one attribute:

restriction
Optional. ENUM. See Section 3.3.1.

3.32.1. IndicatorID Class

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

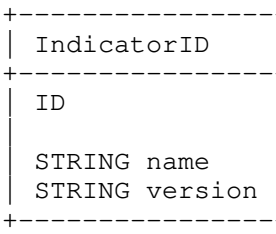

Figure 58: The IndicatorID Class

The IndicatorID class has two attributes:

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

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

3.32.2. AlternativeIndicatorID Class

The AlternativeIndicatorID class lists alternative identifiers for an indicator.

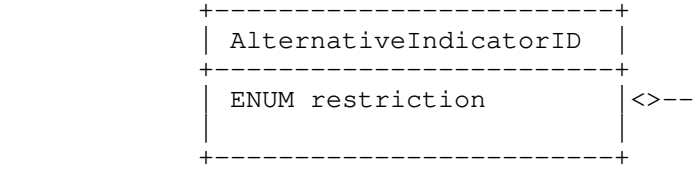

Figure 59: The AlternativeIndicatorID Class

The aggregate class that constitutes AlternativeIndicatorID is:

IndicatorReference
: One or more. A reference to an indicator.

The AlternativeIndicatorID class has one attribute:

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

3.32.3. Observable Class

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

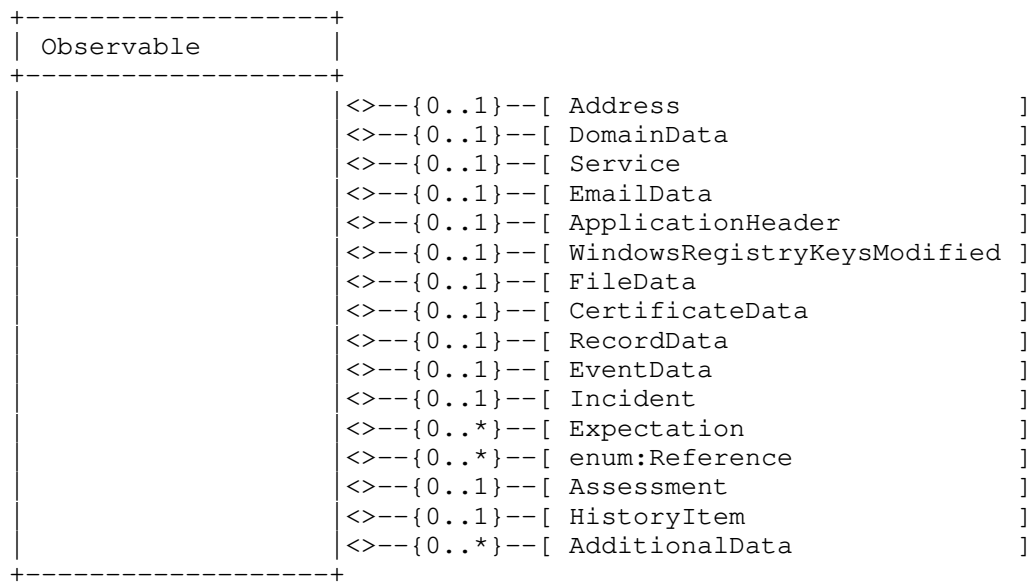

Figure 60: The Observable Class

The aggregate classes that constitute Observable are:

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

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

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

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

ApplicationHeader
: Zero or One. An ApplicationHeader observable. See Section 3.22.1.

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

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

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

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

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

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

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

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

enum:Reference
Zero or One. A Reference observable. See [RFC-ENUM].

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

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

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

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

The Observable class has no attributes.

3.32.4. IndicatorExpression Class

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

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

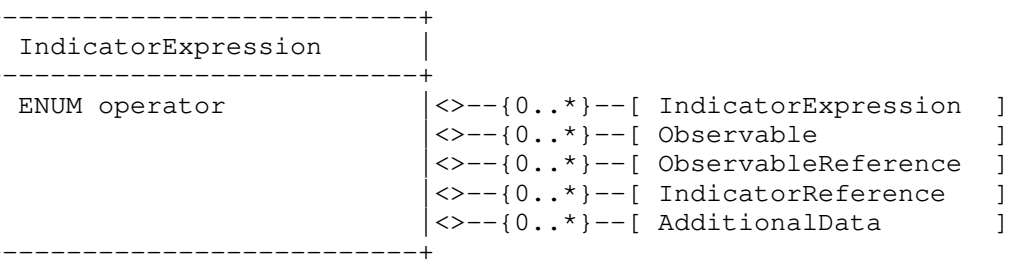

Figure 61: The IndicatorExpression Class

The aggregate classes that constitute IndicatorExpression are:

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

Observable
: Zero or more. A description of an observable.

ObservableReference
: Zero or more. A reference to another observable.

IndicatorReference
: Zero or more. A reference to another indicator.

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

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

The IndicatorExpression class has one attributes:

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

  1. not. negation operator.

  2. and. conjunction operator.

  3. or. disjunction operator.

  4. xor. exclusive disjunction operator.

3.32.5. ObservableReference Class

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

This class has no content.

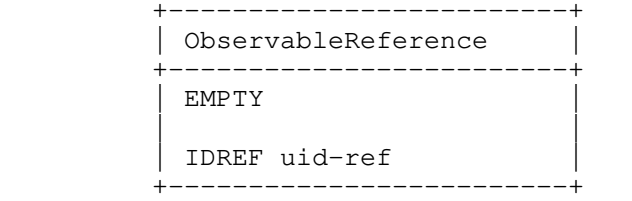

Figure 62: The ObservableReference Class

The ObservableReference class has one attributes:

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

3.32.6. IndicatorReference Class

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

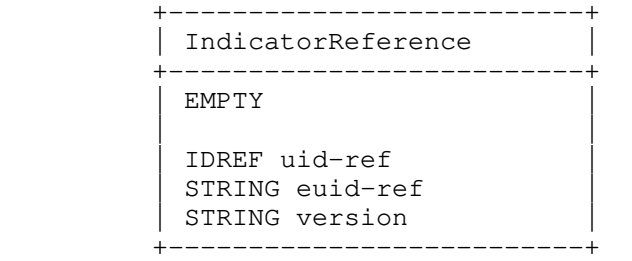

Figure 63: The IndicatorReference Class

The IndicatorReference class has one attributes:

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

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

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

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

4. Processing Considerations

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

4.1. Encoding

Every IODEF document MUST begin with an XML declaration, and MUST specify the XML version used. 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 [RFC5646].

- 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.

- The enumerated values present in this document are a static list that will be incomplete over time as select attributes can be extended by a corresponded IANA registry. See Table 1. Hence, the schema to validate a given document MUST be dynamically generated from these registry values.

4.4. Incompatibilities with v1

Version 2 of the IODEF data model makes a number of changes to [RFC5070]. Largely, these changes were additive in nature -- classes and enumerated values were added. The following is a list of incompatibilities where the data model has changed between versions:

- The IODEF-Document@version attribute is set to "2.0".

- The Service@ip_protocol attribute was renamed to @ip-protocol.

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

- The Node/NodeRole class was moved to System/NodeRole.

- The Reference class is now defined by [RFC-ENUM].

- Extending enumerated values is now handled through collection of IANA registries. All attributes of with a name prefixed by "ext-" have been removed.

- The data previously represented in the Impact class is now in the SystemImpact and IncidentCategory classes. The Impact class has been removed.

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

Select enumerated value of the attributes defined in the data model can be extended by adding entries to the corresponding IANA registry. See Table 1.

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 [RFC5646].

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
          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
            </RecordItem>
              <!-- Additional logs -->
            <RecordItem dtype="url">
               http://mylogs.example.com/logs/httpd_access</RecordItem>
          </RecordData>
        </Record>
      </EventData>
      <History>
        <!-- Contact was previously made with the source network
             owner -->
        <HistoryItem action="contact-source-site">
          <DateTime>2001-09-14T08:19:01+00:00</DateTime>
          <Description>Notification sent to
                       constituency-contact@192.0.2.200</Description>
        </HistoryItem>
      </History>
    </Incident>
  </IODEF-Document>
```

7.2. Reconnaissance

An example of a CSIRT reporting a scanning activity.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- This example describes reconnaissance activity: one-to-one
     and one-to-many scanning -->
<IODEF-Document version="2.00" lang="en"
```

```
     xmlns="urn:ietf:params:xml:ns:iodef-1.0"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="urn:ietf:params:xml:schema:iodef-1.0">
     <Incident purpose="reporting">
       <IncidentID name="csirt.example.com">59334</IncidentID>
       <ReportTime>2006-08-02T05:54:02-05:00</ReportTime>
       <Assessment>
         <Impact type="recon" completion="succeeded" />
       </Assessment>
       <Method>
         <!-- Reference to the scanning tool "nmap" -->
         <Reference>
           <ReferenceName>nmap</ReferenceName>
           <URL>http://nmap.toolsite.example.com</URL>
         </Reference>
       </Method>
       <!-- Organizational contact and that for staff in that
            organization -->
       <Contact role="creator" type="organization">
         <ContactName>CSIRT for example.com</ContactName>
         <Email>contact@csirt.example.com</Email>
         <Telephone>+1 412 555 12345</Telephone>
         <!-- Since this <Contact> is nested, Joe Smith is part of
              the CSIRT for example.com -->
         <Contact role="tech" type="person" restriction="need-to-know">
           <ContactName>Joe Smith</ContactName>
           <Email>smith@csirt.example.com</Email>
         </Contact>
       </Contact>
       <EventData>
         <!-- Scanning activity as follows:
           192.0.2.1:60524 >> 192.0.2.3:137
                  192.0.2.1:60526 >> 192.0.2.3:138
                  192.0.2.1:60527 >> 192.0.2.3:139
                  192.0.2.1:60531 >> 192.0.2.3:445
         -->
         <Flow>
           <System category="source">
             <Node>
               <Address category="ipv4-addr">192.0.2.200</Address>
             </Node>
             <Service ip_protocol="6">
               <Portlist>60524,60526,60527,60531</Portlist>
             </Service>
           </System>
           <System category="target">
             <Node>
               <Address category="ipv4-addr">192.0.2.201</Address>
```

```
            </Node>
            <Service ip_protocol="6">
              <Portlist>137-139,445</Portlist>
            </Service>
          </System>
        </Flow>
        <!-- Scanning activity as follows:
              192.0.2.2 >> 192.0.2.3/28:445 -->
        <Flow>
          <System category="source">
            <Node>
              <Address category="ipv4-addr">192.0.2.240</Address>
            </Node>
          </System>
          <System category="target">
            <Node>
              <Address category="ipv4-net">192.0.2.64/28</Address>
            </Node>
            <Service ip_protocol="6">
              <Port>445</Port>
            </Service>
          </System>
        </Flow>
      </EventData>
    </Incident>
  </IODEF-Document>
```

7.3. Bot-Net Reporting

An example of a CSIRT reporting a bot-network.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- This example describes a compromise and subsequent installation
     of bots -->
<IODEF-Document version="2.00" lang="en"
  xmlns="urn:ietf:params:xml:ns:iodef-1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:ietf:params:xml:schema:iodef-1.0">
  <Incident purpose="mitigation">
    <IncidentID name="csirt.example.com">908711</IncidentID>
    <ReportTime>2006-06-08T05:44:53-05:00</ReportTime>
    <Description>Large bot-net</Description>
    <Assessment>
      <Impact type="dos" severity="high" completion="succeeded" />
    </Assessment>
    <Method>
```

```
        <!-- References a given piece of malware, "GT Bot" -->
        <Reference>
          <ReferenceName>GT Bot</ReferenceName>
        </Reference>
        <!-- References the vulnerability used to compromise the
             machines -->
        <Reference>
          <ReferenceName>CA-2003-22</ReferenceName>
          <URL>http://www.cert.org/advisories/CA-2003-22.html</URL>
          <Description>Root compromise via this IE vulnerability to
                       install the GT Bot</Description>
        </Reference>
      </Method>
      <!-- A member of the CSIRT that is coordinating this
           incident -->
      <Contact type="person" role="irt">
        <ContactName>Joe Smith</ContactName>
        <Email>jsmith@csirt.example.com</Email>
      </Contact>
      <EventData>
        <Description>These hosts are compromised and acting as bots
                     communicating with irc.example.com.</Description>
        <Flow>
          <!-- bot running on 192.0.2.1 and sending DoS traffic at
               10,000 bytes/second -->
          <System category="source">
            <Node>
              <Address category="ipv4-addr">192.0.2.1</Address>
            </Node>
            <Counter type="byte" duration="second">10000</Counter>
            <Description>bot</Description>
          </System>
          <!-- a second bot on 192.0.2.3 -->
          <System category="source">
            <Node>
              <Address category="ipv4-addr">192.0.2.3</Address>
            </Node>
            <Counter type="byte" duration="second">250000</Counter>
            <Description>bot</Description>
          </System>
          <!-- Command-and-control IRC server for these bots-->
          <System category="intermediate">
            <Node>
              <NodeName>irc.example.com</NodeName>
              <Address category="ipv4-addr">192.0.2.20</Address>
              <DateTime>2006-06-08T01:01:03-05:00</DateTime>
            </Node>
            <Description>
```

```
              IRC server on #give-me-cmd channel
            </Description>
          </System>
        </Flow>
        <!-- Request to take these machines offline -->
        <Expectation action="investigate">
          <Description>
            Confirm the source and take machines off-line and
            remediate
          </Description>
        </Expectation>
      </EventData>
    </Incident>
  </IODEF-Document>
```

7.4. Watch List

An example of a CSIRT conveying a watch-list.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- This example demonstrates a trivial IP watch-list -->
<!-- @formatid is set to "watch-list-043" to demonstrate how
     additional semantics about this document could be conveyed
     assuming both parties understood it-->
<IODEF-Document version="2.00" lang="en" formatid="watch-list-043"
  xmlns="urn:ietf:params:xml:ns:iodef-1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:ietf:params:xml:schema:iodef-1.0">
  <Incident purpose="reporting" restriction="private">
    <IncidentID name="csirt.example.com">908711</IncidentID>
    <ReportTime>2006-08-01T00:00:00-05:00</ReportTime>
    <Description>
      Watch-list of known bad IPs or networks
    </Description>
    <Assessment>
      <Impact type="admin" completion="succeeded" />
      <Impact type="recon" completion="succeeded" />
    </Assessment>
    <Contact type="organization" role="creator">
      <ContactName>CSIRT for example.com</ContactName>
      <Email>contact@csirt.example.com</Email>
    </Contact>
    <!-- Separate <EventData> is used to convey
         different <Expectation> -->
    <EventData>
      <Flow>
```

```
          <System category="source">
            <Node>
              <Address category="ipv4-addr">192.0.2.53</Address>
            </Node>
            <Description>Source of numerous attacks</Description>
          </System>
        </Flow>
        <!-- Expectation class indicating that sender of list would
             like to be notified if activity from the host is seen -->
        <Expectation action="contact-sender" />
      </EventData>
      <EventData>
        <Flow>
          <System category="source">
            <Node>
              <Address category="ipv4-net">192.0.2.16/28</Address>
            </Node>
            <Description>
              Source of heavy scanning over past 1-month
            </Description>
          </System>
        </Flow>
        <Flow>
          <System category="source">
            <Node>
              <Address category="ipv4-addr">192.0.2.241</Address>
            </Node>
            <Description>C2 IRC server</Description>
          </System>
        </Flow>
        <!-- Expectation class recommends that these networks
             be filtered -->
        <Expectation action="block-host" />
      </EventData>
    </Incident>
  </IODEF-Document>
```

8. The IODEF Schema

```
<xs:schema targetNamespace="urn:ietf:params:xml:ns:iodef-2.0"
           xmlns="urn:ietf:params:xml:ns:iodef-2.0"
           xmlns:iodef="urn:ietf:params:xml:ns:iodef-2.0"
           xmlns:enum="urn:ietf:params:xml:ns:iodef-enum-1.0"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
           elementFormDefault="qualified"
           attributeFormDefault="unqualified">
```

```
<xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
     schemaLocation="http://www.w3.org/TR/2002/
REC-xmldsig-core-20020212/xmldsig-core-schema.xsd"/>
<xs:import namespace="urn:ietf:params:xml:ns:iodef-enum-1.0"
schemaLocation="http://www.iana.org/xml-registry/schema/iodef-enum-1.0.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:RecoveryTime"
                    minOccurs="0"/>
          <xs:element ref="iodef:ReportTime"/>
          <xs:element ref="iodef:GenerationTime"
                    minOccurs="0"/>
          <xs:element ref="iodef:Description"
                    minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:Discovery"
                    minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:Assessment"
                    maxOccurs="unbounded"/>
          <xs:element ref="iodef:Method"
                    minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:Contact"
                    maxOccurs="unbounded"/>
          <xs:element ref="iodef:EventData"
                    minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:History"
                    minOccurs="0"/>
          <xs:element ref="iodef:AdditionalData"
                    minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="purpose" use="required">
          <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:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="lang"
                    type="xs:language"/>
        <xs:attribute name="restriction"
                    type="iodef:restriction-type" default="private"/>
        <xs:attribute name="observable-id"
                    type="xs:ID" 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>

  <!--
   ========================================================================
   ==  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="leo"/>
              <xs:enumeration value="vendor"/>
              <xs:enumeration value="vendor-services"/>
              <xs:enumeration value="victim"/>
              <xs:enumeration value="victim-notified"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="type" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="person"/>
              <xs:enumeration value="organization"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <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:restriction>
              </xs:simpleType>
            </xs:attribute>
          </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="RecoveryTime"
                type="xs:dateTime"/>
    <xs:element name="GenerationTime"
                type="xs:dateTime"/>
    <xs:element name="Timezone"
                type="iodef:TimezoneType"/>
    <xs:simpleType name="TimezoneType">
      <xs:restriction base="xs:string">
        <xs:pattern value="Z|[\+\-](0[0-9]|1[0-4]):[0-5][0-9]"/>
      </xs:restriction>
    </xs:simpleType>
  <!--
   ================================================================
   ==  History class                                             ==
   ================================================================
  -->
    <xs:element name="History">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="iodef:HistoryItem"
                      maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="restriction"
                      type="iodef:restriction-type"
                      default="default"/>
      </xs: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="observable-id"
                    type="xs:ID" 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="observable-id"
                    type="xs:ID" use="optional"/>
      </xs:complexType>
    </xs:element>

  <!--
   ================================================================
   ==  Discovery class                                           ==
```

```
   ====================================================================
  -->
    <xs:element name="Discovery">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="iodef:Description"
                      minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:Contact"
                      minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:DetectionPattern"
                      minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="source"
                      use="optional" default="unknown">
          <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="nidps"/>
              <xs:enumeration value="hips"/>
              <xs:enumeration value="siem"/>
              <xs:enumeration value="av"/>
              <xs:enumeration value="third-party-monitoring"/>
              <xs:enumeration value="incident"/>
              <xs:enumeration value="os-log"/>
              <xs:enumeration value="application-log"/>
              <xs:enumeration value="device-log"/>
              <xs:enumeration value="network-flow"/>
              <xs:enumeration value="passive-dns"/>
              <xs:enumeration value="investigation"/>
              <xs:enumeration value="audit"/>
              <xs:enumeration value="internal-notification"/>
              <xs:enumeration value="external-notification"/>
              <xs:enumeration value="leo"/>
              <xs:enumeration value="partner"/>
              <xs:enumeration value="actor"/>
              <xs:enumeration value="unknown"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <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="enum: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>

  <!--
   ========================================================================
   ==  Assessment class                                                  ==
   ========================================================================
  -->
    <xs:element name="Assessment">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="IncidentCategory"
                      type="iodef:MLStringType"
                      minOccurs="0" maxOccurs="unbounded"/>
          <xs:choice maxOccurs="unbounded">
            <xs:element ref="iodef:SystemImpact"/>
            <xs:element name="BusinessImpact"
                        type="iodef:BusinessImpactType/>
            <xs:element ref="iodef:TimeImpact"/>
            <xs:element ref="iodef:MonetaryImpact"/>
            <xs:element name="IntendedImpact"
```

```
                    type="iodef:BusinessImpactType/>
        </xs:choice>
        <xs:element ref="iodef:Counter"
                    minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="MitigatingFactor"
                    type="iodef:MLStringType"
                    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="observable-id"
                    type="xs:ID" use="optional"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="SystemImpact">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="iodef:MLStringType">
          <xs:attribute name="severity"
                        type="iodef:severity-type"/>
          <xs:attribute name="completion">
            <xs:simpleType>
              <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="failed"/>
                <xs:enumeration value="succeeded"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
          <xs:attribute name="type"
                        use="optional">
            <xs:simpleType>
              <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="admin"/>
                <xs:enumeration value="takeover-account"/>
                <xs:enumeration value="takeover-service"/>
                <xs:enumeration value="takeover-system"/>
                <xs:enumeration value="cps-manipulation"/>
```

```
                <xs:enumeration value="cps-damage"/>
                <xs:enumeration value="availability-data"/>
                <xs:enumeration value="availibility-account"/>
                <xs:enumeration value="availibility-service"/>
                <xs:enumeration value="availibility-system"/>
                <xs:enumeration value="damaged-system"/>
                <xs:enumeration value="damaged-data"/>
                <xs:enumeration value="breach-proprietary"/>
                <xs:enumeration value="breach-privacy"/>
                <xs:enumeration value="breach-credential"/>
                <xs:enumeration value="breach-configuration"/>
                <xs:enumeration value="integrity-data"/>
                <xs:enumeration value="integrity-configuration"/>
                <xs:enumeration value="integrity-hardware"/>
                <xs:enumeration value="traffic-redirection"/>
                <xs:enumeration value="monitoring-traffic"/>
                <xs:enumeration value="monitoring-host"/>
                <xs:enumeration value="policy"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="BusinessImpactType">
    <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:restriction>
          </xs:simpleType>
        </xs:attribute>
        <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="breach-credential"/>
              <xs:enumeration value="loss-of-integrity"/>
```

```
              <xs:enumeration value="loss-of-service" />
              <xs:enumeration value="theft-financial"/>
              <xs:enumeration value="theft-service"/>
              <xs:enumeration value="degraded-reputation"/>
              <xs:enumeration value="asset-damage"/>
              <xs:enumeration value="asset-manipulation"/>
              <xs:enumeration value="legal"/>
              <xs:enumeration value="extortion"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <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:restriction>
            </xs:simpleType>
          </xs:attribute>
          <xs:attribute name="duration"
                        type="iodef:duration-type"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="MonetaryImpact">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="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:RecoveryTime"
                      minOccurs="0"/>
          <xs:element ref="iodef:ReportTime"
                      minOccurs="0"/>
          <xs:element ref="iodef:Contact"
                      minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:Discovery"
                      minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:Assessment"
                      minOccurs="0"/>
          <xs:element ref="iodef:Method"
                      minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:Flow"
                      minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:Expectation"
```

```
                    minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:Record"
                    minOccurs="0"/>
          <xs:element ref="iodef:EventData"
                    minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:AdditionalData"
                    minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="restriction"
                    type="iodef:restriction-type"
                    default="default"/>
        <xs:attribute name="observable-id"
                    type="xs:ID" 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:NodeRole"
                    minOccurs="0" maxOccurs="unbounded" />
          <xs:element ref="iodef:Service"
                    minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:OperatingSystem"
                    minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:Counter"
```

```
                    minOccurs="0" maxOccurs="unbounded"/>
          <xs:element name="AssetID" type="xs:string"
                    minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:Description"
                    minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:AdditionalData"
                    minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="restriction"
                    type="iodef:restriction-type"/>
        <xs:attribute name="category">
          <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="source"/>
              <xs:enumeration value="target"/>
              <xs:enumeration value="intermediate"/>
              <xs:enumeration value="sensor"/>
              <xs:enumeration value="infrastructure"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <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:restriction>
          </xs:simpleType>
        </xs:attribute>
      </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:restriction>
              </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="vlan-name"
                          type="xs:string"/>
            <xs:attribute name="vlan-num"
                          type="xs:integer"/>
            <xs:attribute name="observable-id"
                          type="xs:ID" use="optional"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
```

```
    </xs:element>

    <xs: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="webmail" />
                  <xs:enumeration value="messaging"/>
                  <xs:enumeration value="streaming"/>
                  <xs:enumeration value="voice"/>
                  <xs:enumeration value="file"/>
                  <xs:enumeration value="ftp"/>
                  <xs:enumeration value="p2p"/>
                  <xs:enumeration value="name"/>
                  <xs:enumeration value="directory"/>
                  <xs:enumeration value="credential"/>
                  <xs:enumeration value="print"/>
                  <xs:enumeration value="application"/>
                  <xs:enumeration value="database"/>
                  <xs:enumeration value="backup"/>
                  <xs:enumeration value="dhcp"/>
                  <xs:enumeration value="assessment"/>
                  <xs:enumeration value="source-control"/>
                  <xs:enumeration value="config-management"/>
                  <xs:enumeration value="monitoring"/>
                  <xs:enumeration value="infra"/>
                  <xs:enumeration value="infra-firewall"/>
                  <xs:enumeration value="infra-router"/>
                  <xs:enumeration value="infra-switch"/>
                  <xs:enumeration value="camera"/>
                  <xs:enumeration value="proxy"/>
                  <xs:enumeration value="remote-access"/>
                  <xs:enumeration value="log"/>
```

```
                  <xs:enumeration value="virtualization"/>
                  <xs:enumeration value="pos"/>
                  <xs:enumeration value="scada"/>
                  <xs:enumeration value="scada-supervisory"/>
                  <xs:enumeration value="sinkhole"/>
                  <xs:enumeration value="honeypot"/>
                  <xs:enumeration value="anonymization"/>
                  <xs:enumeration value="c2-server"/>
                  <xs:enumeration value="malware-distribution"/>
                  <xs:enumeration value="drop-server"/>
                  <xs:enumeration value="hop-point"/>
                  <xs:enumeration value="reflector"/>
                  <xs:enumeration value="phishing-site"/>
                  <xs:enumeration value="spear-phishing-site"/>
                  <xs:enumeration value="recruiting-site"/>
                  <xs:enumeration value="fraudulent-site"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </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="observable-id"
                      type="xs:ID" 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:restriction>
              </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="meaning"
                          type="xs:string" use="optional"/>
            <xs:attribute name="duration"
                          type="iodef:duration-type"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
```

```
  <!--
   ====================================================================
   ==  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:element ref="iodef:HashData"
                    minOccurs="0" />
        <xs:element ref="SignatureData"
                    minOccurs="0" />
       </xs:sequence>
       <xs:attribute name="observable-id"
                     type="xs:ID" use="optional"/>
     </xs:complexType>
   </xs:element>

  <!--
   ====================================================================
   ==   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="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="observable-id"
                type="xs:ID" 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:restriction>
        </xs:simpleType>
      </xs:attribute>
    </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:FileData"
                     minOccurs="0" maxOccurs="unbounded"/>
         <xs:element ref="iodef:WindowsRegistryKeysModified"
                     minOccurs="0" maxOccurs="unbounded"/>
```

```
          <xs:element ref="iodef:CertificateData"
                      minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="iodef:AdditionalData"
                      minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="restriction"
                      type="iodef:restriction-type"/>
        <xs:attribute name="observable-id"
                      type="xs:ID" use="optional"/>
      </xs:complexType>
    </xs:element>

    <xs:element name="RecordPattern">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="type" use="required">
              <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                  <xs:enumeration value="regex"/>
                  <xs:enumeration value="binary"/>
                  <xs:enumeration value="xpath"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="offset"
                          type="xs:integer" use="optional"/>
            <xs:attribute name="offsetunit"
                          use="optional" default="line">
              <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                  <xs:enumeration value="line"/>
                  <xs:enumeration value="byte"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="instance"
                          type="xs:integer" use="optional"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="RecordItem"
                type="iodef:ExtensionType"/>
  <!--
   ========================================================================
   ==  Class to describe Windows Registry Keys                           ==
   ========================================================================
```

```
  -->
    <xs:element name="WindowsRegistryKeysModified">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Key" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
               <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:restriction>
              </xs:simpleType>
            </xs:attribute>
           </xs:complexType>
         </xs:element>
       </xs:sequence>
       <xs:attribute name="observable-id"
                     type="xs:ID" use="optional"/>
     </xs:complexType>
     </xs:element>

<!--
================================================================
==  Classes to describe a file                                ==
================================================================
-->

   <xs:element name="FileData">
     <xs:complexType>
       <xs:sequence>
         <xs:element ref="iodef:File"
                     minOccurs="1" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="observable-id"
                    type="xs:ID" use="optional"/>
        <xs:attribute name="restriction"
                    type="iodef:restriction-type"/>
        </xs:complexType>
```

```
   </xs:element>

   <xs:element name="File">
     <xs:complexType>
       <xs:sequence>
        <xs:element name="FileName" type="iodef:MLStringType"
                    minOccurs="0" />
        <xs:element name="FileSize" type="xs:integer"
                    minOccurs="0" />
        <xs:element ref="iodef:URL"
                    minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="iodef:HashData"
                    minOccurs="0" />
        <xs:element ref="ds:Signature"
                    minOccurs="0" />
        <xs:element ref="iodef:FileProperties"
                    minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="observable-id"
                    type="xs:ID" use="optional"/>
        </xs:complexType>
   </xs:element>

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

<!--
================================================================
==  Classes to describe a hash                                ==
================================================================
-->

  <xs:element name="HashData">
  <xs:complexType>
    <xs:sequence>
        <xs:element name="HashTarget" type="iodef:MLStringType"
                    minOccurs="0"/>
        <xs:element ref="iodef:Hash"
                    minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="iodef:FuzzyHash"
                    minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="scope" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="file-contents"/>
            <xs:enumeration value="file-pe-section"/>
            <xs:enumeration value="file-pe-iat"/>
```

```
            <xs:enumeration value="file-pe-resource"/>
            <xs:enumeration value="file-pdf-object"/>
            <xs:enumeration value="email-hash"/>
            <xs:enumeration value="email-headers-hash"/>
            <xs:enumeration value="email-body-hash"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
    </xs:element>

  <xs:element name="Hash">
   <xs:complexType>
    <xs:sequence>
        <xs:element ref="ds:DigestMethod" />
        <xs:element ref="ds:DigestValue" />
        <xs:element ref="iodef:Application"
                    minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
  </xs:element>

  <xs:element name="FuzzyHash">
   <xs:complexType>
    <xs:sequence>
        <xs:element ref="iodef:AdditionalData" />
        <xs:element ref="iodef:Application"
                    minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
  </xs:element>

<!--
================================================================
==  Classes to describe a signature                           ==
================================================================
-->

  <xs:element name="SignatureData">
   <xs:complexType>
    <xs:sequence>
        <xs:element ref="ds:Signature"
                    maxOccurs="unbounded" />
      </xs:sequence>
   </xs:complexType>
  </xs:element>

<!--
```

```
================================================================
==  Classes to describe a certficate                          ==
================================================================
-->

  <xs:element name="CertificateData">
   <xs:complexType>
      <xs:sequence>
        <xs:element ref="iodef:Certificate"
                    maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="observable-id"
                    type="xs:ID" use="optional"/>
      <xs:attribute name="restriction"
                    type="iodef:restriction-type"/>
    </xs:complexType>
   </xs:element>

  <xs:element name="Certificate">
   <xs:complexType>
    <xs:sequence>
        <xs:element ref="ds:X509Data" />
    </xs:sequence>
    <xs:attribute name="virtual" type="yes-no-type"
                  use="optional" />
    <xs:attribute name="observable-id"
                  type="xs:ID" use="optional"/>
   </xs:complexType>
  </xs:element>

  <!--
   ================================================================
   ==  Classes that describe software                            ==
   ================================================================
  -->
    <xs:complexType name="SoftwareType">
      <xs:sequence>
        <xs:element ref="iodef: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"/>
      <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"/>

  <!--
   ================================================================
   == IndicatorData classes                                      ==
   ================================================================
-->
   <xs:element name="IndicatorData">
     <xs:complexType>
       <xs:sequence>
         <xs:element ref="iodef:Indicator"
                     minOccurs="1" maxOccurs="unbounded"/>
        </xs:sequence>
     </xs:complexType>
   </xs:element>

   <xs:element name="Indicator">
     <xs:complexType>
       <xs:sequence>
         <xs:element ref="iodef:IndicatorID" />
         <xs:element ref="iodef:AlternativeIndicatorID"
                     minOccurs="0" maxOccurs="unbounded"/>
         <xs:element ref="iodef:Description"
                     minOccurs="0" maxOccurs="unbounded"/>
         <xs:element ref="iodef:StartTime"
                     minOccurs="0" />
         <xs:element ref="iodef:EndTime"
                     minOccurs="0" />
         <xs:element ref="iodef:Confidence"
                     minOccurs="0" />
         <xs:element ref="iodef:Contact"
                     minOccurs="0" maxOccurs="unbounded"/>
         <xs:choice>
           <xs:element ref="iodef:Observable" />
           <xs:element ref="iodef:ObservableReference" />
           <xs:element ref="iodef:IndicatorExpression" />
           <xs:element ref="iodef:IndicatorReference" />
         </xs:choice>
```

```
      </xs:sequence>
      <xs:attribute name="restriction"
                    type="iodef:restriction-type"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="IndicatorID">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:ID">
          <xs:attribute name="name"
                        type="xs:string" use="required"/>
          <xs:attribute name="version"
                        type="xs:string" use="required"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

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

  <xs:element name="Observable">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="iodef:Address"
                    minOccurs="0"/>
        <xs:element ref="iodef:DomainData"
                    minOccurs="0"/>
        <xs:element ref="iodef:EmailData"
                    minOccurs="0"/>
        <xs:element name="ApplicationHeader"
                    type="iodef:ApplicationHeaderType"
                    minOccurs="0"/>
        <xs:element ref="iodef:WindowsRegistryKeysModified"
                    minOccurs="0"/>
        <xs:element ref="iodef:FileData"
                    minOccurs="0"/>
        <xs:element ref="iodef:RecordData"
                    minOccurs="0"/>
```

```
        <xs:element ref="iodef:EventData"
                    minOccurs="0"/>
        <xs:element ref="iodef:Incident"
                    minOccurs="0"/>
        <xs:element ref="iodef:Expectation"
                    minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="enum:Reference"
                    minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="iodef:Assessment"
                    minOccurs="0"/>
        <xs:element ref="iodef:HistoryItem"
                    minOccurs="0"/>
        <xs:element ref="iodef:AdditionalData"
                    minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="restriction"
                  type="iodef:restriction-type"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="IndicatorExpression">
    <xs:complexType>
      <xs:sequence>
        <xs:choice>
          <xs:element ref="iodef:IndicatorExpression"
                      minOccurs="0"/>
          <xs:element ref="iodef:Observable"
                      minOccurs="0" />
          <xs:element ref="iodef:ObservableReference"
                      minOccurs="0"/>
          <xs:element ref="iodef:IndicatorReference"
                      minOccurs="0"/>
        </xs:choice>
        <xs:element ref="iodef:AlternativeIndicatorID"
                    minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="operator" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
           <xs:enumeration value="not"/>
           <xs:enumeration value="and"/>
           <xs:enumeration value="or"/>
           <xs:enumeration value="xor"/>
         </xs:restriction>
       </xs:simpleType>
     </xs:attribute>
    </xs:complexType>
  </xs:element>
```

```
   <xs:element name="ObservableReference">
     <xs:complexType>
       <xs:attribute name="uid-ref"
                     type="xs:IDREF" use="required"/>
     </xs:complexType>
   </xs:element>

   <xs:element name="IndicatorReference">
     <xs:complexType>
       <xs:attribute name="uid-ref"
                     type="xs:IDREF" use="optional"/>
       <xs:attribute name="euid-ref"
                     type="xs:string" use="optional"/>
       <xs:attribute name="version"
                     type="xs:string" use="optional"/>
     </xs:complexType>
   </xs:element>
<!--
   ========================================================================
   == Miscellaneous simple classes                                       ==
   ========================================================================
  -->
    <xs:element name="Description"
                type="iodef:MLStringType"/>
    <xs:element name="URL"
                type="xs:anyURI"/>
  <!--
   ========================================================================
   == 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="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="observable-id"
                    type="xs:ID" use="optional"/>
    </xs:complexType>

  <!--
   ================================================================
   == Global attribute type declarations                         ==
   ================================================================
  -->
    <xs:simpleType name="yes-no-type">
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="yes"/>
        <xs:enumeration value="no"/>
      </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="yes-no-unknown-type">
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="yes"/>
        <xs:enumeration value="no"/>
        <xs:enumeration value="unknown"/>
      </xs:restriction>
    </xs:simpleType>
```

```
    <xs:simpleType name="restriction-type">
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="default"/>
        <xs:enumeration value="public"/>
        <xs:enumeration value="partner"/>
        <xs:enumeration value="need-to-know"/>
        <xs:enumeration value="private"/>
        <xs:enumeration value="white"/>
        <xs:enumeration value="green"/>
        <xs:enumeration value="amber"/>
        <xs:enumeration value="red"/>
      </xs: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:restriction>
    </xs:simpleType>

    <xs:simpleType name="action-type">
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="nothing"/>
        <xs:enumeration value="contact-source-site"/>
        <xs:enumeration value="contact-target-site"/>
        <xs:enumeration value="contact-sender"/>
        <xs:enumeration value="investigate"/>
        <xs:enumeration value="block-host"/>
        <xs:enumeration value="block-network"/>
        <xs:enumeration value="block-port"/>
        <xs:enumeration value="rate-limit-host"/>
        <xs:enumeration value="rate-limit-network"/>
        <xs:enumeration value="rate-limit-port"/>
        <xs:enumeration value="redirect-traffic"/>
        <xs:enumeration value="honeypot"/>
```

```
        <xs:enumeration value="upgrade-software"/>
        <xs:enumeration value="rebuild-asset"/>
        <xs:enumeration value="harden-asset"/>
        <xs:enumeration value="remediate-other"/>
        <xs:enumeration value="status-triage"/>
        <xs:enumeration value="status-new-info"/>
        <xs:enumeration value="watch-and-report"/>
        <xs:enumeration value="defined-coa"/>
        <xs:enumeration value="other"/>
      </xs: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: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:restriction>
    </xs:simpleType>
</xs:schema>
```

9. Security Considerations

The IODEF data model itself does not directly introduce security issues. Rather, it simply defines a representation for incident information. As the data encoded by the IODEF might be considered privacy sensitive by the parties exchanging the information or by those described by it, care needs to be taken in ensuring the appropriate disclosure during both document exchange and subsequent processing. The former must be handled by a messaging format, but the latter risk must be addressed by the systems that process, store, and archive IODEF documents and information derived from them.

Executable content could be embedded into the IODEF document directly or through an extension. The IODEF parser should handle this content with care to prevent unintentional automated execution.

The contents of an IODEF document may include a request for action or an IODEF parser may independently have logic to take certain actions based on information that it finds. For this reason, care must be taken by the parser to properly authenticate the recipient of the document and ascribe an appropriate confidence to the data prior to action.

The underlying messaging format and protocol used to exchange instances of the IODEF MUST provide appropriate guarantees of confidentiality, integrity, and authenticity. The use of a standardized security protocol is encouraged. The Real-time Inter-network Defense (RID) protocol [RFC6545] and its associated transport binding IODEF/RID over HTTP/TLS [RFC6546] provide such security.

In order to suggest data processing and handling guidelines of the encoded information, the IODEF allows a document sender to convey a privacy policy using the restriction attribute. The various instances of this attribute allow different data elements of the document to be covered by dissimilar policies. While flexible, it must be stressed that this approach only serves as a guideline from the sender, as the recipient is free to ignore it. The issue of enforcement is not a technical problem.

10. IANA Considerations

This document registers a namespace, XML schema, and a number of registries that map to enumerated values defined in the schema.

10.1. Namespace and Schema

This document uses URNs to describe an XML namespace and schema conforming to a registry mechanism described in [RFC3688]

Registration for the IODEF namespace:

- 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.

10.2. Enumerated Value Registries

This document creates xx identically structured registries to be managed by IANA:

- Name of the parent registry: "Incident Object Description Exchange Format v2 (IODEF)"
- URL of the registry: http://www.iana.org/assignments/iodef2
- Namespace format: A registry entry consists of:
  - Value. An enumerated value for a given IODEF attribute.
  - Description. A short description of the enumerated value.
  - Reference. An optional list of URIs to further describe the value.
- Allocation policy: Expert Review per [RFC5226]

The registries to be created are named in the table below in the "Registry Name" column. The initial values for the Value and Description fields of a given registry are listed in the "IV (Value)" and "IV (Description)" columns respectively. The "IV (Value)" points

to a given schema attribute or type per Section 8. Each enumerated value in the schema gets a corresponding entry in a given registry. The "IV (Description)" points to a section in the text of this document. The initial value of the Reference field of every registry entry described below should be this document.

Registry Name	IV (Value)	IV (Description)
Restriction	iodef-restriction-type	Section 3.3.1
Incident-purpose	Incident@purpose	Section 3.2
Contact-role	Contact@role	Section 3.10
Contact-type	Contact@type	Section 3.10
RegistryHandle-registry	RegistryHandle@registry	Section 3.10.1
Expectation-action	iodef:action-type	Section 3.17
Discovery-source	Discovery@source	Section 3.12
SystemImpact-type	SystemImpact@type	Section 3.14.1
BusinessImpact-severity	BusinessImpact@severity	Section 3.14.2
BusinessImpact-type	BusinessImpact@type	Section 3.14.2
TimeImpact-metrics	TimeImpact@metric	Section 3.14.3
TimeImpact-duration	iodef:duration-type	Section 3.14.3
NodeRole-category	NodeRole@category	Section 3.20.2
System-category	System@category	Section 3.19
System-ownership	System@ownership	Section 3.19
Address-category	Address@category	Section

		3.20.1
Counter-type	Counter@type	Section 3.20.3
DomainData-system-status	DomainData@system-status	Section 3.21
DomainData-domain-status	DomainData@domain-status	Section 3.21
RelatedDNS-record-type	RelatedDNS@record-type	Section 3.21.1
RecordPattern-type	RecordPattern@type	Section 3.25.2
RecordPattern-offsetunit	RecordPattern@offsetunit	Section 3.25.2
Key-registryaction	Key@registryaction	Section 3.26.1
HashData-scope	HashData@scope	Section 3.29
AdditionalData-dtype	iodef:dtype-type	Section 3.9
EmailHeaderField-proto-dtype	iodef:proto-dtype-type	Section 3.22.1

Table 1: IANA Enumerated Value Registries

11. Acknowledgments

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

- 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/>.

[W3C.XMLSIG] World Wide Web Consortium, "XML Signature Syntax and Processing 2.0", W3C Candidate Recommendation , June 2008, <http://www.w3.org/TR/xmldsig-core/>.

[IEEE.POSIX] Institute of Electrical and Electronics Engineers, "Information Technology - Portable Operating System Interface (POSIX) - Part 1: Base Definitions", IEEE 1003.1, June 2001.

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

[RFC5646] Philips, A. and M. Davis, "Tags for Identifying of Languages", RFC 5646, September 2009.

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

[RFC2978] Freed, N. and J. Postel, "IANA Charset Registration Procedures", BCP 2978, October 2000.

[RFC4519] Sciberras, A., "Schema for User Applications", RFC 4519, June 2006.

[RFC5322] Resnick, P., "Internet Message Format", RFC 5322, October 2008.

[RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.

[RFC-ENUM] Montville, A. and D. Black, "IODEF Enumeration Reference Format", RFC ENUM, November 2014.

[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.

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

Authors' Addresses

Roman Danyliw
CERT - Software Engineering Institute
Pittsburgh, PA
USA

EMail: rdd@cert.org

Paul Stoecker
RSA
Reston, VA
USA

EMail: paul.stoecker@rsa.com

MILE Working Group
Internet-Draft
Intended status: Informational
Expires: June 8, 2015

J. Field
EMC

December 5, 2014

Resource-Oriented Lightweight Indicator Exchange
draft-ietf-mile-rolie-00.txt

Abstract

This document defines a resource-oriented approach to cyber security information sharing. Using this approach, a CSIRT or other stakeholder may share and exchange representations of cyber security incidents, indicators, and other related information as Web-addressable resources. The transport protocol binding is specified as HTTP(S) with a MIME media type of Atom+XML. An appropriate set of link relation types specific to cyber security information sharing is defined. The resource representations leverage the existing IODEF [RFC5070] and RID [RFC6545] specifications as appropriate. Coexistence with deployments that conform to existing specifications including RID [RFC6545] and Transport of Real-time Inter-network Defense (RID) Messages over HTTP/TLS [RFC6546] is supported via appropriate use of HTTP status codes.

Table of Contents

1. Introduction

This document defines a resource-oriented approach to cyber security information sharing that follows the REST (Architectural Styles and t he Design of Network-based Software Architectures) architectural style. The resource representations leverage the existing IODEF [RFC5070] and RID [RFC6545] specifications as appropriate. The transport protocol binding is specified as HTTP(S) with a media type of Atom+XML. An appropriate set of link relation types specific to cyber security information sharing is defined. Using this approach, a CSIRT or other stakeholder may exchange cyber security incident and/or indicator information as Web-addressable resources.

The goal of this specification is to define a loosely-coupled, agile approach to cyber security situational awareness. This approach has architectural advantages for some use case scenarios, such as when a CSIRT or other stakeholder is required to share cyber security information broadly (e.g., at internet scale), or when an information sharing consortium requires support for asymmetric interactions amongst their stakeholders.

Coexistence with deployments that conform to existing specifications including RID [RFC6545] and Transport of Real-time Inter-network Defense (RID) Messages over HTTP/TLS [RFC6546] is supported via appropriate use of HTTP status codes.

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 [RFC5070].

3. Background and Motivation

It is well known that Internet security threats are evolving ever more rapidly, and are becoming ever more sophisticated than before. The threat actors are frequently distributed and are not constrained to operating within a fixed, closed consortium. The technical skills needed to perform effective analysis of a security incident, or to even recognize an indicator of compromise are already specialized and relatively scarce. As threats continue to evolve, even an established network of CSIRT may find that it does not always have all of the skills and knowledge required to immediately identify and respond to every new incident. Effective identification of and response to a sophisticated, multi-stage attack frequently depends upon cooperation and collaboration, not only amongst the defending CSIRTs, but also amongst other stakeholders, including, potentially, individual end users.

Existing approaches to cyber security information sharing are based upon message exchange patterns that are point-to-point, and event-driven. Sometimes, information that may be useful to, and sharable with multiple peers is only made available to peers after they have specifically requested it. Unfortunately, a sharing peer may not know, a priori, what information to request from another peer. Sending unsolicited RID reports does provide a mechanism for alerting, however these reports are again sent point-to-point, and must be reviewed for relevance and then prioritized for action by the recipient. Thus, distribution of some relevant incident and indicator information may exhibit significant latency.

In order to appropriately combat the evolving threats, the defending CSIRTs should be enabled to operate in a more agile manner, sharing selected cyber security information proactively, if and as appropriate.

For example, a CSIRT analyst would benefit by having the ability to search a comprehensive collection of indicators that has been published by a government agency, or by another member of a sharing consortium. The representation of each indicator may include links to the related resources, enabling an appropriately authenticated and authorized analyst to freely navigate the information space of indicators, incidents, and other cyber security domain concepts, as needed. In general, a more Web-centric sharing approach will enable a more dynamic and agile collaboration amongst a broader, and varying constituency.

The following sections discuss additional specific technical issues that motivate the development of an alternative approach.

3.1. Message-oriented versus Resource-oriented Architecture

The existing approaches to cyber security information sharing are based upon message-oriented interactions. The following paragraphs explore some of the architectural constraints associated with message-oriented interactions and consider the relative merits of an alternative model based on a Resource-oriented architecture for use in some use case scenarios.

3.1.1. Message-oriented Architecture

In general, message-based integration architectures may be based upon either an RPC-style or a document-style binding. The message types defined by RID represent an example of an RPC-style request. This approach imposes implied requirements for conversational state management on both of the communicating RID endpoint(s). Experience has shown that this state management frequently becomes the limiting factor with respect to the runtime scalability of an RPC-style architecture.

In addition, the practical scalability of a peer-to-peer message-based approach will be limited by the administrative procedures required to manage O(N^2) trust relationships and at least O(N) policy groups.

As long as the number of CSIRTs participating in an information sharing consortium is limited to a relatively smaller number of nodes (i.e., O(2^N), where N < 5), these scalability constraints may not represent a critical concern. However, when there is a requirement to support a significantly larger number of participating peers, a different architectural approach will be required. One alternative to the message-based approach that has demonstrated scalability is the REST [REST] architectural style.

3.1.2. Resource-Oriented Architecture

Applying the REST architectural style to the problem domain of cyber security information sharing would take the approach of exposing incidents, indicators, and any other relevant types as simple Web-addressable resources. By using this approach, a CSIRT or other organization can more quickly and easily share relevant incident and indicator information with a much larger and potentially more diverse constituency. A client may leverage virtually any available HTTP user agent in order to make requests of the service provider. This

improved ease of use could enable more rapid adoption and broader participation, thereby improving security for everyone.

A key interoperability aspect of any RESTful Web service will be the choices regarding the available resource representations. For example, clients may request that a given resource representation be returned as either XML or JSON. In order to enable back-compatibility and interoperability with existing CSIRT implementations, IODEF [RFC5070] is specified for this transport binding as a mandatory to implement (MTI) data representation for incident and indicator resources. In addition to the REQUIRED representation, an implementation MAY support additional representations if and as needed such as IODEF extensions, the RID schema, or other schemas. For example, an implementation may choose to provide support for returning a JSON representation of an incident resource.

Finally, an important principle of the REST architectural style is the use of hypertext links as the embodiment of application state (HATEOAS). Rather than the server maintaining conversational state for each client context, the server will instead include a suitable set of hyperlinks in the resource representation that is returned to the client. In this way, the server remains stateless with respect to a series of client requests. The included hyperlinks provide the client with a specific set of permitted state transitions. Using these links the client may perform an operation, such as updating or deleting the resource representation. The client may also be provided with hypertext links that can be used to navigate to any related resource. For example, the resource representation for an incident object may contain links to the related indicator resource(s).

This document specifies the use of Atom Syndication Format [RFC4287] and Atom Publishing Protocol [RFC5023] as the mechanism for representing the required hypertext links.

3.1.2.1. A Resource-Oriented Use Case: "Mashup"

In this section we consider a non-normative example use case scenario for creating a cyber security "mashup".

Any CSIRT can enable any authenticated and authorized client that is a member of the sharing community to quickly and easily navigate through any of the cyber security information that that provider is willing to share. An authenticated and authorized analyst may then make HTTP(S) requests to collect incident and indicator information known at one CSIRT with threat actor data being made available from another CSIRT. The resulting correlations may yield new insights

that enable a more timely and effective defensive response. Of course, this report may, in turn, be made available to others as a new Web-addressable resource, reachable via another URL. By employing the RESTful Web service approach the effectiveness of the collaboration amongst a consortium of CSIRTs and their stakeholders can be greatly improved.

3.2. Authentication of Users

In the store-and-forward, message-based model for information sharing client authentication is provided via a Public Key Infrastructure (PKI) -based trust and mutually authenticated TLS between the messaging system endpoints. There is no provision to support authentication of a client by another means. As a result, participation in the sharing community is limited to those organizations that have sufficient resources and capabilities to manage a PKI.

A CSIRT may apply XML Security to the content of a message, however the contact information provided within the message body represents a self-asserted identity, and there is no guarantee that the contact information will be recognized by the peer. As a result, the audit trail and the granularity of any authorization policies is limited to the identity of the peer CSIRT organization.

A CSIRT implementing this specification MUST implement server-authenticated TLS. The CSIRT may choose to authenticate its client users via any suitable authentication scheme that can be implemented via HTTP(S). A participating CSIRT MAY choose to support more than one authentication method. Support for use of a Federated Identity approach is RECOMMENDED. Establishing a specific end user identity prior to processing a request is RECOMMENDED. Doing so will enable the source system to maintain a more complete audit trail of exactly what cyber security incident and indicator information has been shared, when, and with whom.

3.3. Authorization Policy Enforcement

A key aspect of any cyber security information sharing arrangement is assigning the responsibility for authorization policy enforcement. The authorization policy must be enforced either at the destination system, or the source system, or both. The following sections discuss these alternatives in greater detail.

3.3.1. Enforcement at Destination System

The store-and-forward, message-based approach to cyber security information sharing requires that the origin system delegate authorization policy enforcement to the destination system. The origin system may leverage XML Encryption and DigitalSignature to protect the message content. In addition, the origin system assigns a number of policy-related attribute values, including a "restriction" attribute, before the message is sent. These labels indicate the sender's expectation for confidentiality enforcement and appropriate handling at the destination. Section 9.1 of RFC6545 provides specific guidance to implementers on use of the XML security standards in order to achieve the required levels of security for the exchange of incident information.

Once the message has been received at the destination system, the XML encryption and digital signature protections on the message will be processed, and based upon the pre-established PKI-based trust relationships, the message content is validated and decrypted. Typical implementations will then pass the cleartext data to an internal Incident Handling System (IHS) for further review and/or action by a human operator or analyst. Regardless of where in the deployment architecture the XML message-level security is being handled, eventually the message content will be made available as cleartext for handling by human systems analysts and other operational staff.

The authorization policy enforcement of the message contents must then be provided by the destination IHS. It is the responsibility of the destination system to honor the intent of the policy restriction labels assigned by the origin system. Ideally, these policy labels would serve as part of a distributed Mandatory Access Control scheme. However, in practice a typical IHS will employ a Discretionary Access Control (DAC) model rather than a MAC model and so the policy related attributes are defined to represent handling "hints" and provide no guarantee of enforcement at the destination.

As a result, ensuring that the destination system or counterparty will in fact correctly enforce the intended authorization policies becomes a key issue when entering into any information sharing agreements. The origin CSIRT must accept a non-zero risk of information leakage, and therefore must rely upon legal recourse as a compensating control. Establishing such legal sharing agreements can be a slow and difficult process, as it assumes a high level of trust in the peer, with respect to both intent and also technical capabilities.

3.3.2. Enforcement at Source System

In this model, the required authorization policy enforcements are implemented entirely within the source system. Enforcing the required authorization policy controls at the source system eliminates the risk of subsequent information leakage at the destination system due to inadequate or incomplete implementation of the expected controls. The destination system is not expected to perform any additional authorization enforcements. Authorization enforcement at the source system may be based on, e.g. Role-based Access Controls applied in the context of an established user identity. The source system may use any appropriate authentication mechanism in order to determine the user identity of the requestor, including, e.g. federated identity. An analyst or operator at a CSIRT may request specific information on a given incident or indicator from a peer CSIRT, and the source system will return a suitable representation of that resource based upon the specific role of the requestor. A different authenticated user (perhaps from the same destination CSIRT) may receive a different representation of the same resource, based upon the source system applying suitable Role-based Access Control policy enforcements for the second user identity.

Consistent with HTTP [RFC2616] a user's request MAY be denied with a resulting HTTP status code value of 4xx such as 401 Unauthorized, 403 Forbidden, or 404 Not Found, or 405 Method Not Allowed, if and as appropriate.

4. RESTful Usage Model

This section describes the basic use of Atom Syndication Format [RFC4287] and Atom Publishing Protocol [RFC5023] as a RESTful transport binding and dynamic discovery protocol, respectively, for cyber security information sharing.

As described in Atom Publishing Protocol [RFC5023], an Atom Service Document is an XML-based document format that allows a client to dynamically discover the collections provided by a publisher.

As described in Atom Syndication Format [RFC4287], Atom is an XML-based document format that describes lists of related information items known as collections, or "feeds". Each feed document contains a collection of zero or more related information items called "member entries" or "entries".

When applied to the problem domain of cyber security information sharing, an Atom feed may be used to represent any meaningful collection of information resources such as a set of incidents, or

indicators. Each entry in a feed could then represent an individual incident, or indicator, or some other resource, as appropriate. Additional feeds could be used to represent other meaningful and useful collections of cyber security resources. A feed may be categorized, and any feed may contain information from zero or more categories. The naming scheme and the semantic meaning of the terms used to identify an Atom category are application-defined.

4.1. Dynamic Service Discovery versus Static URL Template

In order to specify a protocol for cyber security information sharing using the REST architectural style it is necessary to define the set of resources to be modeled, and how these resources are related. Based on this interface contract, clients will then interact with the REST service by navigating the modeled entities, and their relationships. The interface contract between the client and the server may either be statically bound or dynamically bound.

In the statically bound case, the clients have a priori knowledge of the resources that are supported. In the REST architectural style this static interface contract takes the form of a URL template. This approach is not appropriate for the cyber security information sharing domain for at least two reasons.

First, there is no standard for a cyber security domain model. While information security practitioners can generally agree on some of the basic concepts that are important to modeling the cyber security domain -- such as "indicator," "incident," or "attacker," -- there is no single domain model that can been referenced as the basis for specifying a standardized RESTful URI Template. Second, the use of static URL templates creates a tighter coupling between the client implementation and the server implementation. Security threats on the internet are evolving ever more rapidly, and it will never be possible to establish a statically defined resource model and URL Template. Even if there were an initial agreement on an appropriate URL template, it would eventually need to change. If and when a CSIRT finds that it needs to change the URL template, then any existing deployed clients would need to be upgraded.

Thus, rather than attempting to define a fixed set of resources via a URI Template, this document has instead specified an approach based on dynamic discovery of resources via an Atom Publishing Protocol Service Document. By using this approach, it is possible to standardize the RESTful usage model, without needing to standardize on the definitions of specific, strongly-typed resources. A client can dynamically discover what resources are provided by a given CSIRT, and then navigate that domain model accordingly A specific server implementation may still embody a particular URL template,

however the client does not need a priori knowledge of the format of the links, and the URL itself is effectively opaque to the client. Clients are not bound to any particular server's interface.

The following paragraphs provide a number of non-normative examples to illustrate the use of Atom Publishing Protocol for basic cyber security information sharing service discovery, as well as the use of Atom Syndication Format as a mechanism to publish cyber security information feeds.

Normative requirements are defined below, in Section 5.

4.2. Non-Normative Examples

4.2.1. Service Discovery

This section provides a non-normative example of a client doing service discovery.

An Atom service document enables a client to dynamically discover what feeds a particular publisher makes available. Thus, a CSIRT may use an Atom service document to enable clients of the CSIRT to determine what specific cyber security information the CSIRT makes available to the community. The service document could be made available at any well known location, such as via a link from the CSIRT's home page. One common technique is to include a link in the <HEAD> section of the organization's home page, as shown below:

Example of bootstrapping Service Document discovery:

```
   <link rel="introspection"  type="application/atomsvc+xml" title="Atom Publ
ishing Protocol Service Document" href="/csirt/svcdoc.xml" />
```

A client may then format an HTTP GET request to retrieve the service document:

```
GET /csirt/svcdoc.xml
Host: www.example.org
Accept: application/atomsvc+xml
```

Notice the use of the HTTP Accept: request header, indicating the MIME type for Atom service discovery. The response to this GET request will be an XML document that contains information on the specific feed collections that are provided by the CSIRT.

Example HTTP GET response:

```
HTTP/1.1 200 OK
Date: Fri, 24 Aug 2012 17:09:11 GMT
Content-Length: 570
Content-Type: application/atomsvc+xml;charset="utf-8"

<?xml version="1.0" encoding="UTF-8"?>
<service xmlns="http://www.w3.org/2007/app"
         xmlns:atom="http://www.w3.org/2005/Atom">
    <workspace xml:lang="en-US" xmlns:xml="http://www.w3.org/XML/1998/name
space">
      <atom:title type="text">Incidents</atom:title>
      <collection  href="http://example.org/csirt/incidents">
         <atom:title type="text">Incidents Feed</atom:title>
         <accept>application/atom+xml; type=entry</accept>
      </collection>
    </workspace>
</service>
```

This simple Service Document example shows that this CSIRT provides one workspace, named "Incidents." Within that workspace, the CSIRT makes one feed collection available. When attempting to GET or POST entries to that feed collection, the client must indicate a content type of application/atom+xml.

A CSIRT may also offer a number of different feeds, each containing different types of cyber security information. In the following example, the feeds have been categorized. This categorization will help the clients to decide which feeds will meet their needs.

```
      HTTP/1.1 200 OK
      Date: Fri, 24 Aug 2012 17:10:11 GMT
      Content-Length: 1912
      Content-Type: application/atomsvc+xml;charset="utf-8"

       <?xml version="1.0" encoding='utf-8'?>
          <service xmlns="http://www.w3.org/2007/app"
            xmlns:atom="http://www.w3.org/2005/Atom">
            <workspace>
              <atom:title>Cyber Security Information Sharing</atom:title>
              <collection href="http://example.org/csirt/public/indicators" >
                <atom:title>Public Indicators</atom:title>
                <categories fixed="yes">
                  <atom:category scheme="http://example.org/csirt/restriction" t
erm="public" />
                  <atom:category scheme="http://example.org/csirt/purpose" term=
"reporting" />
                </categoies>
                <accept>application/atom+xml; type=entry</accept>
              </collection>
              <collection href="http://example.org/csirt/public/incidents" >
                <atom:title>Public Incidents</atom:title>
                <categories fixed="yes">
                  <atom:category scheme="http://example.org/csirt/restriction" t
erm="public" />
                  <atom:category scheme="http://example.org/csirt/purpose" term=
"reporting" />
                </categoies>
                <accept>application/atom+xml; type=entry</accept>
            </collection>
            </workspace>
            <workspace>
              <atom:title>Private Consortium Sharing</atom:title>
              <collection href="http://example.org/csirt/private/incidents" >
                <atom:title>Incidents</atom:title>
                <accept>application/atom+xml;type=entry</accept>
                <categories fixed="yes">
                  <atom:category scheme="http://example.org/csirt/purpose" term=
"traceback, mitigation, reporting" />
                  <atom:category scheme="http://example.org/csirt/restriction" t
erm="private, need-to-know" />
                </categories>
              </collection>
            </workspace>
          </service>
```

In this example, the CSIRT is providing a total of three feed collections, organized into two different workspaces. The first workspace contains two feeds, consisting of publicly available indicators and publicly available incidents, respectively. The second workspace provides one additional feed, for use by a sharing consortium. The feed contains incident information containing entries related to three purposes: traceback, mitigation, and

reporting. The entries in this feed are categorized with a restriction of either "Need-to-Know" or "private". An appropriately authenticated and authorized client may then proceed to make GET requests for one or more of these feeds. The publicly provided incident information may be accessible with or without authentication. However, users accessing the feed targeted to the private sharing consortium would be expected to authenticate, and appropriate authorization policies would subsequently be enforced by the feed provider.

4.2.2. Feed Retrieval

This section provides a non-normative example of a client retrieving an incident feed.

Having discovered the available cyber security information sharing feeds, an authenticated and authorized client who is a member of the private sharing consortium may be interested in receiving the feed of known incidents. The client may retrieve this feed by performing an HTTP GET operation on the indicated URL.

Example HTTP GET request for a Feed:

```
GET /csirt/private/incidents
Host: www.example.org
Accept: application/atom+xml
```

The corresponding HTTP response would be an XML document containing the incidents feed:

Example HTTP GET response for a Feed:

```
      HTTP/1.1 200 OK
      Date: Fri, 24 Aug 2012 17:20:11 GMT
      Content-Length: 2882
      Content-Type: application/atom+xml;type=feed;charset="utf-8"

      <?xml version="1.0" encoding="UTF-8"?>
      <feed xmlns="http://www.w3.org/2005/Atom"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://www.w3.org/2005/Atom file:/C:/schemas/atom.
xsd
                              urn:ietf:params:xml:ns:iodef-1.0 file:/C:/schemas/
iodef-1.0.xsd"
          xml:lang="en-US">

          <generator version="1.0" xml:lang="en-US">emc-csirt-iodef-feed-service
</generator>
          <id xml:lang="en-US">http://www.example.org/csirt/private/incidents</i
d>
          <title type="text" xml:lang="en-US">Atom formatted representation of a
 feed of IODEF documents</title>
          <updated xml:lang="en-US">2012-05-04T18:13:51.0Z</updated>
          <author>
              <email>csirt@example.org</email>
              <name>EMC CSIRT</name>
          </author>

          <!-- By convention there is usually a self link for the feed -->
          <link href="http://www.example.org/csirt/private/incidents" rel="self"
/>

          <entry>
              <id>http://www.example.org/csirt/private/incidents/123456</id>
              <title>Sample Incident</title>
              <link href="http://www.example.org/csirt/private/incidents/123456"
 rel="self"/>       <!-- by convention -->
              <link href="http://www.example.org/csirt/private/incidents/123456"
 rel="alternate"/>  <!-- required by Atom spec -->
              <published>2012-08-04T18:13:51.0Z</published>
              <updated>2012-08-05T18:13:51.0Z</updated>
              <!-- The category is based upon IODEF purpose and restriction attr
ibutes -->
              <category term="traceback" scheme="purpose" label="trace back" />
              <category term="need-to-know" scheme="restriction" label="need to
know" />
              <summary>A short description of this incident, extracted from the
IODEF Incident class, <description> element. </summary>
          </entry>

          <entry>
              <!-- ...another entry... -->
          </entry>

      </feed>
```

This feed document has two atom entries, one of which has been elided. The completed entry illustrates an Atom <entry> element that provides a summary of essential details about one particular

incident. Based upon this summary information and the provided category information, a client may choose to do an HTTP GET operation to retrieve the full details of the incident. This example provides a RESTful alterntive to the RID investigation request messaage, as described in sections 6.1 and 7.2 of RFC6545.

4.2.3. Entry Retrieval

This section provides a non-normative example of a client retrieving an incident as an Atom entry.

Having retrieved the feed of interest, the client may then decide based on the description and/or category information that one of the entries in the feed is of further interest. The client may retrieve this incident Entry by performing an HTTP GET operation on the indicated URL.

Example HTTP GET request for an Entry:

```
GET /csirt/private/incidents/123456
Host: www.example.org
Accept: application/atom+xml
```

The corresponding HTTP response would be an XML document containing the incident:

Example HTTP GET response for an Entry:

```
   HTTP/1.1 200 OK
   Date: Fri, 24 Aug 2012 17:30:11 GMT
   Content-Length: 4965
   Content-Type: application/atom+xml;type=entry;charset="utf-8"

   <?xml version="1.0" encoding="UTF-8"?>
   <entry>
     <id>http://www.example.org/csirt/private/incidents/123456</id>
     <title>Sample Incident</title>
     <link href="http://www.example.org/csirt/private/incidents/123456" rel="
self"/>       <!-- by convention -->
     <link href="http://www.example.org/csirt/private/incidents/123456" rel="
alternate"/>  <!-- required by Atom spec -->
     <published>2012-08-04T18:13:51.0Z</published>
     <updated>2012-08-05T18:13:51.0Z</updated>
     <!-- The category is based upon IODEF purpose and restriction attributes
 -->
     <category term="traceback" scheme="purpose" label="trace back" />
     <category term="need-to-know" scheme="restriction" label="need to know"
/>
     <summary>A short description of this incident, extracted from the IODEF
Incident class, <description> element. </summary>
```

```
        <!-- Refer to section 5.9 for the list of supported (cyber information-s
pecific) link relationships -->
        <!-- Typical operations that can be performed on this IODEF message incl
ude edit -->
        <link href="http://www.example.org/csirt/private/incidents/123456" rel="
edit"/>

        <!-- the next and previous are just sequential access, may not map to an
ything related to this IODEF Incident ID -->
        <link href="http://www.example.org/csirt/private/incidents/123457" rel="
next"/>
        <link href="http://www.example.org/csirt/private/incidents/123455" rel="
previous"/>

        <!-- navigate up to the full collection.  Might also be rel="collection"
 as per IANA registry -->
        <link href="http://www.example.org/csirt/private/incidents" rel="up"/>

        <content type="application/xml">
          <iodef:IODEF-Document lang="en" xmlns:iodef="urn:ietf:params:xml:ns:io
def-1.0">
            <iodef:Incident purpose="traceback" restriction="need-to-know">

              <!-- Note that the ID is assigned using a namespace that is our ba
se URL, so that it can also be leveraged as an Atom link -->
              <iodef:IncidentID name="http://www.example.org/csirt/private/incid
ents">123456</iodef:IncidentID>

              <iodef:DetectTime>2004-02-02T22:49:24+00:00</iodef:DetectTime>
              <iodef:StartTime>2004-02-02T22:19:24+00:00</iodef:StartTime>
              <iodef:ReportTime>2004-02-02T23:20:24+00:00</iodef:ReportTime>
              <iodef:Description>
                Host involved in DoS attack
              </iodef:Description>
              <iodef:Assessment>
                <iodef:Impact completion="failed" severity="low" type="dos"/>
              </iodef:Assessment>
              <iodef:Contact role="creator" type="organization">
                <iodef:ContactName>Constituency-contact for 192.0.2.35
                </iodef:ContactName>
                <iodef:Email>Constituency-contact@192.0.2.35</iodef:Email>
              </iodef:Contact>
              <iodef:EventData>
                <iodef:Flow>
                  <iodef:System category="source">
                    <iodef:Node>
                      <iodef:Address category="ipv4-addr">192.0.2.35
                      </iodef:Address>
                    </iodef:Node>
                    <iodef:Service ip_protocol="6">
                      <iodef:Port>38765</iodef:Port>
                    </iodef:Service>
                  </iodef:System>
                  <iodef:System category="target">
                    <iodef:Node>
                      <iodef:Address category="ipv4-addr">192.0.2.67
                      </iodef:Address>
                    </iodef:Node>
```

```
                    <iodef:Service ip_protocol="6">
                      <iodef:Port>80</iodef:Port>
                    </iodef:Service>
                  </iodef:System>
                </iodef:Flow>
                <iodef:Expectation action="rate-limit-host" severity="high">
                  <iodef:Description>
                    Rate-limit traffic close to source
                  </iodef:Description>
                </iodef:Expectation>
                <iodef:Record>
                  <iodef:RecordData>
                    <iodef:Description>
                      The IPv4 packet included was used in the described attack
                    </iodef:Description>
                    <iodef:RecordItem dtype="ipv4-packet">450000522ad9
                      0000ff06c41fc0a801020a010102976d0050103e020810d9
                      4a1350021000ad6700005468616e6b20796f7520666f7220
                      6361726566756c6c792072656164696e6720746869732052
                      46432e0a
                    </iodef:RecordItem>
                  </iodef:RecordData>
                </iodef:Record>
              </iodef:EventData>
            </iodef:Incident>
          </iodef:IODEF-Document>
        </content>
      </entry>
```

As can be seen in the example response, above, an IODEF document is contained within the Atom <content> element. The client may now process the IODEF document as needed.

Note also that, as described previously, the content of the Atom <category> element is application-defined. In the present context, the Atom categories have been assigned based on a mapping of the <restriction> and <purpose> attributes, as defined in the IODEF schema. In addition, the IODEF <incidentID> element has been judiciously chosen so that the associated name attribute, as well as the corresponding incidentID value, can be concatenated in order to easily create the corresponding <id> element for the Atom entry. These and other mappings are normatively defined in Section 5, below.

Finally, it should be noted that in order to optimize the client experience, and avoid an additional round trip, a feed provider may choose to include the entry content inline, as part of the feed document. That is, an Atom <entry> element within a Feed document

may contain an Atom <content> element as a child. In this case, the client will receive the full content of the entries within the feed. The decision of whether to include the entry content inline or to include it as a link is a design choice left to the feed provider (e.g. based upon local environmental factors such as the number of entries contained in a feed, the available network bandwidth, the available server compute cycles, the expected client usage patterns, etc.).

4.2.4. Use of Link Relations

As noted previously, a key benefit of using the RESTful architectural style is the ability to enable the client to navigate to related resources through the use of hypermedia links. In the Atom Syndication Format, the type of the related resource identified in a <link> element is indicated via the "rel" attribute, where the value of this attribute identifies the kind of related resource available at the corresponding "href" attribute. Thus, in lieu of a well-known URI template the URI itself is effectively opaque to the client, and therefore the client must understand the semantic meaning of the "rel" attribute in order to successfully navigate. Broad interoperability may be based upon a sharing consortium defining a well-known set of Atom Link Relation types. These Link Relation types may either be registered with IANA, or held in a private registry.

Individual CSIRTs may always define their own link relation types in order to support specific use cases, however support for a core set of well-known link relation types is encouraged as this will maximize interoperability.

In addition, it may be beneficial to define use case profiles that correspond to specific groupings of supported link relationship types. In this way, a CSIRT may unambiguously specify the classes of use cases for which a client can expect to find support.

The following sections provide NON-NORMATIVE examples of link relation usage. Four distinct cyber security information sharing use case scenarios are described. In each use case, the unique benefits of adopting a resource-oriented approach to information sharing are illustrated. It is important to note that these use cases are intended to be a small representative set and is by no means meant to be an exhaustive list. The intent is to illustrate how the use of link relationship types will enable this resource-oriented approach to cyber security information sharing to successfully support the complete range of existing use cases, and also to motivate an initial list of well-defined link relationship types.

4.2.4.1. Use Case: Incident Sharing

This section provides a non-normative example of an incident sharing use case.

In this use case, a member CSIRT shares incident information with another member CSIRT in the same consortium. The client CSIRT retreives a feed of incidents, and is able to identify one particular entry of interest. The client then does an HTTP GET on that entry, and the representation of that resource contains link relationships for both the associated "indicators" and the incident "history", and so on. The client CSIRT recognizes that some of the indicator and history may be relevant within her local environment, and can respond proactively.

Example HTTP GET response for an incident entry:

```
      <?xml version="1.0" encoding="UTF-8"?>
      <entry>
        <id>http://www.example.org/csirt/private/incidents/123456</id>
        <title>Sample Incident</title>
        <link href="http://www.example.org/csirt/private/incidents/123456" rel="
self"/>       <!-- by convention -->
        <link href="http://www.example.org/csirt/private/incidents/123456" rel="
alternate"/>  <!-- required by Atom spec -->
        <published>2012-08-04T18:13:51.0Z</published>
        <updated>2012-08-05T18:13:51.0Z</updated>

        <link href="http://www.example.org/csirt/private/incidents/123456" rel="
edit"/>

        <!-- The links to indicators related to this incident, and the history o
f this incident, and so on.... -->
        <link href="http://www.example.org/csirt/private/incidents/123456/relati
onships/indicators" rel="indicators"/>
        <link href="http://www.example.org/csirt/private/incidents/1234456/relat
ionships/history" rel="history"/>
        <link href="http://www.example.org/csirt/private/incidents/1234456/relat
ionships/campaign" rel="campaign"/>

        <!-- navigate up to the full collection.  Might also be rel="collection"
 as per IANA registry -->
        <link href="http://www.example.org/csirt/private/incidents" rel="up"/>

        <content type="application/xml">
          <iodef:IODEF-Document lang="en" xmlns:iodef="urn:ietf:params:xml:ns:io
def-1.0">
            <iodef:Incident purpose="traceback" restriction="need-to-know">
              <iodef:IncidentID name="http://www.example.org/csirt/private/incid
ents">123456</iodef:IncidentID>
              <!-- ...additional incident data.... -->
              </iodef:Incident>
          </iodef:IODEF-Document>
        </content>
      </entry>
```

As can be seen in the example response, the Atom <link> elements enable the client to navigate to the related indicator resources, and/or the history entries associated with this incident.

4.2.4.2. Use Case: Collaborative Investigation

This section provides a non-normative example of a collaborative investigation use case.

In this use case, two member CSIRTs that belong to a closed sharing consortium are collaborating on an incident investigation. The initiating CSIRT performs an HTTP GET to retrieve the service document of the peer CSIRT, and determines the collection name to be used for creating a new investigation request. The initiating CSIRT then POSTs a new incident entry to the appropriate collection URL.

The target CSIRT acknowledges the request by responding with an HTTP status code 201 Created.

Example HTTP GET response for the service document:

```
HTTP/1.1 200 OK
Date: Fri, 24 Aug 2012 17:09:11 GMT
Content-Length: 934
Content-Type: application/atomsvc+xml;charset="utf-8"

<?xml version="1.0" encoding="UTF-8"?>
<service xmlns="http://www.w3.org/2007/app"
         xmlns:atom="http://www.w3.org/2005/Atom">
    <workspace xml:lang="en-US" xmlns:xml="http://www.w3.org/XML/1998/name
space">
      <atom:title type="text">RID Use Case Requests</atom:title>
      <collection  href="http://www.example.org/csirt/RID/InvestigationReq
uests">
         <atom:title type="text">Investigation Requests</atom:title>
         <accept>application/atom+xml; type=entry</accept>  <!-- perhaps w
e should have a more specific media type -->
      </collection>
      <collection  href="http://www.example.org/csirt/RID/TraceRequests">
         <atom:title type="text">Trace Requests</atom:title>
         <accept>application/atom+xml; type=entry</accept>
      </collection>
      <!-- ...and so on.... -->
    </workspace>
</service>
```

As can be seen in the example response, the Atom <collection> elements enable the client to determine the appropriate collection URL to request an investigation or a trace.

The client CSIRT then POSTs a new entry to the appropriate feed collection. Note that the <content> element of the new entry may contain a RID message of type "InvestigationRequest" if desired, however this would NOT be required. The entry content itself need only be an IODEF document, with the choice of the target collection resource URL indicating the callers intent. A CSIRT would be free to use any URI template to accept investigationRequests.

```
POST /csirt/RID/InvestigationRequests HTTP/1.1
Host: www.example.org
Content-Type: application/atom+xml;type=entry
Content-Length: 852

<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://www.w3.org/2005/Atom">
  <title>New Investigation Request</title>
  <id>http://www.example2.org/csirt/private/incidents/123456</id>  <!-- id and u
pdated not guranteed to be preserved -->
  <updated>2012-08-12T11:08:22Z</updated>                          <!-- may want
to profile that behavior in this document -->
  <author><name>Name of peer CSIRT</name></author>
  <content type="application/xml">
    <iodef:IODEF-Document lang="en" xmlns:iodef="urn:ietf:params:xml:ns:iodef-1.
0">
      <iodef:Incident purpose="traceback" restriction="need-to-know">
      <iodef:IncidentID name="http://www.example2.org/csirt/private/incidents">1
23</iodef:IncidentID>
        <!-- ...additional incident data.... -->
      </iodef:Incident>
    </iodef:IODEF-Document>
  </content>
</entry>
```

The receiving CSIRT acknowledges the request with HTTP return code 201 Created.

```
HTTP/1.1 201 Created
Date: Fri, 24 Aug 2012 19:17:11 GMT
Content-Length: 906
Content-Type: application/atom+xml;type=entry
Location: http://www.example.org/csirt/RID/InvestigationRequests/823
ETag: "8a9h9he4qphqh"

<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://www.w3.org/2005/Atom">
  <title>New Investigation Request</title>
  <id>http://www.example.org/csirt/RID/InvestigationRequests/823</id>  <!-- id a
nd updated not guranteed to be preserved -->
  <updated>2012-08-12T11:08:30Z</updated>                              <!-- may
want to profile that behavior in this document -->
  <published>2012-08-12T11:08:30Z</published>
  <author><name>Name of peer CSIRT</name></author>
  <content type="application/xml">
    <iodef:IODEF-Document lang="en" xmlns:iodef="urn:ietf:params:xml:ns:iodef-1.
0">
      <iodef:Incident purpose="traceback" restriction="need-to-know">
      <iodef:IncidentID name="http://www.example.org/csirt/private/incidents">12
3</iodef:IncidentID>
        <!-- ...additional incident data.... -->
      </iodef:Incident>
    </iodef:IODEF-Document>
  </content>
</entry>
```

Consistent with HTTP/1.1 RFC, the location header indicates the URL of the newly created InvestigationRequest. If for some reason the request were not authorized, the client would receive an HTTP status code 403 Unauthorized. In this case the HTTP response body may contain additional details, if an as appropriate.

4.2.4.3. Use Case: Search (Query)

This section provides a non-normative example of a search use case.

The following example provides a RESTful alternative to the RID Query messaage, as described in sections 6.5 and 7.4 of RFC6545. Note that in the RESTful approach described herein there is no requirement to define a query language specific to RID queries. Instead, CSIRTs may provide support for search operations via existing search facilities, and advertise these capabilities via an appropriate URL template. Clients dynamically retrieve the search description document, and invoke specific searches via an instantiated URL template.

An HTTP response body may include a link relationship of type "search." This link provides a reference to an OpenSearch description document.

Example HTTP response that includes a "search" link:

```
HTTP/1.1 200 OK
Date: Fri, 24 Aug 2012 17:20:11 GMT
Content-Length: nnnn
Content-Type: application/atom+xml;type=feed;charset="utf-8"

<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.w3.org/2005/Atom file:/C:/schemas/atom.
xsd
                        urn:ietf:params:xml:ns:iodef-1.0 file:/C:/schemas/
iodef-1.0.xsd"
    xml:lang="en-US">

    <link href="http://www.example.org/opensearchdescription.xml" rel="sea
rch"
            type="application/opensearchdescription+xml"
            title="CSIRT search facility" />

    <!-- ...other links... -->

    <entry>
        <!-- ...zero or more entries... -->
    </entry>

</feed>
```

The OpenSearch Description document contains the information needed by a client to request a search. An example of an Open Search description document is shown below:

Example HTTP response that includes a "search" link:

```
<?xml version="1.0" encoding="UTF-8"?>
<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/
">
  <ShortName>CSIRT search example</ShortName>
  <Description>Cyber security information sharing consortium searc
h interface</Description>
  <Tags>example csirt indicator search</Tags>
  <Contact>admin@example.org</Contact>
  <!-- ...optionally, other elements, as per OpenSearch specificat
ion... -->
  <Url type="application/opensearchdescription+xml" rel="self" tem
plate="http://www.example.com/csirt/opensearchdescription.xml"/>
  <Url type="application/atom+xml" rel="results" template="http://
www.example.org/csirt?q={searchTerms}&amp;format=Atom+xml"/>
  <LongName>www.example.org CSIRT search</LongName>
  <Query role="example" searchTerms="incident" />
  <Language>en-us</Language>
  <OutputEncoding>UTF-8</OutputEncoding>
  <InputEncoding>UTF-8</InputEncoding>
</OpenSearchDescription>
```

The OpenSearch Description document shown above contains two <Url> elements that contain parameterized URL templates. These templates provide a representation of how the client should make search requests. The exact format of the query string, including the parameterization is specified by the feed provider.

This OpenSearch Description Document also contains an example of a <Query> element. Each <Query> element describes a specific search request that can be made by the client. Note that the parameters of the <Query> element correspond to the URL template parameters. In this way, a provider may fully describe the search interface available to the clients. Section 5.12, below, provides specific NORMATIVE requirements for the use of Open Search.

4.2.4.4. Use Case: Cyber Data Repository

This section provides a non-normative example of a cyber security data repository use case.

In this use case a client accesses a persistent repository of cyber security data via a RESTful usage model. Retrieving a feed collection is analogous to an SQL SELECT statement producing a result set. Retrieving an individual Atom Entry is analogous to a SQL SELECT statement based upon a primary key producing a unique record. The cyber security data contained in the repository may include different data types, including indicators, incidents, becnmarks, or

any other related resources. In this use case, the repository is queried via HTTP GET, and the results that are returned to the client may optionally contain URL references to other cyber security resources that are known to be related. These related resources may also be persisted locally, or they may exist at another (remote) cyber data respository.

Example HTTP GET request to a persistent repository for any resources representing Distributed Denial of Service (DDOS) attacks:

```
GET /csirt/repository/ddos
Host: www.example.org
Accept: application/atom+xml
```

The corresponding HTTP response would be an XML document containing the DDOS feed.

Example HTTP GET response for a DDOS feed:

```
      HTTP/1.1 200 OK
      Date: Fri, 24 Aug 2012 17:20:11 GMT
      Content-Length: nnnn
      Content-Type: application/atom+xml;type=feed;charset="utf-8"

      <?xml version="1.0" encoding="UTF-8"?>
      <feed xmlns="http://www.w3.org/2005/Atom"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://www.w3.org/2005/Atom file:/C:/schemas/atom.
xsd
                              urn:ietf:params:xml:ns:iodef-1.0 file:/C:/schemas/
iodef-1.0.xsd"
          xml:lang="en-US">

          <generator version="1.0" xml:lang="en-US">emc-csirt-iodef-feed-service
</generator>
          <id xml:lang="en-US">http://www.example.org/csirt/repository/ddos</id>
          <title type="text" xml:lang="en-US">Atom formatted representation of a
 feed of known ddos resources.</title>
          <updated xml:lang="en-US">2012-05-04T18:13:51.0Z</updated>
          <author>
              <email>csirt@example.org</email>
              <name>EMC CSIRT</name>
          </author>

          <!-- By convention there is usually a self link for the feed -->
          <link href="http://www.example.org/csirt/repository/ddos" rel="self"/>

          <entry>
              <id>http://www.example.org/csirt/repository/ddos/123456</id>
              <title>Sample DDOS Incident</title>
              <link href="http://www.example.org/csirt/repository/ddos/123456" r
el="self"/>          <!-- by convention -->
              <link href="http://www.example.org/csirt/repository/ddos/123456" r
el="alternate"/>     <!-- required by Atom spec -->
              <link href="http://www.example.org/csirt/repository/ddos/987654" r
el="related"/>       <!-- link to a related DDOS resource in this repository -->
              <link href="http://www.cyber-agency.gov/repository/indicators/1a2b
3c" rel="related"/>  <!-- link to a related DDOS resource in another repository
-->
              <published>2012-08-04T18:13:51.0Z</published>
              <updated>2012-08-05T18:13:51.0Z</updated>
              <!-- The category is based upon IODEF purpose and restriction attr
ibutes -->
              <category term="traceback" scheme="purpose" label="trace back" />
              <category term="need-to-know" scheme="restriction" label="need to
know" />
              <category term="ddos" scheme="ttp" label="tactics, techniques, and
 procedures"/>
              <summary>A short description of this DDOS attack, extracted from t
he IODEF Incident class, <description> element. </summary>
          </entry>

          <entry>
              <!-- ...another entry... -->
          </entry>

      </feed>
```

This feed document has two atom entries, one of which has been elided. The completed entry illustrates an Atom <entry> element that provides a summary of essential details about one particular DDOS incident. Based upon this summary information and the provided category information, a client may choose to do an HTTP GET operation to retrieve the full details of the DDOS incident. This example shows how a persistent repository may provide links to additional resources, both local and remote.

Note that the provider of a persistent repostory is not obligated to follow any particular URL template scheme. The repository available at the hypothetical provider "www.example.com" uses a different URL pattern than the hypothetical repository available at "www.cyber-agency.gov". When a client de-references a link to resource that is located in a remote repository the client may be challenged for authentication credentials acceptable to that provider. If the two repository providers choose to support a federated identity scheme or some other form of single-sign-on technology, then the user experience can be improved for interactive clients (e.g., a human user at a browser). However, this is not required and is an implementation choice that is out of scope for this specification.

5. Requirements for RESTful (Atom+xml) Binding

This section provides the NORMATIVE requirements for using Atom format and Atom Pub as a RESTful binding for cyber security information sharing.

5.1. Transport Layer Security

Servers implementing this specification MUST support server-authenticated TLS.

Servers MAY support mutually authenticated TLS.

5.2. User Authentication

Servers MUST require user authentication.

Servers MAY support more than one client authentication method.

Servers participating in an information sharing consotium and supporting interactive user logins by members of the consortium SHOULD support client authentication via a federated identity scheme as per SAML 2.0.

Servers MAY support client authenticated TLS.

5.3. User Authorization

This document does not mandate the use of any specific user authorization mechanisms. However, service implementers SHOULD provide appropriate authorization checking for all resource accesses, including individual Atom Entries, Atom Feeds, and Atom Service Documents.

Authorization for a resource MAY be adjudicated based on the value(s) of the associated Atom <category> element(s).

When the content model for the Atom <content> element of an Atom Entry contains an <IODEF-Document>, then authorization MUST be adjudicated based upon the Atom <category> element(s), whose values have been mapped as per Section 5.7.

Any use of the <category> element(s) as an input to an authorization policy decision MUST include both the "scheme" and "term" attributes contained therein. As described in Section 5.7 below, the namespace of the "term" attribute is scoped by the associated "scheme" attribute.

5.4. Content Model

Member entry resources providing a representation of an incident resource (e.g., as specified in the link relation type) MUST use the IODEF schema as the content model for the Atom Entry <content> element.

Member Entry resources providing a representation of an indicator resource (e.g., as specified in the link relation type) MUST use the IODEF schema as the content model for the Atom Entry <content> element.

The resource representation MAY include an appropriate indicator schema type within the <AdditionalData> element of the IODEF Incident class. Supported indicator schema types SHALL be registered via an IANA table (todo: IANA registration/review).

Member Entry resources providing a representation of a RID report resource (e.g., as specified in the link relation type) MUST use the RID schema as the content model for the Atom Entry <content> element.

Member Entry resources providing representation of other types, SHOULD use the IODEF schema as the content model for the Atom Entry <content> element.

If the member entry content model is not IODEF, then the <content> element of the Atom entry MUST contain an appropriate XML namespace declaration.

5.5. HTTP methods

The following table defines the HTTP [RFC2616] uniform interface methods supported by this specification:

HTTP method	Description
GET	Returns a representation of an individual member entry resource, or a feed collection.
PUT	Replaces the current representation of the specified member entry resource with the representation provided in the HTTP request body.
POST	Creates a new instance of a member entry resource. The representation of the new resource is provided in the HTTP request body.
DELETE	Removes the indicated member entry resource, or feed collection.
HEAD	Returns metadata about the member entry resource, or feed collection, contained in HTTP response headers.
PATCH	Support TBD.

Table 1: Uniform Interface for Resource-Oriented Lightweight Indicator Exchange

Clients MUST be capable of recognizing and prepared to process any standard HTTP status code, as defined in [RFC2616]

5.6. Service Discovery

This specification requires that a CSIRT MUST publish an Atom Service Document that describes the set of cyber security information sharing feeds that are provided.

The service document SHOULD be discoverable via the CSIRT organization's Web home page or another well-known public resource.

5.6.1. Workspaces

The service document MAY include multiple workspaces. Any CSIRT providing both public feeds and private consortium feeds MUST place these different classes of feeds into different workspaces, and

provide appropriate descriptions and naming conventions to indicate the intended audience of each workspace.

5.6.2. Collections

A CSIRT MAY provide any number of collections within a given Workspace. It is RECOMMENDED that each collection appear in only a single Workspace. It is RECOMMENDED that at least one collection be provided that accepts new incident reports from users. At least one collection MUST provide a feed of incident information for which the content model for the entries uses the IODEF schema. The title of this collection SHOULD be "Incidents".

5.6.3. Service Document Security

Access to the service document MUST be protected via server-authenticated TLS and a server-side certificate.

When deploying a service document for use by a closed consortium, the service document MAY also be digitally signed and/or encrypted, using XML DigSig and/or XML Encryption, respectively.

5.7. Category Mapping

This section defines normative requirements for mapping IODEF metadata to corresponding Atom category elements. (todo: decide between IANA registration of scheme, or use a full URI).

5.7.1. Collection Category

An Atom collection MAY hold entries from one or more categories. The collection category set MUST contain at least the union of all the member entry categories. A collection MAY have additional category metadata that are unique to the collection, and not applicable to any individual member entry. A collection containing IODEF incident content MUST contain at least two <category> elements. One category MUST be specified with the value of the "scheme" attribute as "restriction". One category MUST be specified with the value of the "scheme" attribute as "purpose". The value of the "fixed" attribute for both of these category elements MUST be "yes". When the category scheme="restriction", the allowable values for the "term" attribute are constrained as per section 3.2 of IODEF, e.g. public, need-to-know, private, default. When the category scheme="purpose", the allowable values for the "term" attribute are constrained as per section 3.2 of IODEF, e.g. traceback, mitigation, reporting, other.

5.7.2. Entry Category

An Atom entry containing IODEF content MUST contain at least two <category> elements. One category MUST be specified with the value of the "scheme" attribute as "restriction". One category MUST be specified with the value of the "scheme" attribute as "purpose". When the category scheme="restriction", the value of the "term" attribute must be exactly one of (public, need-to-know, private, default). When the category scheme="purpose", the value of the "term" attribute must be exactly one of (traceback, mitigation, reporting, other). When the purpose is "other"....

Any member entry MAY have any number of additional categories.

5.8. Entry ID

The ID element for an Atom entry SHOULD be established via the concatenation of the value of the name attribute from the IODEF <IncidentID> element and the corresponding value of the <IncidentID> element. This requirement ensures a simple and direct one-to-one relationship between an IODEF incident ID and a corresponding Feed entry ID and avoids the need for any system to maintain a persistent store of these identity mappings.

(todo: Note that this implies a constraint on the IODEF document that is more restrictive than the current IODEF schema. IODEF section 3.3 requires only that the name be a STRING type. Here we are stating that name must be an IRI. Possible request to update IODEF to constrain, or to support a new element or attribute).

5.9. Entry Content

The <content> element of an Atom <entry> SHOULD include an IODEF document. The <entry> element SHOULD include an appropriate XML namespace declaration for the IODEF schema. If the content model of the <entry> element does not follow the IODEF schema, then the <entry> element MUST include an appropriate XML namespace declaration.

A client MAY ignore content that is not using the IODEF schema.

5.10. Link Relations

In addition to the standard Link Relations defined by the Atom specification, this specification defines the following additional Link Relation terms, which are introduced specifically in support of the Resource-Oriented Lightweight Indicator Exchange protocol.

Name	Description	Conformance
service	Provides a link to an atom service document associated with the collection feed.	MUST
search	Provides a link to an associated Open Search document that describes a URL template for search queries.	MUST
history	Provides a link to a collection of zero or more historical entries that are associated with the resource.	MUST
incidents	Provides a link to a collection of zero or more instances of actual cyber security event(s) that are associated with the resource.	MUST
indicators	Provides a link to a collection of zero or more instances of cyber security indicators that are associated with the resource.	MUST
evidence	Provides a link to a collection of zero or more resources that provides some proof of attribution for an incident. The evidence may or may not have any identified chain of custody.	SHOULD
campaign	Provides a link to a collection of zero or more resources that provides a representation of the associated cyber attack campaign.	SHOULD
attacker	Provides a link to a collection of zero or more resources that provides a representation of the attacker.	SHOULD
vector	Provides a link to a	SHOULD

	collection of zero or more resources that provides a representation of the method used by the attacker.	
assessments	Provides a link to a collection of zero or more resources that represent the results of executing a benchmark.	SHOULD
reports	Provides a link to a collection of zero or more resources that represent RID reports.	SHOULD
traceRequests	Provides a link to a collection of zero or more resources that represent RID traceRequests.	SHOULD
investigationRequests	Provides a link to a collection of zero or more resources that represent RID investigationRequests.	SHOULD

Table 2: Link Relations for Resource-Oriented Lightweight Indicator Exchange

Unless specifically registered with IANA these short names MUST be fully qualified via concatenation with a base-uri. An appropriate base-uri could be established via agreement amongst the members of an information sharing consortium. For example, the rel="indicators" relationship would become rel="http://www.example.org/csirt/incidents/relationships/indicators."

5.10.1. Additional Link Relation Requirements

An IODEF document that is carried in an Atom Entry SHOULD NOT contain a <relatedActivity> element. Instead, the related activity SHOULD be available via a link rel=related.

An IODEF document that is carried in an Atom Entry SHOULD NOT contain a <history> element. Instead, the related history SHOULD be available via a link rel="history" (todo: or a fully qualified link rek name). The associated href MAY leverage OpenSearch to specify the required query.

An Atom Entry MAY include additional link relationships not specified here. If a client encounters a link relationship of an unkown type the client MUST ignore the offending link and continue processing the remaining resource representation as if the offending link element did not appear.

5.11. Member Entry Forward Security

As described in Authorization Policy Enforcement (Authorization Policy Enforcement) a RESTful model for cyber security information sharing requires that all of the required security enforcement for feeds and entries MUST be enforced at the source system, at the point the representation of the given resource(s) is created. A CSIRT provider SHALL NOT return any feed content or member entry content for which the client identity has not been specifically authenticated, authorized, and audited.

Sharing communities that have a requirement for forward message security (such that client systems are required to participate in providing message level security and/or distributed authorization policy enforcement), MUST use the RID schema as the content model for the member entry <content> element.

5.12. Date Mapping

The Atom feed <updated> element MUST be populated with the current time at the instant the feed representation was generated. The Atom entry <published> element MUST be populated with the same time value as the <reportTime> element from the IODEF document.

5.13. Search

Implementers MUST support OpenSearch 1.1 [opensearch] as the mechanism for describing how clients may form search requests.

Implementers MUST provide a link with a relationship type of "search". This link SHALL return an Open Search Description Document as defined in OpenSearch 1.1.

Implementers MUST support an OpenSearch 1.1 compliant search URL template that enables a search query via Atom Category, including the scheme attribute and terms attribute as search parameters.

Implementers SHOULD support search based upon the IODEF AlternativeID class as a search parameter.

Implementers SHOULD support search based upon the four timestamp elements of the IODEF Incident class: <startTime>, <EndTime>, <DetectTime>, and <ReportTime>.

Implementers MAY support additional search capabilities based upon any of the remaining elements of the IODEF Incident class, including the <Description> element.

Collections that support use of the RID schema as a content model in the Atom member entry <content> element (e.g. in a report resource representation reachable via the "report" link relationship) MUST support search operations that include the RID MessageType as a search parameter, in addition to the aforementioned IODEF schema elements, as contained within the <ReportSchema> element.

Implementers MUST fully qualify all OpenSearch URL template parameter names using the defined IODEF or RID XML namespaces, as appropriate.

5.14. / (forward slash) Resource URL

The "/" resource MAY be provided for compatibility with existing deployments that are using Transport of Real-time Inter-network Defense (RID) Messages over HTTP/TLS [RFC6546]. Consistent with RFC6546 errata, a client requesting a GET on "/" MUST receive an HTTP status code 405 Method Not Allowed. An implementation MAY provide full support for RFC6546 such that a POST to "/" containing a recognized RID message type just works. Alternatively, a client requesting a POST to "/" MAY receive an HTTP status code 307 Temporary Redirect. In this case, the location header in the HTTP response will provide the URL of the appropriate RID endpoint, and the client may repeat the POST method at the indicated location. This resource could also leverage the new draft by reschke that proposes HTTP status code 308 (cf: draft-reschke-http-status-308-07.txt).

6. Security Considerations

This document defines a resource-oriented approach to lightweight indicator exchange using HTTP, TLS, Atom Syndicate Format, and Atom Publishing Protocol. As such, implementers must understand the security considerations described in those specifications.

In addition, there are a number of additional security considerations that are unique to this specification.

As described above in the section Authentication of Users (Section 3.2), the approach described herein is based upon all policy enforcements being implemented at the point when a resource

representation is created. As such, CSIRTS sharing cyber security information using this specification must take care to authenticate their HTTP clients using a suitably strong user authentication mechanism. Sharing communities that are exchanging information on well-known indicators and incidents for purposes of public education may choose to rely upon, e.g. HTTP Authentication, or similar. However, sharing communities that are engaged in sensitive collaborative analysis and/or operational response for indicators and incidents targeting high value information systems should adopt a suitably stronger user authentication solution, such as TLS client certificates, or a risk-based or multi-factor approach. In general, trust in the sharing consortium will depend upon the members maintaining adequate user authentication mechanisms.

Collaborating consortiums may benefit from the adoption of a federated identity solution, such as those based upon SAML-core [SAML-core] and SAML-bind [SAML-bind] and SAML-prof [SAML-prof] for Web-based authentication and cross-organizational single sign-on. Dependency on a trusted third party identity provider implies that appropriate care must be exercised to sufficiently secure the Identity provider. Any attacks on the federated identity system would present a risk to the CISRT, as a relying party. Potential mitigations include deployment of a federation-aware identity provider that is under the control of the information sharing consortium, with suitably stringent technical and management controls.

As discussed above in the section Authorization Policy Enforcement (Section 3.3), authorization of resource representations is the responsibility of the source system, i.e. based on the authenticated user identity associated with an HTTP(S) request. The required authorization policies that are to be enforced must therefore be managed by the security administrators of the source system. Various authorization architectures would be suitable for this purpose, such as RBAC [1] and/or ABAC, as embodied in XACML [XACML]. In particular, implementers adopting XACML may benefit from the capability to represent their authorization policies in a standardized, interoperable format.

Additional security requirements such as enforcing message-level security at the destination system could supplement the security enforcements performed at the source system, however these destination-provided policy enforcements are out of scope for this specification. Implementers requiring this capability should consider leveraging, e.g. the <RIDPolicy> element in the RID schema. Refer to RFC6545 section 9 for more information.

When security policies relevant to the source system are to be enforced at both the source and destination systems, implementers must take care to avoid unintended interactions of the separately enforced policies. Potential risks will include unintended denial of service and/or unintended information leakage. These problems may be mitigated by avoiding any dependence upon enforcements performed at the destination system. When distributed enforcement is unavoidable, the usage of a standard language (e.g. XACML) for the expression of authorization policies will enable the source and destination systems to better coordinate and align their respective policy expressions.

Adoption of the information sharing approach described in this document will enable users to more easily perform correlations across separate, and potentially unrelated, cyber security information providers. A client may succeed in assembling a data set that would not have been permitted within the context of the authorization policies of either provider when considered individually. Thus, providers may face a risk of an attacker obtaining an access that constitutes an undetected separation of duties (SOD) violation. It is important to note that this risk is not unique to this specification, and a similar potential for abuse exists with any other cyber security information sharing protocol. However, the wide availability of tools for HTTP clients and Atom feed handling implies that the resources and technical skills required for a successful exploit may be less than it was previously. This risk can be best mitigated through appropriate vetting of the client at account provisioning time. In addition, any increase in the risk of this type of abuse should be offset by the corresponding increase in effectiveness that that this specification affords to the defenders.

While it is a goal of this specification to enable more agile cyber security information sharing across a broader and varying constituency, there is nothing in this specification that necessarily requires this type of deployment. A cyber security information sharing consortium may chose to adopt this specification while continuing to operate as a gated community with strictly limited membership.

7. IANA Considerations

If the values of the newly defined link relations are not fully qualified URIs then we need to register these link types with IANA (e.g. rel="history") It is possible to adjust this document so that it has no actions for IANA.

8. ToDo and Open Issues

The following is the "todo" and open issues list:

1. Need to make a decision on whether new IANA link registrations are required, or whether fully qualified (private) link types are sufficient.

2. Should we require Atom categories that correspond to IODEF Expectation class and/or IODEF Impact class?

3. Should we include specific requirements for Archive and Paging? Perhaps just reference RFC 5005?

4. We need more requirements input on use cases involving RID schema in the Atom member entry content model for link rel=report.

5. An Atom service document will have categories, but this is still coarse-grained, and not visible at the transport protocol level. Should we include a MIME media type parameter to support negotiation and better document the content model schema contained in a collection, i.e.:

   Accept: application/atom+xml;type=entry;content=iodef

   Accept: application/atom+xml;type=entry;content=rid

   Accept: application/atom+xml;type=entry;content=iodef+openioc

6. If so, I think these parameters may require media type registration as per RFC4288?

9. Acknowledgements

The author gratefully acknowledges the valuable contributions of Tom Maguire, Kathleen Moriarty, and Vijayanand Bharadwaj. These individuals provided detailed review comments on earlier drafts, and many suggestions that have helped to improve this document .

10. References

10.1. Normative References

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

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC4287] Nottingham, M., Ed. and R. Sayre, Ed., "The Atom Syndication Format", RFC 4287, December 2005.

[RFC5023] Gregorio, J. and B. de hOra, "The Atom Publishing Protocol", RFC 5023, October 2007.

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

[RFC6545] Moriarty, K., "Real-time Inter-network Defense (RID)", RFC 6545, April 2012.

[opensearch] Clinton, D., "OpenSearch 1.1 draft 5 specification", 2011, <http://www.opensearch.org/Specifications/OpenSearch/1.1>.

[SAML-core] Cantor, S., Kemp, J., Philpott, R., and E. Mahler, "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard , March 2005, <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>.

[SAML-prof] Hughes, J., Cantor, S., Hodges, J., Hirsch, F., Mishra, P., Philpott, R., and E. Mahler, "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard , March 2005, <http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf>.

[SAML-bind] Cantor, S., Hirsch, F., Kemp, J., Philpott, R., and E. Mahler, "Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard , March 2005, <http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf>.

10.2. Informative References

[XMLencrypt] Imaura, T., Dillaway, B., and E. Simon, "XML Encryption Syntax and Processing", W3C Recommendation , December 2002, <http://www.w3.org/TR/xmlenc-core/>.

[XMLsig] Bartel, M., Boyer, J., Fox, B., LaMaccia, B., and E. Simon, "XML-Signature Syntax and Processing", W3C Recommendation Second Edition, June 2008, <http://www.w3.org/TR/xmldsig-core/>.

[XACML] Rissanen, E., "eXtensible Access Control Markup Language (XACML) Version 3.0", August 2010, <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cs-01-en.pdf>.

[REST] Fielding, R., "Architectural Styles and the Design of Network-based Software Architectures", 2000, <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>.

[RFC2396] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.

[RFC2822] Resnick, P., "Internet Message Format", RFC 2822, April 2001.

[RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

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

[RFC6546] Trammell, B., "Transport of Real-time Inter-network Defense (RID) Messages over HTTP/TLS", RFC 6546, April 2012.

10.3. URIs

[1] http://csrc.nist.gov/groups/SNS/rbac/

Appendix A. Change Tracking

Changes since -00 version, September 5, 2012 to Feb 15, 2013:

- Fixed a small number of typographical errors and a few misspellings throughout.

- Added a number of missing internal cross references to improve readability.
- Updated the text in the Introduction section for improved brevity and clarity of goal. See: Section 1
- Added new non-normative text describing the use of HTTP 4xx status codes for authorization. See: Section 3.3.2
- Added a new non-normative example illustrating a persistent repository use case. See: Section 4.2.4.4
- Added new normative text recommending use of SAML2 for authentication of interactive end users who are members of a sharing consortium. See: Section 5.2
- Added new normative text describing requirements for user authorization. See: Section 5.3
- Added non-normative appendix for change tracking. See: Appendix A
- Added non-normative appendix describing a suggested approach to a XACML profile. See: Appendix B

Appendix B. Resource Authorization Model

As described in Section 3.3.2 above, ROLIE assumes that all authorization policy enforcement is provided at the source server. The implementation details of the authorization scheme chosen by a ROLIE-compliant provider are out of scope for this specification. Implementers are free to choose any suitable authorization mechanism that is capable of fulfilling the policy enforcement requirements relevant to their consortium and/or organization.

It is well known that one of the major barriers to information sharing is ensuring acceptable use of the information shared. In the case of ROLIE, one way to lower that barrier may be to develop a XACML profile. Use of XACML would allow a ROLIE-compliant provider to express their information sharing authorization policies in a standards-compliant, and machine-readable format.

This improved interoperability may, in turn, enable more agile interactions in the cyber security sharing community. For example, a peer CSIRT, or another interested stakeholder such as an auditor, would be able to review and compare CSIRT sharing policies using appropriate tooling.

The XACML 3.0 standard is based upon the notion that authorization policies are defined in terms of predicate logic expressions written against the attributes associated with one or more of the following four entities:

- SUBJECT
- ACTION
- RESOURCE
- ENVIRONMENT

Thus, a suitable approach to a XACML 3.0 profile for ROLIE authorization policies could begin by using the 3-tuple of [SUBJECT, ACTION, RESOURCE] where:

- SUBJECT is the suitably authenticated identity of the requestor.
- ACTION is the associated HTTP method, GET, PUT, POST, DELETE, HEAD, (PATCH).
- RESOURCE is an XPath expression that uniquely identifies the instance or type of the ROLIE resource being requested.

Implementers who have a need may also choose to evaluate based upon the additional ENVIRONMENT factors, such as current threat level, and so on. One could also write policy to consider the CVSS score associated with the resource, or the lifecycle phase of the resource (vulnerability unverified, confirmed, patch available, etc.), and so on.

Having these policies expressed in a standards-compliant and machine-readable format could improve the agility and effectiveness of a cyber security information sharing group or consortium, and enable better cyber defenses.

B.1. Example XACML Profile

Work-in-Progress. If this aproach finds support in the community then this section (or a new draft, as a seperate document) could provide a more complete XACML 3.0 compliant example.

Author's Address

John P. Field
EMC Corporation
1133 Westchester Avenue
White Plains, New York
USA

Phone: 914-461-3522
Email: jfield@pivotal.io